About This Manual

This chapter describes the purpose, organization, and conventions of this document.

Audience and Purpose

This reference manual details the syntax of each AFS® command and is intended for the experienced AFS administrator, programmer, or user.

In general, this document does not explain when to use a command or its place in the sequence of commands that make up a complete procedure. For that type of information, refer to the *AFS System Administrator's Guide*.

Organization

This document organizes the command descriptions into suites and presents the suites in alphabetical order. (A ***suite*** is a group of related commands that begin with the same command name.) Commands that do not belong to a suite appear in alphabetical order in the final *Miscellaneous Commands* section. A *Command Summary* at the end lists the syntax for all commands in this manual.

Tabbed dividers separate the command suites, as follows:

- Table of Contents
- BACKUP
- BOS
- FS
- KAS
- PACKAGE
- PTS
- USS
- VOS
- Miscellaneous AFS Commands
- Modified UNIX Commands
- Command Summary

- Index

The command descriptions appear alphabetically within each command suite. Commands always begin on a new page and are described using the following common format:

- COMMAND NAME and a brief description of the command's function appears on the first line.
- COMPLETE COMMAND SYNTAX specifies the required order for all arguments and flags. The syntax matches the online help and uses the symbols described on page x.
- ACCEPTABLE ABBREVIATIONS/ALIASES lists the shortest legal abbreviation for the command's operation code, switches, and flags. If the operation code has an alias, it also appears here. Note that any longer abbreviations than those shown are also legal.
- DESCRIPTION explains the command's function in detail.
- ARGUMENTS describes the function and required form of each argument and flag.
- WARNINGS indicates potential complications or damaging (that is, irreversible) effects of issuing the command. This section appears only when necessary.
- OUTPUT describes the on-screen output of the command, if any. This section is omitted if the command does not produce output or if the only output is a message confirming that the command executed successfully.
- EXAMPLES provides one or more sample commands and resulting output (if any).
- PRIVILEGE REQUIRED lists the privilege(s) required to perform the command.
- MORE INFORMATION points to related commands, if any.

How to Use This Document

Refer to this document when you need detailed information about a specific command. For quick reference to command syntax only, refer to the *Command Summary* at the end of the document.

This document is intended to be used as a reference, not as a guide to performing administrative tasks. For a description of all the steps in a procedure, refer to the *AFS System Administrator's Guide*.

Related Documents

You should also be familiar with the following documents, included in the AFS Documentation Kit:

AFS Installation Guide

This guide provides instructions for installing an AFS system. The procedures described in this document should be performed by an experienced UNIX® administrator when setting up the initial system.

Prior to using any other AFS document, the administrator should verify that AFS has been set up in the cell in accordance with this guide. Some commands only work as described if that is the case.

AFS System Administrator's Guide

This guide describes the concepts and procedures that a system administrator must know to manage an AFS system. It assumes familiarity with UNIX, but requires no previous knowledge of AFS.

The first section of the *AFS System Administrator's Guide* presents basic concepts and guidelines. It is critical that administrators be familiar with the information in this section before attempting to use or manage an AFS system. The remaining sections in the guide provide step-by-step instructions for specific administrative tasks, along with discussions of the concepts important to that particular task.

The *AFS System Administrator's Guide* is complemented by the *AFS Command Reference Manual*. It describes commands in the context of specific tasks, but not as completely as the *AFS Command Reference Manual*.

AFS User's Guide

This guide presents the basic concepts and procedures necessary for using AFS effectively. It assumes that the reader has some experience with UNIX, but does not require familiarity with networking or AFS.

The guide explains how to perform basic functions including logging in, changing a password, listing information, protecting files, creating groups, and troubleshooting. It provides illustrative examples for each function and describes some of the differences between the UNIX file system and AFS.

Document Conventions

This document uses the following conventions:

1. Command names, switches, and flags appear in **bold type** in syntax definitions, examples, and running text.
2. Variable information appears in *italic type*. This includes user-supplied information on command lines and the parts of prompts that differ depending on who issues the command.
3. Names of directories, files, file server machines, and partitions appear in *italic type*.
4. New terms appear in ***bold italic type***.
5. Volume names and examples of screen output and file contents appear in `typewriter type`.
6. The following symbols appear in command syntax definitions, both in the manuals and online. When issuing a command, do not type any of these symbols.
    - Square brackets [ ] surround optional items.
    - Angle brackets < > surround instances (user-supplied information).
    - A plus sign + follows an argument that accepts a list.
    - A percent sign % represents the command shell prompt. Your prompt may differ.

Command Syntax

AFS commands have the following structure:

command_suite <*operation_code*> **-switch** <*instance*>[+] **-flag**

The Command Name

Together, the **command_suite** and *operation_code* make up the ***command name***.

The **command_suite** specifies the group of related commands to which the command belongs. The name indicates which command interpreter and/or server process executes the command. AFS has several command suites, including **bos**, **fs**, **kas**, **package**, **pts**, **scout**, **uss** and **vos**. Some of these suites have an interactive mode in which the **command_suite** is omitted.

The *operation_code* tells the command interpreter/server process what action to perform. Most command suites include multiple operation codes. The *AFS Command Reference Manual* describes each operation code in detail, and the *AFS System Administrator's Guide* describes how to use them in the context of performing administrative tasks.

Note: There is also a miscellaneous group of commands that are not associated with any **command_suite**.

The Argument

Following the command name come one or more ***arguments***, each of which has two parts: a **-switch** and an *instance*. Arguments tell the command interpreter/server process which entities to act on while executing the command (for example, which file server machine, server process, or file). In general, the issuer should provide a command's arguments in the order prescribed in its *AFS Command Reference Manual* command description and the online help.

The two parts of an argument appear in this order:

- The **-switch** specifies the argument's type and is preceded by a hyphen (-). For instance, the switch **-server** usually indicates that the argument names a file server machine. Switches may usually be omitted, subject to the *Rules for Omitting Switches* presented on page xiii.
- The *instance* specifies a particular entity of the type specified by the preceding switch. For example, the proper instance for a **-server** switch is a file server machine name like *fs1.transarc.com*. Unlike switches (which have a required form), instances vary depending on what the issuer wants to accomplish. Instances appear surrounded by angle brackets (< >) in command descriptions and the online help to show that they are user-supplied variable information.

Some arguments accept lists of instances, rather than just one instance. A plus sign (+) follows a ***list-taking argument*** in the command descriptions and online help. The number and prescribed order of list-taking arguments in a command affects when it is legal to omit the switches. See the **Rules for Omitting Switches** section on page xiii.

Some commands have optional as well as required arguments; the command descriptions and online help show optional arguments in square brackets ([ ]).

The Flag

Some commands have one or more **-flag**s, which specify the manner in which the command interpreter/server process should execute the command or what kind of output it should produce. Flags are preceded by hyphens like switches, but they take no instances. Although the command descriptions and online help generally list all flags after all instances, there is no prescribed order for flags. They may appear anywhere on the command line following the operation code, except in between a switch and its instances. Flags are always optional.

An Example Command

In the example command

```
% bos getdate -server fs1.transarc.com -file ptserver kaserver -noauth
```

- **bos** is the command suite. The BOS Server executes most of the commands in this suite.
- **getdate** is the operation code. It tells the BOS Server on the specified file server machine (in this case *fs1.transarc.com*) to report the modification dates of binary files in the */usr/afs/bin* directory on its local disk.
- **-server fs1.transarc.com** is one argument, with **-server** as the switch and **fs1.transarc.com** as the instance. This argument specifies the file server machine whose BOS Server should report binary dates.
- **-file ptserver kaserver** is another argument—a list-taking one—with **-file** as the switch and **ptserver** and **kaserver** as two instances. This argument tells the BOS Server to report the modification dates on the files */usr/afs/bin/kaserver* and */usr/afs/bin/ptserver*.
- **-noauth** is a flag. It tells the BOS Server not to attempt to authenticate the issuer of the command.

Rules for Typing AFS Commands

General Rules

Type each AFS command on a single line, followed by a carriage return only at the end of the command. Some commands in this document appear broken across multiple lines, but that is for legibility reasons only (fitting them on a single line would require using very small type).

Use a space to separate each element on a command line (command suite, operation code, switch, instance, flag) from its neighbors. Spaces (not commas) also separate multiple instances of an argument.

In many cases, it is possible to reduce the typing necessary on a command line by

- Omitting the switch part of arguments.
- Using accepted abbreviations for operation codes, switches (if they are included at all), and some types of instances.

The following sections explain the conditions necessary for omitting or shortening parts of the command line. It is always legal to type a command in full, with all of its switches and no abbreviations.

Rules for Omitting Switches

It is always legal to type the switch part of an argument, but in many cases it is not necessary. Specifically, a command's switches may be omitted if the following conditions are met.

Condition 1	All of the command's required arguments appear in the order prescribed by the command description (the order is the same in the online help).
Condition 2	No switches are used on any arguments, even if they are in the correct order.
Condition 3	There is only one instance for each argument. **Note:** The important exception to this condition is if the final required argument is list-taking; in this case, it is legal to provide multiple instances for the argument.

Why Omitting Switches is Possible

Omitting switches is possible only because there is a prescribed order for the arguments in each AFS command. If no switches are provided to distinguish arguments from one another, then the command interpreter relies instead on the order of arguments; it assumes that the first element after the operation code is the command's first argument, that the next element is the command's second argument, and so on. This is why arguments must appear in the prescribed order and only one instance be given per argument when there are no switches (but note the exception discussed in the next paragraph).

The Important Exception

The important exception to the one-instance-per-argument condition (Condition 3) is when a command's final required argument is a list-taking one. In this case, the command interpreter assumes that the issuer has correctly provided one instance for each argument up through the list-taking one, so that any "extra" arguments at the end are additional instances for the list-taking argument.

An Example of Omitting Switches

Consider again the example command from page xii:

```
% bos getdate -server fs1.transarc.com -file ptserver kaserver -noauth
```

This command has two required arguments—file server machine name (identified by **-server**) and binary file name (identified by **-file**). The second argument (**-file**) is list-taking. By obeying all three conditions, the issuer can omit the switches:

```
% bos getdate fs1.transarc.com ptserver kaserver -noauth
```

Since there are no switches (**-noauth** is a flag, not a switch), the **bos** command interpreter relies on the order of arguments. It assumes that the first element following the operation code (i.e., **fs1.transarc.com**) is the file server machine name, and that the next argument (**ptserver**) is a binary file name. Then, because the command's second (and last) argument is list-taking, the command interpreter correctly interprets **kaserver** as an additional instance for the list-taking argument.

On the other hand, the following is not acceptable because it violates conditions 1 and 2, even though it meets condition 3: even though there is only one instance per argument, the arguments do not appear in the prescribed order and a switch is provided for one argument but not the other.

```
% bos getdate ptserver -server safe1.fs -noauth
```

When Omitting Switches is Illegal

The following list summarizes the rules for omitting switches from the opposite perspective, by defining when it is necessary to provide the switch for an argument. An argument's switch must be provided when

- The command's arguments do not appear in the prescribed order.
- An argument (even an optional one) that precedes this argument in the command definition is missing.
- A switch is provided for a preceding argument.
- More than one instance is supplied for a preceding argument (which must be list-taking, of course). Without a switch on the current argument, the command interpreter thinks that the current argument is simply another item in the list for the preceding argument.

Rules for Using Abbreviations and Aliases

Shortening Operation Codes (Abbreviating and Aliasing)

It is legal to abbreviate an operation code to the shortest form that still distinguishes it from the other operation codes in its suite.

Abbreviations

For example, it is legal to shorten **bos install** to **bos i** because there are no other operation codes in the **bos** command suite that begin with the letter "i". In contrast, there are several **bos** operation codes that start with the letter "s", so the abbreviations have to be longer to remain unambiguous:

bos sa	for **bos salvage**
bos seta	for **bos setauth**
bos setc	for **bos setcellname**
bos setr	for **bos setrestart**
bos sh	for **bos shutdown**
bos start	for **bos start**
bos startu	for **bos startup**
bos stat	for **bos status**
bos sto	for **bos stop**

Aliases

In addition to the acceptable, unambiguous abbreviations, some operation codes have an ***alias***, a short form that is not derived by abbreviating the operation code to its shortest unambiguous form. For example, the alias for the **fs setacl** command is **fs sa**, whereas the shortest unambiguous abbreviation is **fs seta**.

There are two usual reasons an operation code has an alias:

- Because the command is frequently issued, it is convenient to have a form shorter than the one derived by abbreviating. The **fs setacl** command is an example.
- Because the command's name has changed, but users of previous versions of AFS may be familiar with the former name. For example, **bos listhosts** has the alias **bos getcell**, its former name. Note that it is legal to abbreviate aliases of this type to their shortest unambiguous form—for instance, **bos getcell** to **bos getc**.

Even if an operation code has an alias, it is still legal to use the shortest unambiguous form. Thus, the **fs setacl** command has three acceptable forms:

fs setacl	full form
fs seta	shortest unambiguous form derived from full form
fs sa	alias

The ACCEPTABLE ABBREVIATIONS/ALIASES section of each command description in the *AFS Command Reference Manual* shows the shortest unambiguous form of the operation codes, along with alias(es) if any.

Abbreviating Switches

It is legal to shorten a switch to the shortest form that distinguishes it from the other switches for its operation code. The ACCEPTABLE ABBREVIATIONS/ALIASES section of each command description in the *AFS Command Reference Manual* shows the shortest unambiguous form for each switch.

Keep in mind that switches can often be omitted entirely, subject to the conditions listed in the **Rules for Omitting Switches** section (see page xiii).

Abbreviating File Server Machine Names

All file server machines should be assigned proper Internet-style host names (for example, *fs1.transarc.com*). However, typing full names like this on every command line can become tedious. AFS commands accept unambiguous shortened forms, but which abbreviations are acceptable depends on the domain naming service available in the local cell at the time the command is issued.

It is also legal to identify a machine by its four-part decimal Internet address rather than its name.

Abbreviating Partition Names

Partitions that house AFS volumes must have names of the form */vicep**x***, where ***x*** is a letter of the alphabet. The first such partition on a file server machine is called */vicepa*, the second */vicepb* and so on. The *AFS Installation Guide* explains how to configure and name a file server machine's partitions in preparation for storing AFS volumes on them.

When issuing AFS commands, you can abbreviate a partition name using any of the following forms:

/vicepa = vicepa = a = 0

/vicepb = vicepb = b = 1

and so on through

/vicepz = vicepz = z = 25

Abbreviating Cell Names

A cell's full name is its proper Internet domain name (such as *andrew.cmu.edu* for the Andrew cell at Carnegie Mellon University or *transarc.com* for Transarc® Corporation). Like file server machine names, cell names are often long and can become tedious to type. Unambiguous shortened forms are allowed in AFS commands, but which abbreviations are acceptable depends on the domain naming service available in the local cell at the time the command is issued.

Abbreviating Other Instances

It is not possible to abbreviate instances other than file server machine names, partition names, and cell names.

Getting Online Help on AFS Commands

You can access online help for AFS commands using the **help** and **apropos** operation codes available in most suites, and the **-help** flag available on every command. In addition, UNIX-style manual pages are available for the command suites.

The online help entry for each command consists of two or three lines:

- The first line names the command and briefly describes what it does.
- If the command has aliases, they will appear on the next line.
- The final line, which begins with `Usage:`, lists the command's arguments, in the prescribed order, and flags. Online help entries use the same symbols (brackets, etc.) as described in the **Document Conventions** section on page x.

The **help** operation code displays either the first line (short description) for every operation code in the suite, if no operation code is specified or the complete online entry (short description, alias if any, and syntax) for each specified operation code. Use it to see a complete list of a suite's operation codes or when you need to review the syntax for a certain command.

The **-help** flag displays the syntax for the command (it does not display the short description or alias).

The **apropos** operation code lists a short description of any command in a suite whose operation code or short online description includes the specified keyword. Use the **apropos** operation code when you cannot remember the exact name of the command that achieves the effect you are interested in.

Using the Help Operation Code or Flag

To list the brief descriptions of all commands in a suite, type

% **<***command_suite***> help**

without an operation code.

To display the complete help entry for one or more operation codes, type

% <*command_name*> **help** *operation_code*+

or

% <*command_name*> <*operation_code*> **-help**

Example

To see the complete online help entry for the **fs setacl** command, type the following:

```
% fs help setacl
fs setacl: set access control list
aliases: sa
Usage: fs setacl -dir <directory>+
-acl <access list entries>+ [-clear] [-negative]
```

To see only the syntax, type the following:

```
% fs setacl -help
Usage: fs setacl -dir <directory>+
-acl <access list entries>+ [-clear] [-negative]
```

Using the Apropos Operation Code

The **apropos** operation code lists the short description of any command in a suite whose name or short description contains the specified keyword string. If the keyword string includes more than one word, surround it with double quotes or other delimiters.

To display all commands that concern a certain keyword *help string*, type the following:

% <*command_name*> **apropos** *"<help string>"*

For example, suppose that a user wants to list the quota on her directory. She knows that the relevant command belongs to the **fs** suite, but cannot remember the operation code. She could use "list" as the keyword:

```
% fs apropos list
cleanacl: clean up access control list
copyacl: Copy access control list
listacl: list access control list
listcells: list configured cells
listquota: list volume quota
lsmount: list mount point
primarycell: obsolete (listed primary cell)
setacl: set access control list
whereis: list file's location
whichcell: list file's cell
wscell: list workstation's cell
```

However, using "quota" would probably be more appropriate:

```
% fs apropos quota
listquota: list volume quota
quota: show volume quota usage
setquota: set volume quota
```

Note that if she used "list quota" as a keyword, she would see a message indicating that no command name or short description contained that string:

```
% fs apropos "list quota"
Sorry, no commands found
```

Using the man Command to View Manual Pages

UNIX-style manual pages are available for the AFS command suites (**bos**, **fs**, and so on). The manual page lists all of the operation codes in the suite, describes which AFS server or client process the suite provides an interface to, and details the syntax of each operation code.

To view a man page for a command suite, type the following:

```
% man <command_suite>
```

Ordering, Support, and Training Information

Ordering

To order additional software, documentation, or training, contact Transarc File System Marketing at (412) 338-4400.

Product Support

Product support is covered under basic software maintenance and includes telephone and electronic mail support for installation problems, high priority bugs, and questions. All support calls to Transarc should originate from the Site System Administrator/Site Contact.

If you have trouble with AFS or the procedures described in this document, contact your Site System Administrator/Site Contact. If necessary, the Site System Administrator/Site Contact will contact Transarc through the usual AFS Product Support channels.

Training

Transarc offers two AFS training courses: User's Training and System Administrator's Training. Training classes are usually held at customer sites; however, a limited number of Transarc-based classes are offered. A list of the topics covered in training classes is available on request.

1. The backup Commands

This chapter defines the **backup** commands that system administrators issue when using the AFS Backup System. The chapter assumes familiarity with the concepts described in the *AFS System Administrator's Guide*.

Refer to the *Command Summary* at the end of this document for a complete list of **backup** commands and their syntax. For information on the miscellaneous (non-backup suite) commands relevant to backup (**buserver**, **butc**, and **fms**) see chapter 9.

1.1. The Information in Backup Database Entries

1.1.1. The Backup Database

A large amount of the data used by the Backup System is stored in the ***Backup Database***. There is only one Backup Database per cell; however, a copy of the Backup Database is stored on every database server machine. Synchronization of the original Backup Database and its copies is controlled by the Ubik Coordinator (see the *AFS System Administrator's Guide* for more information about Ubik).

The Backup Database contains the following information:

- The **host name**/**port offset number** pair associated with each defined Tape Coordinator/tape drive on each Tape Coordinator machine.
- The **dump levels** associated with the dump hierarchy and the **expiration date** associated with each dump level.
- The **volume sets** that can be backed up and the **volume entries** that make up each volume set.
- A record of each **dump**, the name of each tape containing that dump, each volume contained in that dump, the dump level of the dump, the expiration date of the dump, and the dump ID of the initial dump in the dump set if the dump is an appended dump or an initial dump with one or more appended dumps.

1.2. Common Arguments and Flags

Many **backup** commands accept the following optional argument and flag. They are listed in the command descriptions and are described here in detail:

[-portoffset]

This argument indicates which Tape Coordinator process is to execute the command by specifying its "port offset number." A port offset number is simply a unique identification number for a Tape Coordinator and its associated tape drive.

The Backup Database can accommodate up to 58,511 tape drives, so the legal values for this argument are the integers *0* through *58510*. If the issuer omits the argument, it defaults to *0*, so the tape drive used most often should be assigned a port offset of *0*. The issuer needs to provide this argument only when directing the command to a Tape Coordinator with a port offset other than *0*.

The Backup Database contains the port offset number of each Tape Coordinator on each Tape Coordinator machine. A Tape Coordinator machine is an AFS client machine that has an attached tape drive and an associated Tape Coordinator process (Tape Coordinator/tape drive pair). A Tape Coordinator machine can have multiple Tape Coordinator/tape drive pairs.

When a user issues a **backup** command that involves a Tape Coordinator, the Backup System sends a request to the corresponding Tape Coordinator process (**butc**). The **butc** process then looks up the specified port offset number of the Tape Coordinator in the file */usr/afs/backup/tapeconfig* on the Tape Coordinator machine. This file maps each port offset to the tape drive's UNIX device name. A Tape Coordinator process must be running on that port offset number for the request to go through.

If the issuer forgets a Tape Coordinator/tape drive's port offset, but knows the drive's device name, he or she can look in the *tapeconfig* file on the appropriate Tape Coordinator machine. (The issuer can also use the **backup listhosts** command described later in this document to list the port offset numbers of all Tape Coordinators defined in the Backup Database.)

Note: The maximum number of port offsets was set at 58,511 for the following reason. A UDP socket is defined by a 16 bit integer. Backup starts at port 7025. The number of port offsets from UDP socket 7,025 to 65,535 is (65,535 - 7,025 =) 58,510. And when you count in portoffset zero, the maximum rises to 58,511.

Although you can define up to 58,511 portoffsets for a cell, you cannot run 58,511 tape drives for dump and/or restore purposes simultaneously. Other limits are hit before that:

- The maximum number of dumps and restores that can occur simultaneously is 64.
- The **-portoffset** argument of the **backup volrestore** command can be used to specify a maximum of 128 port offsets.
- The **butc** command does not reserve UDP sockets; thus a particular socket may already be in use by another application that happens to coincide with that port offset plus 7025. If this should occur, the following error message will be generated:

```
bind: Address already in use
       rxi_GetUDPSocket: bind failed
```

[-help]

This flag has the same function as the **backup help** command: it prints a command's online help message on the screen. No other arguments or flags should be provided at the same time. If they are, this flag overrides them, and the only effect of issuing the command is that the help message appears.

1.3. The Privilege Required for backup Commands

To issue any **backup** command, the issuer must appear in the file */usr/afs/etc/UserList*. (It is assumed that a common *UserList* file is distributed to all database server and file server machines, as recommended.) See the chapter on privileged users in the *AFS System Administrator's Guide* for more information on this type of privilege.

To issue the **butc** command to start a Tape Coordinator, the issuer must appear in the file */usr/afs/etc/UserList*. The issuer also must be able to insert and write to files in the */usr/afs/backup* directory on the Tape Coordinator machine housing the Tape Coordinator. This is the directory in which the Tape Coordinator writes its log (*TL_<device_name>*) and error (*TE_<device_name>*) files. If the log and error files for the Tape Coordinator already exist, the issuer need only be able to write to them.

1.4. Running Backup Commands in a Foreign Cell

It is possible to run the **butc** command and the **backup** commands in a foreign cell. When determining what cell to execute these commands, the Backup System will look first at the value of the environment variable AFSCELL. If this variable is undefined, it will look at the cell named in */usr/afs/etc/ThisCell*. (The **butc** command also has a **-cell** argument which will be used in preference to either the AFSCELL variable or the contents of */usr/afs/etc/ThisCell* file.)

The issuer of the **butc** command and the **backup** commands must be an authenticated user listed in the foreign cell's */usr/afs/etc/UserList*.

backup adddump — define dump level in the dump hierarchy.

backup adddump -dump <*dump level name*>+ **[-expires** <*expiration date*>**]**
[-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup addd -d <*dump level name*>+ **[-e** <*expiration date*>**] [-h]**

DESCRIPTION

Creates one or more dump levels in the Backup Database and optionally assigns an absolute or relative expiration date to each dump level. All of the dump levels in the Backup Database collectively constitute the dump hierarchy.

To create a full dump level, specify the *dump level name* preceded by a slash (for example, */full*). To create an incremental dump level, you must first define all of the parent dump levels above that incremental level. For example, before defining the incremental dump level */full/incremental1/incremental2*, you must define the dump levels */full* and */full/incremental1*. Be sure to separate each dump level with a slash.

The maximum length for any single level (the text between slashes) in a dump level name is 28 characters. This does not include the slash that precedes the level.

Dump level names can have any number of levels, but they can be no longer than 256 characters (this includes any slashes). You should not use periods in dump level names because dump names are created by joining the volume set and dump level name with a period. If you use metacharacters in a dump level name, when working outside interactive mode, enclose the instance within double quotes or escape the metacharacter with a backslash.

Use the **-expires** argument to associates an expiration date with each specified *dump level name*. The expiration date is applied to each dump created at the corresponding dump level.

Expiration dates may be either absolute or relative:

- Absolute expiration dates are defined as a specific month/day/year and, optionally, hour and minutes. A dump with an absolute expiration date expires at that date and time, regardless of when the dump was created. (If the expiration date predates the dump creation, the dump is immediately treated as expired.)
- Relative dates are defined as a number of years, months, and/or days. When the Backup System creates a dump at the dump level, it calculates the dump's actual expiration date by adding the relative date to the start time of the dump operation. Thus, dumps made at a dump level with a relative expiration date expire at the specified number of years, months, and/or days after the creation of the dump. Each dump expires at the time of day at which it was started.

If no expiration date is defined for a dump level, the dump is considered to be immediately expired. However, it is strongly recommended that you associate expiration dates with every dump level.

ARGUMENTS

-dump names each dump level to be added to the dump hierarchy.

-expires defines the expiration date to be associated with each dump level created with the command. Absolute expiration dates have the format

[at] <*mm*/*dd*/*yy* **[***hh***:***mm***]>**

or

[at] NEVER

where the optional word **at** is followed by a date (month/day/year) and, optionally, a time (hours:minutes) or by the key word NEVER (which specifies that dumps at that dump level never expire). If provided, the time should be in 24-hour format (for example, *20:30* is 8:30 p.m.); if omitted, the time defaults to 12:00 a.m. (00:00). Valid values for *yy* run from *00* to *37*, which are interpreted as the years 2000 through 2037, and from *70* to *99*, which are interpreted as 1970 through 1999. (Values between *38* and *69* are not valid, because the years to which they correspond, 2038 through 2069, exceed the capacity of the standard UNIX representation of dates. Such values are reduced to 2038.)

Relative expiration dates have the format

[in] [<*integer*>**y] [**<*integer*>**m] [**<*integer*>**d]**

where the optional word **in** is followed by a number of years (maximum 9999), months (maximum 12), and/or days (maximum 31). At least one of the three must be provided, and the units abbreviations (**y**, **m**, or **d**) must always be used. If more than one of the three is provided, they must appear in the specified order. See the EXAMPLES section for illustrations of acceptable values. As with absolute dates, any dates later than the year 2038 are reduced to 2038.

Note: It is not possible to specify more than one expiration date even though the usage message for this command indicates that it can. A plus sign follows this argument because a multi-word argument is required to define an expiration date, not because multiple dates can be specified. An expiration date provided with the **-expires** argument applies to each dump level specified with the **-dump** argument.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following defines a full dump called */1994* with a relative expiration date of one year:

% backup addd /1994 in 1y

The following defines an incremental dump called */sunday1/monday1* with a relative expiration date of 13 days:

% backup addd /sunday1/monday1 in 13d

The following defines two dump levels, */Monthly/Week1* and */Monthly/Week2*, both incremental from the same parent, */Monthly*. They are both defined to expire at 12:00 a.m. on 1 January 1994.

% backup adddump -d /Monthly/Week1 /Monthly/Week2 -e at 01/01/94

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup deldump

backup listdumps

backup setexp

backup addhost — add Tape Coordinator entry to the Backup Database.

backup addhost -tapehost <*tape machine name*> **[-portoffset** <*TC port offset*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup addh -t <*tape machine name*> **[-p** <*TC port offset*>**] [-h]**

DESCRIPTION

Creates an entry in the Backup Database that defines a Tape Coordinator. The entry specifies

- The Tape Coordinator machine for which the Tape Coordinator is being defined (with **-tapehost**).
- The Tape Coordinator's port offset number (with **-portoffset**). The mapping between a Tape Coordinator's port offset number and its UNIX device name is recorded in */usr/afs/backup/tapeconfig* on the Tape Coordinator machine. Remember to alter that file when you issue this command. See the chapter on the AFS Backup System in the *AFS System Administrator's Guide* for instructions.

WARNING

Do not add multiple Tape Coordinators with the same port offset to the Backup Database. Only the Tape Coordinator associated with the first occurrence of the port offset can be used. Before issuing this command, use the **backup listhosts** command to determine the Tape Coordinators currently defined in the Backup Database. The command does prevent you from creating a host entry with the same Tape Coordinator machine and port offset pair as a previous entry.

ARGUMENTS

-tapehost	specifies the Tape Coordinator machine to which the Tape Coordinator is to be added (the machine to which the associated tape drive is attached). The form of the name given here affects how it appears in the output from **backup listhosts**. The machine to be added must have an entry in */etc/hosts* or the cell's naming service.
-portoffset	specifies the Tape Coordinator's port offset number. If not provided, it defaults to *0*. If provided, it should be one of the integers *0* through *58510*, and it should match the port offset number assigned to this tape drive/Tape Coordinator in */usr/afs/backup/tapeconfig* file on the corresponding Tape Coordinator machine.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following creates an entry in the Backup Database that allocates port offset number *4* to a Tape Coordinator/tape drive on *backup1.transarc.com*:

% backup addh backup1.transarc.com 4

The following creates an entry in the Backup Database that allocates port offset number *0* to a tape drive on *backup3.transarc.com*:

% backup addh backup3.transarc.com

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup delhost

backup listhosts

backup addvolentry — define volume entry in a volume set.

backup addvolentry **-name** <*volume set name*> **-server** <*machine name*>
-partition <*partition name*>
-volumes <*volume name (regular expression)*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup addvole **-n** <*volume set name*> **-s** <*machine name*> **-p** <*partition name*>
-v <*volume name (regular expression)*> **[-h]**

DESCRIPTION

Adds a volume entry definition to the volume set specified with **-name**. The volume set must already have been created with **backup addvolset**.

A volume entry definition can specify one or more volumes, depending on how the issuer combines the **-server**, **-partition**, and **-volumes** arguments. For the **-server** and **-partition** arguments, the issuer can provide either a single, specific value or the metacharacter expression ".*" (a period and asterisk). This metacharacter expression matches any string, so the effect is to match every machine name or partition name found in the VLDB.

For the **-volumes** argument, the issuer may specify any combination of regular characters and metacharacters (not just the .* expression) to wild card part or all of the volume name. The METACHARACTER NOTATION section below describes the acceptable metacharacters.

METACHARACTER NOTATION

For the **-server** and **-partition** arguments, the only acceptable metacharacter expression is ".*" (the period and asterisk).

For the **-volumes** argument, any combination of regular characters and metacharacters can be combined in order to specify groups of volumes. The following metacharacters are acceptable:

. The period matches any single character.

* The asterisk matches any number (0 or more) of the preceding character. The asterisk may be combined with any other metacharacter or regular character.

[ ] Square brackets around a list of characters match a single instance of any of the characters, but no other characters; for example, *[abc]* matches *a* or *b* or *c*, but not *d* or *A*. This expression may be combined with the asterisk.

^ The caret, when used as the first character in a square-bracketed set, indicates a match with any single character except the characters that follow it; for example, *[^a]* matches any single character except lowercase *a*. This expression may be combined with the asterisk.

\ The backslash may precede any of the characters in this list so that they match their literal value only. For example, the expression \. matches a single period, * a single asterisk, and \\ a single

backslash. Such expressions may be combined with the asterisk (\.* matches any number of periods).

Perhaps the most common metacharacter expression is the period followed by an asterisk (.*). This expression matches any string of any length. The period matches any character, and the asterisk means any number of that character. As mentioned above, it is the only metacharacter expression that can be used with the **-server** and **-partition** arguments. In the volume field it may stand alone (in which case it matches every volume the Backup System finds listed in the VLDB), or it may be combined with regular characters. In the following example, the combination of letters and a metacharacters matches any string that begins with *user* and ends with *backup*:

*user.*backup*

WARNING

When issuing this command outside of interactive mode, it is necessary to enclose any strings containing metacharacters with double quotes or escape the metacharacters with other delimiters. Otherwise, the shell attempts to resolve the metacharacters.

ARGUMENTS

-name	names the volume set to which this volume entry definition is to be added. The volume set must already have been created with **backup addvolset**.
-server	indicates the file server machine(s) that house the volumes in the volume entry. Legal values are a specific machine name or the .* metacharacter expression which matches all machine names.
-partition	indicates the partition(s) that house the volumes in the volume entry. Legal values are a specific partition name or the .* metacharacter expression, which matches all machine names.
-volumes	indicates the volume(s) included in the volume entry. This argument may include any combination of regular characters and metacharacters (the latter are explained in the METACHARACTER NOTATION section).
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following adds a volume entry to the volume set called *sys* that includes all volumes in the system that begin with "*sun4c_52.*":

% backup addvole sys ".*" ".*" "sun4c_52\..*"

The following adds a volume entry to the volume set called *fs2* that includes all volumes on the */vicepb* partition of file server machine *fs2.transarc.com*:

% backup addvole fs2 fs2.transarc.com /vicepb ".*"

The chapter on the AFS Backup System in the *AFS System Administrator's Guide* presents additional examples as well as advice on the proper way to group volumes.

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup addvolset

backup delvolentry

backup delvolset

backup listvolsets

backup addvolset — create a new (empty) volume set in Backup Database.

backup addvolset -name <*volume set name*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup addvols -n <*volume set name*> **[-h]**

DESCRIPTION

Creates a new volume set in the Backup Database specified with **-name**. The name should give some indication of the volume set's contents; for example, *user* for the volume set that includes all user volumes in the file system. The volume set name must be unique within the Backup Database of the local cell.

After issuing this command, the user needs to issue **backup addvolentry** to define the volume entries included in the volume set.

ARGUMENTS

-name	names the new volume set. The name can include no more than 31 characters and should not contain periods.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following creates a volume set called *sys*:

% backup addvols sys

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup addvolentry

backup delvolset

backup delvolentry

backup listvolsets

backup apropos — show each help entry containing keyword.

backup apropos -topic <*help string*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup ap -t <*help string*> **[-h]**

DESCRIPTION

Displays the first line of the help entry for any **backup** command that has *help string* in its name or short description.

ARGUMENTS

-topic	specifies the keyword string for which to search. If the string contains spaces, surround it with double quotes or other delimiters. Type each *help string* in lowercase letters.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The first line of the online help entry available for each **backup** command. To see the remaining lines in a help entry, which provide the command's alias (if any) and syntax, use the **backup help** command.

EXAMPLES

The following lists all **backup** commands that have the word "*tape*" in their operation codes or short descriptions:

% backup ap tape

```
labeltape: label a tape
readlabel: read the label on tape
scantape: dump information recovery from tape
status: get tape coordinator status
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

backup help

backup dbverify — check the status of the Backup Database.

backup dbverify [-detail] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup db [-d] [-h]

DESCRIPTION

Checks the status of the Backup Database. The command displays a message indicating whether the Backup Database is undamaged or damaged. If the Backup Database is undamaged, it can be accessed.

WARNING

You should only issue this command when the corresponding database is not being used.

ARGUMENTS

-detail	causes additional information about the Backup Database to be displayed.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

Depending on the condition of the Backup Database, the command displays one of the following two messages:

- "`Database OK`". This message indicates that the database is undamaged and can be used.
- "`Database not OK`". This message indicates that the database is damaged. The database must be deleted and then restored from tape using the **backup restoredb** command (provided it has previously been backed up to tape with the **backup savedb** command).

If the **-detail** flag is included with the command, additional information about the Backup Database is provided. One good reason to use the **-detail** flag is to determine if your Backup Database has any ***orphan blocks***, which are blocks that the Backup Database preallocated but cannot use. Orphan blocks are not a problem for the database. However, if you are concerned with disk usage on the partition on which the database resides, you can eliminate the unusable blocks by saving the database to tape with **backup savedb** and then restoring it with **backup restoredb**.

Using the **-detail** flag also causes the command to report any additional inconsistencies it finds in the database. For example, the command reports any invalid hostnames for Tape Coordinator machines it finds in the database.

The name of the database server machine which ran a database verification is provided when the **-detail** flag is specified. If the database is damaged, more details on the failure can be found in the */usr/afs/logs/BackupLog* file on that machine.

EXAMPLES

The following verifies that the Backup Database is undamaged:

% **backup db**

```
Database OK
```

The following verifies that the Backup Database is undamaged and that it has no orphan blocks. The database server machine *brick.transarc.com* made the check.

% **backup dbverify -detail**

```
Database OK
Orphan blocks 0
Database checker was brick.transarc.com
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup restoredb

backup savedb

backup deldump — delete a dump level from Backup Database.

backup deldump -dump <*dump level name*> [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup deld -d <*dump level name*> **[-h]**

DESCRIPTION

Deletes the indicated dump level and all of its child dump levels from the dump hierarchy in the Backup Database. Use the **backup listdumps** command to display the dump hierarchy in the Backup Database.

ARGUMENTS

-dump	specifies the complete pathname of the dump level to be deleted.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following deletes the dump level called */sunday1/monday1* from the dump hierarchy. Any "child" dump levels of */sunday1/monday1* are also deleted.

```
% backup deld /sunday1/monday1
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup adddump

backup listdumps

backup deletedump — delete one or more records from Backup Database.

backup deletedump [-dumpid <*dump id*>+**] [-from** <*date time*>**]**
[-to <*date time*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup dele [-d <*dump id*>+**] [-f** <*date time*>**] [-t** <*date time*>**] [-help]**

DESCRIPTION

Deletes the record(s) from the Backup Database associated with either each dump ID specified with the **-dumpid** argument or a range of dump IDs created within a specified time range identified using either the **-to** argument or the **-from** and **-to** arguments. Only initial dump records can be directly deleted using this command; however, when an initial dump record is deleted, all records of appended dumps in the associated dump set are also deleted from the Backup Database.

This command may be used to remove dump records that

- Contain incorrect information (possibly because a dump operation was interrupted or failed).
- Contain information associated with a tape that is being discarded.
- Correspond to a Backup Database dump. Using the **backup deletedump** command (with the **-dumpid** argument) is the only way to remove a record of a Backup Database dump.

WARNING

Database dump records cannot be deleted with the **-from** and **-to** arguments. They must be explicitly deleted with the **-dumpid** argument.

If a dump set contains dumps that were created between the **-from** and **-to** time range but also contains at least one dump that falls outside of this range, the dump set will not be deleted.

ARGUMENTS

-dumpid specifies the dump ID of each dump to be deleted from the Backup Database. Each dump ID must be associated with an initial dump; appended dumps cannot be directly deleted. Using this argument is the only way to delete Backup Database dump records.

-from specifies the bottom end of the date range for the group of dump IDs that you want to delete from the Backup Database. If you use the **-to** argument without this argument, the from date and time will default to *0* (the UNIX equivalent of January 1, 1970).

The instance must have the following format: *mm/dd/yy hh:MM*, where *mm/dd/yy* indicates the month (mm), day (dd), and year (yy) and *hh:MM* indicates the hour (hh) and minutes (MM) in 24-hour format. The date is required. The time is optional; if it is excluded, the time defaults to *12:00:59 am*.

-to specifies the top end of the date range for the group of dump IDs that you want to delete from the Backup Database. If you use the **-to** argument without the **-from** argument, the from date and time will default to *0* (the UNIX equivalent of January 1, 1970).

The instance must have the following format: *mm/dd/yy hh:MM*, where *mm/dd/yy* indicates the month (mm), day (dd), and year (yy) and *hh:MM* indicates the hour (hh) and minutes (MM) in 24-hour format. The date is required. The time is optional; if it is excluded, the time defaults to *12:00:59 am*.

Note: You can also use **NOW** for this instance to specify the current date and time. Using **NOW** without the **-to** argument will delete all records in the Backup Database except for Backup Database dump records.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The output from the command lists the dump IDs of all deleted dump records:

```
The following dumps were deleted:
     741366407
     741366616
     741366676
```

EXAMPLES

The following deletes the dump record associated with dump ID *653777462* from the Backup Database as well as the records of any dumps appended to *653777462*:

% backup dele 653777462

The following deletes all dump records created between 12:00 am, January 1, 1991 and 12:00:59 am, December 31, 1991 from the Backup Database:

% backup dele -f 01/01/91 -t 12/31/91

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup dumpinfo

backup delhost — delete a Tape Coordinator entry from the Backup Database.

backup delhost -tapehost <*tape machine name*> **[-portoffset** <*TC port offset*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup delh -t <*tape machine name*> **[-p** <*TC port offset*>**] [-help]**

DESCRIPTION

Deletes the indicated Tape Coordinator entry from the Backup Database. Backup requests are no longer sent to that Tape Coordinator, even if it is still running on the machine. You should also remove the corresponding line from the */usr/afs/backup/tapeconfig* on the appropriate Tape Coordinator machine.

The **backup listhosts** command can be used to list the Tape Coordinator machines and their respective Tape Coordinators currently defined in the Backup Database.

ARGUMENTS

-tapehost	specifies the Tape Coordinator machine housing the Tape Coordinator to be deleted.
-portoffset	specifies the port offset number of the Tape Coordinator to be deleted. If omitted, it defaults to *0*. If provided, it should be one of the integers *0* through *58510*, and it should match the port offset number assigned to the Tape Coordinator's tape drive in */usr/afs/backup/tapeconfig* on the Tape Coordinator machine indicated with **-tapehost**. Section 1.2 has more information on port offset numbers.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following deletes the entry for the Tape Coordinator with port offset *2* on the Tape Coordinator machine *backup3.transarc.com* from the Backup Database:

% backup delh backup3.transarc.com 2

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup addhost

backup listhosts

backup delvolentry — delete volume entry from volume set.

backup delvolentry **-name** <*volume set name*> **-entry** <*volume set index*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup delvole **-n** <*volume set name*> **-e** <*volume set index*> **[-h]**

DESCRIPTION

Deletes the indicated volume entry from the volume set specified with **-name**. Use **-entry** to identify the volume entry by its index number. The **backup listvolsets** command can be used to display the index number of each volume entry in a volume set.

ARGUMENTS

-name	names the volume set from which the volume entry is to be deleted.
-entry	identifies the index number of the volume entry to be deleted. The **backup listvolsets** command can be used to display the index number of each volume entry in a volume set (the first entry defined in a volume set has index *1*, the second *2*, and so on).
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following deletes the fourth volume entry from the volume set called *sys*:

% backup delvole sys 4

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup addvolentry

backup addvolset

backup delvolset

backup listvolsets

backup delvolset — delete one or more volume sets from Backup Database.

backup delvolset -name <*volume set name*>+ **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup delvols -n <*volume set name*>+ **[-h]**

DESCRIPTION

Deletes each specified volume set (and the volume entries each contains) from the Backup Database. The **backup listvolsets** command can be used to list the volume sets (and their volume entries) currently defined in the Backup Database.

ARGUMENTS

-name	specifies each volume set to be deleted.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following deletes the volume set called *user* and all volume entries it contains:

% backup delvols user

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup addvolentry

backup addvolset

backup delvolentry

backup listvolsets

backup diskrestore — restore entire contents of partition.

backup diskrestore **-server** <*machine to restore*>
-partition <*partition to restore*>
[-portoffset <*TC port offset*>**]**
[-newserver <*destination machine*>**]**
[-newpartition <*destination partition*>**] [-n] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup di -s <*machine to restore*> **-pa** <*partition to restore*>
[-po <*TC port offset*>**] [-news** <*destination machine*>**]**
[-newp <*destination partition*>**] [-n] [-h]**

DESCRIPTION

Restores the contents of the partition specified with **-server** and **-partition** to the file system. To do this, the command interpreter contacts the VL Server for a listing from the VLDB of all the volumes that resided on the specified partition. It then consults the Backup Database to learn which tapes contain the full and incremental dumps needed to restore every volume. This command is useful if a disk or machine crash destroys the data on an entire partition.

By default, the partition's contents are restored to the same file server machine and partition (the machine and partition specified with **-server** and **-partition**). The issuer can restore the partition's contents to an alternate site if desired. To restore the contents to

- A different partition on the same file server machine, specify the new partition with **-newpartition**.
- The partition of the same name on a different file server machine, specify the new file server machine with **-newserver**.
- A completely different site, specify the new file server machine with **-newserver** and the new partition with **-newpartition**.

The creation date of each restored volume is set to the date and time at which it is restored with this command.

It is possible to have the command interpreter print out a list of the tapes the Backup System needs to perform the indicated restore, without actually performing the restore. To get this list, include the **-n** flag along with all of the other flags and arguments to be used on the actual command.

By default, when you execute this command, the **butc** process will prompt you to insert the first tape in the correct drive and press **Return**. However, if the **butc** process is running with the **-noautoquery** flag, you will not be prompted in this manner. Instead, you are expected to insert the appropriate tape in the drive prior to executing this command; if you do not, you will be prompted to insert a tape via an error message on the Tape Coordinator window. Of course, if the restore requires multiple tapes, operator intervention will be required to make tape changes.

Use the **backup volrestore** command to restore one or more volumes.

WARNING

If you believe that a disk has crashed or is otherwise damaged, do not issue the **vos syncserv** command before issuing this command.

All of the dump sets needed to complete a **backup diskrestore** operation must be on the same type of tape or must be on tapes readable by a single tape drive because a single Tape Coordinator must perform the entire operation. If the dumps required to complete the restore operation are on different types of tapes, the restore cannot be completed. Therefore, it is recommended that all volume sets containing volumes from the same partition be dumped to the same type of tape.

ARGUMENTS

-server	names the file server machine housing the partition to be restored.
-partition	names the partition to be restored.
-portoffset	specifies the port offset number of the Tape Coordinator/tape drive that you intend to use. See section 1.2 for more details.
-newserver	names an alternate file server machine to which to restore the data. If this argument is omitted, the data is restored to the file server machine on which it formerly resided (**-server**).
-newpartition	names an alternate partition to which to restore the data. If this argument is omitted, the data is restored to the partition with the same name as the one on which it formerly resided (**-partition**).
-n	instructs the command interpreter to print out the list of tapes necessary to perform the indicated restore operation, without actually performing the operation.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

If a tape error occurs during the restore operation, the Tape Coordinator displays the following messages:

```
"Restore operation on volume name failed due to tape error"
"Do you want to continue (y/n) ?"
```

where *name* is the name of the volume that was being restored when the tape error occurred. The issuer can choose to continue, realizing the volume indicated by *name* is not restored from the tape, or cancel and attempt to determine the source of the tape error.

If the issuer includes the **-n** flag with the command, a "`Tapes needed:`" header appears, followed by a list of the tapes necessary to complete the restore operation.

EXAMPLES

The following restores the volumes residing on the */vicepd* partition of *fs5.transarc.com* to that partition. The issuer uses the Tape Coordinator/tape drive associated with port offset *3*.

% backup di fs5.transarc.com /vicepd 3

The following restores the volumes residing on the */vicepb* partition of *fs1.transarc.com* to a new site: */vicepa* on *fs3.transarc.com*. The issuer uses the Tape Coordinator/tape drive associated with port offset *0*.

% backup di fs1.transarc.com /vicepb -news fs3.transarc.com -newp /vicepa

The following shows the list of tapes that would be required to restore the volumes from */vicepm* of *fs4.transarc.com*:

% backup di fs4.transarc.com /vicepm -n

```
Tapes needed:
user.sunday1.1
user.sunday1.2
user.monday1.1
user.tuesday1.1
user.wednesday1.1
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup dump

backup volrestore

butc

backup dump — dump a specified volume set at a particular dump level.

backup dump -volumeset <*volume set name*> **-dump** <*dump level name*>
[-portoffset <*TC port offset*>**] [-at** <*Date/time to start dump*>**]**
[-append] [-n] [-file <*load file*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup dump -v <*volume set name*> **-d** <*dump level name*> **[-p** <*TC port offset*>**]**
[-at <*Date/time to start dump*>$^{+}$**] [-ap] [-n] [-f** <*load file*>**] [-h]**

DESCRIPTION

Dumps the volume set specified with **-volumeset** at the dump level specified with **-dump** and records the dump in the Backup Database. A dump set is created when a volume set is dumped onto a new tape or onto a tape containing another dump set that is being recycled. The first dump in a dump set is called the initial dump; all dumps appended to the dump set are called appended dumps.

To place a new dump set on a tape containing an old dump set (referred to as recycling), two preconditions must be met:

- The name of the initial dump of the new dump set must match the name of the initial dump of the old dump set. In other words, the name on the tape must match the name of the initial dump.
- Each dump associated with the old dump set must be expired.

Once a dump set is created, additional dumps may be appended to it. To append a dump to an existing dump set, use the **-append** flag. To append a dump to an existing dump set, the following conditions must be met:

- The tape name must be recorded in the Backup Database.
- The tape must be the last tape in the dump set.
- The last dump on the tape must have completed successfully.
- The tape must have been created under the 3.3 version of AFS.

Note: The ability to append dumps to an existing dump set is a new feature of AFS 3.3 that is completely optional. If desired, you may continue to restrict backups to one dump per dump set.

The number of tapes in a dump set is theoretically unlimited; however, there are two good reasons to keep your dump sets small (that is, less than five tapes to a dump set for an average cell):

- If a bad spot develops on one of the tapes in a dump set, the subsequent data on that tape and the subsequent tapes in the dump set cannot be scanned to reconstruct the Backup Database (using the **backup scantape** command). Data can still be restored from these tapes if the information is properly stored in the database.

- If you archive backup data on an as-needed basis, the entire dump set containing the relevant data must be archived rather than just the tapes containing that data. If your dump sets are large, you will be forced to archive a large number of tapes containing irrelevant data in order to properly archive the relevant data. This wastes both tape and archive storage space. (See the *AFS System Administrator's Guide* for more information on archiving.)

Each tape in a dump set is named using parameters from the initial dump and the numerical position of the tape in the dump set, as follows:

<volume set name>.<dump level name>.<tape index>

The *volume set name* is the name of the volume set from the original dump. The *dump level name* is the final element in the dump level name; for example, *monday1* in the dump level */sunday1/monday1*. The *tape index* represents the order of the tape in the dump set; for example, the first tape in a dump set will be labeled *1*, the second tape in a dump set will be labeled *2*, etc. Thus, the first tape in a dump set created by dumping the volume set *user* at the dump level */sunday2/monday* would be named as follows:

user.monday.1

A volume set can either be dumped at a full or incremental level. A full dump is a complete copy of all of the data in the volumes of the specified volume set.

An incremental dump is a copy of all data in the volume set that has changed since the volume set was dumped at the incremental level's parent dump level. This is determined by comparing the dump time of the parent dump with the modification time stamps on the file and directory elements in the volumes of the volume set. For example, if **-dump** is */full/week1/thursday*, the dump includes only those files that have changed since the volume set was dumped at the */full/week1* level.

You must dump a volume set at its parent dump level before dumping it at a subsequent incremental dump level. If the command interpreter cannot locate a dump created at the parent dump level, it looks in the Backup Database for a dump created at one level higher in the hierarchy, and so on up to the full dump level. If it fails to find a dump at the full dump level, it dumps the volume set at the full dump level; if it finds a dump at a certain point in the hierarchy, it dumps the volume set at the next lower incremental dump level.

If the Backup System is unable to access a volume in the volume set definition during a dump (perhaps because of a file server machine or Volume Server outage), it skips the volume and dumps the remaining volumes in the volume set. It then attempts to dump each volume it skipped on the first pass. If it still cannot dump a volume, it queries the operator as to whether it should again attempt to dump the volume, omit the volume from the dump, or abort the dump altogether. When prompted, the operator can attempt to solve whatever problem prevented the volumes from being accessed.

If the Tape Coordinator performing the dump was initialized at debug level *1*, its window reports the failure to include the volume in the dump set. The Tape Coordinator's log and error files always report the omission of any volume from a dump set.

The Backup System does not require that a volume fit entirely on a single tape. If the Tape Coordinator reaches the end of a tape in the middle of dumping a volume, it puts the remaining data on the next tape. The Backup Database automatically records that the volume resides on multiple tapes.

It is possible to direct one or more **backup dump** commands to execute at specified times in the future. These types of dumps are referred to as timed or scheduled dumps. There are two ways to set up timed dumps:

- You can instruct a single **backup dump** command to execute at a specified time by executing the **backup dump** command with the **-at** argument.
- You can instruct one or more **backup dump** commands to execute at a specified time by executing the **backup dump** command with the **-file** and **-at** arguments.

Before a timed dump is executed, it is first checked for validity. The following validity checks are made:

- The volume set must exist within the Backup Database.
- The dump level must exist within the Backup Database.
- The port offset must be valid.
- The specified date and time for command execution must be in the future.

Valid tokens must be available for use by a timed dump. If no valid tokens are available at the time of or during execution of a timed dump, the operation will fail.

If the command is valid, it is assigned a job number and added to the jobs list. These jobs may be listed with the **(backup) jobs** command and killed with the **(backup) kill** command.

Note: Because this feature requires unexpired tokens and backup tapes to be left unattended, you may want to institute additional physical security measures to protect against improper access.

By default, when this command is executed, the **butc** process will prompt you to insert the appropriate tape in the correct drive and press **Return**. However, if that **butc** process was executed using the **-noautoquery** flag, you will not be prompted in this manner. Instead, you are expected to insert the tape in the drive prior to command execution; if you do not insert the appropriate tape, you will be prompted to insert a tape via an error message on the Tape Coordinator window. Of course, if the dump requires multiple tapes, operator intervention will be required to make tape changes.

WARNING

If a dump operation is interrupted or fails for any reason, all volumes written to tape *before* the interrupt are complete and usable on the tape. An entry will appear in the Backup Database for the aborted dump and for the volumes that were successfully dumped. See the *AFS System Administrator's Guide* for information on dealing with interrupted dumps.

Also, although a **backup volrestore** operation can be used to restore a volume that was dumped to different types of tapes, a **backup diskrestore** operation cannot. Because only a single Tape Coordinator can be used to perform a **backup diskrestore** command, all of the tapes to be accessed during the

operation must be of the same type (making them readable by a single tape drive). Therefore, it is recommended that all volume sets containing volumes from the same partition be dumped to the same type of tape.

ARGUMENTS

-volumeset names the volume set (already defined in the Backup Database using **backup addvolset** and **backup addvolentry**) to be dumped.

-dump names the dump level (already defined in the Backup Database using **backup adddump**) to be used in dumping the volume set. Provide the dump level's complete pathname. This argument determines whether the dump is full or incremental and, in the latter case, which dump level serves as the parent for this dump.

-portoffset specifies the port offset number of the Tape Coordinator/tape drive that you intend to use. See section 1.2 for more details.

-at specifies a date and time in the future at which the **backup dump** command will executed. The instance must have the following format: *mm/dd/yy hh:MM*, where *mm/dd/yy* indicates the month (mm), day (dd), and year (yy) and *hh:MM* indicates the hour (hh) and minutes (MM) in 24-hour format. The date is required. The time is optional; if it is excluded, the time defaults to *12:00 am*.

For example, *12/23/93 20:20* indicates that the process should execute at 8:20 pm on December 23, 1993.

-append specifies that the dump should be appended onto the dump set associated with the target tape. If the tape is not associated with a dump set, a new dump set will be created using the parameters of this dump. If the tape is not the last tape in the dump set, you will be prompted to insert the appropriate tape.

-n indicates that the command interpreter is to display the volumes that would be included in the indicated dump, without actually performing the dump operation.

-file specifies the pathname of the file containing **backup dump** commands. You can specify the full pathname of the file or a partial pathname if the path includes the current directory.

The format of the individual lines in the *load file* are the same as the syntax of the **backup dump** command except the word **backup** should not be included. Each command must be placed on a single line.

Below is an example of a **backup dump** file:

```
dump user /sunday1/wednesday 1 02/05/94
dump rt /sunday1/wednesday 2 02/05/94
dump sun /sunday1/wednesday 2 02/05/94 02:00 -append
```

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The following header appears:

```
Preparing to dump the following volumes:
```

The header is followed by a list of volumes, each identified by name and volume ID number, to be included in the dump. This list is the result of the Backup System matching any metacharacters it finds in the volume set's entries against the information in the VLDB. The following message then appears, indicating that the Backup System has passed the dump request onto the indicated Tape Coordinator:

```
Starting dump.
```

If the issuer includes the **-n** flag, the output is of the form

```
Starting dump of volume set 'volume set' (dump set 'dump level')
Total number of volumes : number
Would have dumped the following volumes:
list of volumes
```

where *list of volumes* is the list of volumes, identified by name and volume ID number, that would be included in the actual dump set were the **-n** flag omitted.

EXAMPLES

The following dumps the volumes in the volume set called *user* at the dump level */full/sunday2/monday*. The issuer places the necessary tapes in the drive with port offset *5*.

% backup dump user /full/sunday2/monday 5

```
Preparing to dump the following volumes:
user.jones.backup   387623900
user.pat.backup     486219245
user.smith.backup   597315841
       .                .
       .                .
Starting dump.
```

The following displays the list of volumes to be dumped when the user dumps the *sys.sun* volume set according to the */full* dump level.

% backup dump sys.sun /full -n

```
Starting dump of volume set 'sys.sun' (dump set '/full')
Total number of volumes: 24
Would have dumped the following volumes:
sun3x_411      124857238
sun3x_411.bin  124857241
    .             .
    .             .
sun4_411       124857997
    .             .
    .             .
```

The following creates a timed dump of the volumes in the volume set *user* at the dump level */sunday2/monday1*. The job will execute automatically at 11:00 pm on January 5, 1994. The issuer uses port offset *0*.

% backup dump user /sunday2/monday1 -at 01/05/94 23:00

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup adddump

backup addvolentry

backup addvolset

backup diskrestore

backup labeltape

backup volrestore

butc

backup dumpinfo — list information about specified backups.

backup dumpinfo [-ndumps <*no. of dumps*>**] [-id** <*dump id*>**] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup dumpi [-n <*no. of dumps*>**] [-i** <*dump id*>**] [-v] [-h]**

DESCRIPTION

Lists information about the specified dumps taken from the Backup Database. If a number is specified with **-ndumps**, information about the most recent *no. of dumps* number of dumps is displayed. If a specific dump ID number is specified with **-id**, information about that dump is displayed. If both **-ndumps** and **-id** are omitted, information about the last 10 dumps is displayed. If both the **-id** argument and the **-verbose** flag are specified, more detailed information about the specified dump is displayed.

ARGUMENTS

-ndumps	specifies the number of dumps about which information is to be displayed; information about the most recent *no. of dumps* number of dumps is displayed. If fewer than the specified number of dumps exist, information is displayed about all existing dumps. Use **-ndumps** or use **-id**; omit both arguments to list information about the last 10 dumps.
-id	specifies the unique dump ID number of a specific dump about which information is to be displayed. Use **-id** or use **-ndumps**; omit both arguments to list information about the last 10 dumps. Be sure to include the **-id** switch; otherwise, the command will interpret the *dump ID* to be the *no. of dumps* instance of the **-ndumps** argument.
-verbose	instructs the command to provide detailed information about the dump specified with **-id**. This argument can be used only with **-id**.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

If **-ndumps** is specified, the following information is provided about each displayed dump:

- `dumpid`: The dump ID number.
- `parentid`: The dump ID number of the dump's parent dump. A value of `0` indicates this is a full dump and therefore has no parent.
- `lv`: The depth in the dump hierarchy of the dump level used to create the dump. A value of `0` indicates a full dump, in which case `parentid` should also be `0`. A value of `1` or greater indicates an incremental dump made at the specified level in the dump hierarchy.
- `created`: The date and time at which the Backup System started the dump operation that created the dump.

- `nt`: The number of tapes required to record the information in the dump. A value of `0` indicates a dump operation that was aborted or that failed, but for which an entry still exists in the database.
- `nvols`: The number of volumes from which information was included in the dump. A value of `0` indicates a dump operation that was aborted or that failed, but for which an entry persists in the database; `nt` should also be `0`. If a volume spans tapes, it is counted twice.
- `dump name`: The dump name in the form

  <*volume set name*>.<*dump level name*> (*initial dump ID*)

  where *volume set name* is the name of the dumped volume set and *dump level name* is the name of the last element in the dump level at which the volume set was dumped.

  The *initial dump ID*, if one is displayed, contains the dump ID of the initial dump in the dump set and tells you that that dump set contains two or more dumps. If there is no *initial dump ID*, the dump is the initial dump in a dump set that contains no appended dumps.

If just the **-id** is specified, the command displays the following information:

- The first line, labeled `Dump:`, displays the following information about the specified tape:
  - The dump ID number.
  - The depth in the dump hierarchy of the dump level used to create the dump. A value of `0` indicates a full dump, in which case `parentid` should also be `0`. A value of `1` or greater indicates an incremental dump made at the specified level in the dump hierarchy.
  - The number of volumes in the dump.
  - The creation date of the dump.
- The second line, labeled `Tape:`, displays the tape name containing the dump followed by the dump ID of the initial dump of the dump set (in parentheses).
- The third line displays the number of volumes in the dump and the creation date of the dump.
- The fourth line contains header labels for the remaining lines in the output. The data in these columns include the following information about the volumes in this dump for the tape listed in the second line.
  - `Pos`: The relative position of each volume on the corresponding tape beginning with position *2*. (Position *1* is used by the tape label.)
  - `Clone time`: For a Backup or ReadOnly volume, this is the time at which it was cloned from its ReadWrite source. For a ReadWrite volume, this is the same as the creation date for the corresponding dump.
  - `Nbytes`: The size of each volume in bytes.
  - `Volume`: The name of each volume.

If both **-id** and **-verbose** are specified, the command displays the following information:

- The heading `Dump` followed by a number of lines displaying detailed information about the specified dump.

- The heading `Tape` followed by a number of lines displaying detailed information about the tape containing the specified dump.
- Zero or more occurrences of the heading `Volume`, each of which is followed by detailed information about one of the volumes included in the specified dump.

EXAMPLES

The following displays information about the last five dumps:

% backup dumpi 5

```
   dumpid   parentid lv created        nt nvols dump name
690428867  690169399 1  02/05/92 05:29  1    62 usr.wed (690169399)
690514342  690169399 1  02/06/92 05:21  1    46 usr.thu (690169399)
690600291  690169399 1  02/07/92 05:23  1    58 usr.fri (690169399)
690872726          0 0  02/09/92 04:12  1    22 sys.week
690874954          0 0  02/09/92 05:31  2    73 usr.sun
```

The following displays more detailed information about the dump set with dump ID *741366407*:

% backup dumpi -id 741366407 -verb

```
Dump
----
id = 741366407
Initial id = 0
Appended id = 741366616
parent = 0
level = 0
flags = 0x0
volumeSet = user
dump path = /monday1
name = user.monday1
created = Tue Jun 29 11:06:46 1993
nVolumes = 1
```

/ continued on following page */*

```
id  = 0
tapeServer =
format = user.monday1.%d
maxTapes = 1
Start Tape Seq = 1
name = kdu
instance =
cell =

Tape
----
name = user.monday1.1
flags = 0x20
written = Tue Jun 29 11:09:45 1993
expires = NEVER
kBytes Tape Used = 121
nMBytes Data = 0
nBytes  Data = 19092
nFiles = 0
nVolumes = 1
seq = 1
tapeid = 0
useCount = 1
dump = 741366407

Volume
------
name = testvol.1
flags = 0x18
id = 536871640
server =
partition = 0
nFrags = 1
position = 2
clone = Tue Jun 29 11:06:45 1993
startByte = 0
nBytes = 19092
seq = 0
dump = 741366407
tape = user.monday1.1
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup deletedump

backup help — show syntax of specified **backup** command(s) or list functional descriptions of all of them.

backup help [-topic <*help string*>+] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup h [-t <*help string*>+**] [-h]**

DESCRIPTION

Displays the first line (name and short description) of every **backup** command's help entry if no *help string* is provided. For each operation code specified with **-topic**, it outputs the entire help entry. See the OUTPUT section below.

ARGUMENTS

-topic	specifies the operation code(s) for which syntax is to be provided. If it is omitted, the output instead provides a short description of all **backup** commands.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The online help entry for each **backup** command consists of two or three lines:

- The first line names the command and briefly describes what it does.
- The second line displays any aliases the command has (this line does not appear for every command).
- The final line, which begins with "`Usage:`", lists the command's arguments and flags in the prescribed order. Online help entries use the same symbols (for example, brackets) as the command definitions in this manual. For an explanation of their meaning, see page x of the introductory *About This Manual* chapter.

EXAMPLES

The following displays the online help entry for the **backup dump** command:

% backup help dump

```
backup dump: start dump
Usage: backup dump -volumeset <volume set name> -dump <dump level name>
[-portoffset <TC port offset>] [-at <Date/time to start dump>+]
[-append] [-n] [-file <load file>] [-help]
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup apropos

backup jobs — list operations currently in progress.

(backup) jobs [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

(backup) j [-h]

DESCRIPTION

When working in interactive mode, this command can be used to list the job ID number of each **backup** operation currently executing or pending. The output is similar to that of the **backup status** command. However, the output from **backup status** reports on the activity of a specific tape drive, whereas the output from **(backup) jobs** reports on the status of jobs executed within the current interactive backup session, indicating the job ID number of each operation. An operation's job ID can be used to terminate it with the **(backup) kill** command. See the OUTPUT section for details on the information in the output.

Note: This command can be issued in interactive mode only.

ARGUMENTS

-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The command always displays the expiration date and time of the tokens used by the Backup Server to execute **backup** commands. Without a valid token for the specified cell, no **backup** commands can be executed.

If operations are pending or executing, the following information is displayed for each operation:

- The job ID number.
- The date and time of execution for scheduled dumps.
- A name describing the operation:
  - `Dump (`*dump name*`)` for a dump operation initiated with the **backup dump** command. In the operation name, *dump name* is the name of the dump in the form <*volume set name*>.<*dump level name*>.
  - `Restore` for a restore operation initiated with the **backup volrestore** or **backup diskrestore** command.
  - `Labeltape (`*tape label*`)` for a tape labeling operation started with the **backup labeltape** command. In the operation name, *tape label* is the tape label specified with the **backup labeltape** command's arguments.
  - `Scantape` for a tape scanning operation initiated with the **backup scantape** command.
  - `SaveDb` for a **backup savedb** operation.

  - `RestoreDb` for a **backup restoredb** operation.

- For a dump operation, the number of kilobytes transferred from file system to tape. For a restore operation, the number of kilobytes transferred from tape to file system.
- For a dump operation, the string `volume` followed by the name of the volume currently being dumped. For a restore operation, the string `restore.volume`.
- One of the following messages indicating the status of the operation. No message is displayed if the operation is executing normally.
  - `[abort requested]` means the **(backup) kill** command was issued, but it has yet to be sent to the Tape Coordinator.
  - `[abort sent]` means the operation is canceled. Once the system removes it from the queue or stops its execution, the operation no longer appears in the listing from the **(backup) jobs** command.
  - `[butc contact lost]` means the **backup** program cannot access the Tape Coordinator. It can indicate that the Tape Coordinator executing the operation was killed or failed while the operation was executing, or it can mean that the connection to the Tape Coordinator timed out.
  - `[drive wait]` means the operation is waiting for the specified tape drive to become free.
  - `[operator wait]` means the Tape Coordinator is waiting for the operator monitoring the operation's execution to insert a tape in the drive.

EXAMPLES

The following shows that two restore operations and one dump operation have been queued and that the tokens required by the Backup Server expire on April 22, 1993 at 10:45 am:

```
backup> jobs
Job 1 Restore : 13060566 bytes transferred : restore.volume
Job 2 Dump (user.sunday1) : 34923 bytes transferred : volume
user.pat.backup
Job 3 Restore : 24987527 bytes transferred : restore.volume
       04/22/93 10:45: TOKEN EXPIRATION
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

(backup) kill

(backup) quit

backup kill — terminate a pending or executing operation.

(backup) kill -id <*job ID or dump set name*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

(backup) k -i <*job ID or dump set name*> **[-h]**

DESCRIPTION

When in interactive mode, this command cancels a specified operation. A dump operation can be killed by specifying either the job ID number associated with the operation or the name of the dump being backed up. Other operations can be killed *only* by specifying the job ID number associated with the operation:

- If you abort a pending operation, it is removed from the "queue" with no other effects.
- If you abort a tape labeling or tape scanning operation as it executes, the operation terminates with no further effects.
- If you abort a dump operation, all of the volumes written to the tape before the kill signal is received are complete and usable.
- If you abort a restore operation, completely restored volumes are online and usable. However, it is unlikely that many volumes are completely restored, given that complete restoration usually requires data from multiple tapes. If the kill signal comes before the Backup System has accessed all of the necessary tapes, each volume is only partially written and is never brought online. The restore operation must be restarted to avoid possible inconsistencies.

Note: This command can be issued in interactive mode only.

WARNING

If a restore operation is interrupted while overwriting one or more existing volumes, it is possible to lose the volumes entirely (that is, to lose the contents of each volume as it was before the restore and any data that was restored before the kill signal). However, the data being restored still exists on the tape.

ARGUMENTS

-id identifies the backup operation to kill. Two types of values are legal for this argument:

- The operation's job ID number, as displayed in the output of **(backup) jobs**.
- If it is a dump operation, the name of the dump in the form

  <volume set name>.<dump level name>

  where *volume set name* is the name of the volume set being dumped and *dump level name* is the last element in the dump level at which the volume set is being dumped. The dump name also appears in the output of **(backup) jobs** along with the job ID number.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following kills the operation with job ID 5:

```
backup> kill 5
```

The following kills the dump operation of a dump named *user.sunday1*:

```
backup> kill user.sunday1
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

(backup) jobs

backup labeltape — create label on tape.

backup labeltape [-name <*tape name, defaults to NULL*>**]**
[-size <*tape size in Kbytes, defaults to size in tapeconfig*>**]**
[-portoffset <*TC port offset*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup la [-n <*tape name, defaults to NULL*>**]**
[-s <*tape size in Kbytes, defaults to size in tapeconfig*>**] [-p** <*TC port offset*>**] [-h]**

DESCRIPTION

Creates a label, readable by the Backup System, at the beginning of a tape. The issuer may either assign a name or label the tape as "empty." See the ARGUMENTS section below to learn the required format for tape names.

You will need to relabel a tape if

- You want to create a new dump set on a tape and the existing name on the tape label is different from the name of the new initial dump name.
- You want to reuse a tape before all of the expiration dates associated with the dump set have expired.
- The tape capacity of the tape is different from the default capacity for the tape drive that you intend to use. The default tape capacity for a tape drive is recorded in the */usr/afs/backup/tapeconfig* file.

Besides replacing a tape's label, relabeling a tape also removes all information about the tapes corresponding dump set (both its initial and appended dumps) from the Backup Database.

The **-size** argument is useful if a tape's capacity differs from the capacity of tapes usually used in the tape drive (the file */usr/afs/backup/tapeconfig* records the default capacity of the drive). If the issuer omits this argument, the Backup System labels the tape with the default tape size it finds in the *tapeconfig* file; if the tape was already labeled with a size, the size in the label is overwritten with the size from the *tapeconfig* file.

Before performing a **backup dump** command, the Tape Coordinator looks at the size listed in the tape's label. Only that amount of data is dumped to the tape. If the tape's label contains no size, the amount of data specified in the *tapeconfig* file is dumped to the tape. In the latter case, make sure the tapes are at least as large as the tape size listed in the *tapeconfig* file.

By default, when you execute this command, the **butc** process will prompt you to insert the appropriate tape in the correct drive and press **Return**. However, if that **butc** process was executed using the **-noautoquery** flag, you will not be prompted in this manner. Instead, you are expected to insert the tape in the drive prior to executing the command; if you do not insert the appropriate tape, you will be prompted to insert a tape via an error message on the Tape Coordinator window.

ARGUMENTS

-name specifies the name to assign to the tape. If this argument is omitted, the tape is marked as "" (that is, a *NULL* indicator is included on the tape's label). An assigned name must reflect the name of the dump set (based on the initial dump) that will go on the tape; the tape name must have the following format:

<volume set name>.<dump level name>.<tape index>

where *volume set name* is the name of the volume set of the initial dump that will be dumped onto the tape, *dump level name* is the last element in the dump level at which the volume will be dumped, and *tape index* indicates the order of the tape in the dump set (for a new dump set this must be *1*).

-size indicates the tape's capacity. See the DESCRIPTION section for information about how the size in a tape's label is used during a dump operation.

Specify a number and a letter. Add a `k` or `K` to the number to indicate kilobytes, an `m` or `M` to indicate megabytes, or a `g` or `G` to indicate gigabytes. The default unit is kilobytes.

Specify a size only if the size of the tape is different from what is listed for the drive in the */usr/afs/backup/tapeconfig* file. Omit this argument to label the tape with the size specified for the drive in the *tapeconfig* file (even if the tape was previously labeled, in which case it already has a size recorded in its label).

-portoffset specifies the port offset number of the Tape Coordinator/tape drive that you intend to use. See section 1.2 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following puts the label *user.monthly.1* on the tape in the drive with port offset *3*:

% **backup la user.monthly.1 -p 3**

The following three commands are equivalent in effect: they all mark the tape in the drive with port offset *4* with a capacity of 2 gigabytes and the default name *NULL*.

% **backup label -size 2g -port 4**
% **backup label -size 2048M -port 4**
% **backup label -size 2097152 -port 4**

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup readlabel

butc

backup listdumps — display dump hierarchy from Backup Database.

backup listdumps [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup listd [-h]

DESCRIPTION

Displays the dump hierarchy from the Backup Database.

ARGUMENTS

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The output displays the dump hierarchy and the relationship among the full, parent, and incremental dump levels. If an expiration date is associated with a dump level, it is displayed also.

Full dump levels appear at the far left margin. There may be more than one full dump level in the hierarchy; each full dump defines a "subhierarchy" of dump levels that can be used for dumping different volume sets.

Incremental dump levels appear below and indented to the right of their parent dump levels, which may be either full or incremental. An incremental dump level does not have to be directly below its parent; the amount of indentation indicates the parent/child relationship.

EXAMPLES

The following depicts a dump hierarchy with six subhierarchies. The first two subhierarchies might be used to dump data that does not change very often. The last four subhierarchies might be used to dump data that changes often.

% **backup listdumps**

```
/week1  expires in  27d
      /tuesday  expires in  13d
            /thursday  expires in  13d
      /sunday  expires in  13d
            /tuesday expires in  13d
                  /thursday expires in  13d
/week3  expires in  27d
      /tuesday  expires in  13d
            /thursday  expires in  13d
      /sunday  expires in  13d
            /tuesday  expires in  13d
                  /thursday  expires in  13d
sunday1   expires in  27d
      /monday1  expires in  13d
      /tuesday1  expires in  13d
      /wednesday1  expires in  13d
      /thursday1  expires in  13d
      /friday1  expires in  13d
sunday2   expires in  27d
      /monday2  expires in  13d
      /tuesday2  expires in  13d
      /wednesday2  expires in  13d
      /thursday2  expires in  13d
      /friday2  expires in  13d
sunday3   expires in  27d
      /monday1  expires in  13d
      /tuesday1  expires in  13d
      /wednesday1  expires in  13d
      /thursday1  expires in  13d
      /friday1  expires in  13d
sunday4   expires in  27d
      /monday2  expires in  13d
      /tuesday2  expires in  13d
      /wednesday2  expires in  13d
      /thursday2  expires in  13d
      /friday2  expires in  13d
```

The expiration dates for all incremental dump levels is 13 days so that the corresponding tapes can be recycled two weeks after their creation. The expiration dates for all full dump levels is 27 days so that the corresponding tapes can be recycled four weeks after their creation.

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup adddump **backup deldump**

backup listhosts — list machines running tape drives.

backup listhosts [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup listh [-h]

DESCRIPTION

Displays the port offset number and Tape Coordinator machine of each Tape Coordinator currently defined in the Backup Database. Each defined Tape Coordinator is available for use; a Tape Coordinator need not be running to appear in this list.

ARGUMENTS

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

After a `Tape hosts:` header, the output reports two things about each Tape Coordinator currently defined in the Backup Database:

- The name of the machine housing the Tape Coordinator. The format of this name depends on the format specified by the issuer of the **backup addhost** command.
- The port offset number of the Tape Coordinator.

The Tape Coordinators are displayed in the order in which they were added to the Backup Database.

EXAMPLES

The following shows the result of the command:

% backup listhosts

```
Tape hosts:
    Host backup1.transarc.com, port offset 0
    Host backup1.transarc.com, port offset 1
    Host backup3.transarc.com, port offset 4
    Host backup2.transarc.com, port offset 3
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup addhost

backup delhost

backup listvolsets — list volume set(s) defined in Backup Database.

backup listvolsets [-name <*volume set name*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup listv [-n <*volume set name*>**] [-h]**

DESCRIPTION

Lists the volume sets defined in the Backup Database. It also lists the volume entries defined in each volume set.

ARGUMENTS

-name	names the volume set to be displayed. If it is omitted, all volume sets in the Backup Database are displayed.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The output includes a separate entry for each volume set. The entry lists the volume entries that make up the volume set and displays the index number assigned to each volume entry. The issuer of **backup delvolentry** can use the index number to identify volume entries to be deleted.

EXAMPLES

The following shows the volume entries in the three volume sets currently defined in the Backup Database:

% backup listv

```
Volume set user:
    Entry   1: server .*, partition .*, volumes: user.*\.backup

Volume set sun
    Entry   1: server .*, partition .*, volumes: vax_ul4\..*
    Entry   2: server .*, partition .*, volumes: vax_ul43\..*

Volume set rt
    Entry   1: server .*, partition .*, volumes: sun4c_52\..*
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup addvolentry

backup addvolset

backup delvolentry

backup delvolset

backup quit — leave interactive mode.

(backup) quit [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

(backup) q [-h]

DESCRIPTION

Exits backup interactive mode. The issuer's regular shell prompt replaces the "backup>" interactive prompt. The command has no effect when issued outside interactive mode. (Control-d can also be used to exit interactive mode.)

WARNING

To exit interactive mode, all jobs must be completed.

ARGUMENTS

-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following exits interactive mode:

```
backup> q
%
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

(backup) jobs

(backup) kill

backup readlabel — read and display a tape's label.

backup readlabel [-portoffset <*TC port offset*>] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup rea [-p <*TC port offset*>**] [-h]**

DESCRIPTION

Displays information from the magnetic tape label of a tape. Among the information in the label is the tape's name and size, which are placed on the tape with either **backup dump** or **backup labeltape**.

By default, when you execute this command, the **butc** process will prompt you to insert the appropriate tape in the correct drive and press **Return**. However, if that **butc** process was executed using the **-noautoquery** flag, you will not be prompted in this manner. Instead, you are expected to insert the tape in the drive prior to executing this **backup** command; if you do not insert the appropriate tape, you will be prompted to insert a tape via an error message on the Tape Coordinator window.

ARGUMENTS

-portoffset	specifies the port offset number of the Tape Coordinator/tape drive that you intend to use. See section 1.2 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The following output is displayed from the label of a tape. Note that some of the information may not be included for tapes labeled as empty (NULL).

- `name` is the tape name, in the format <*volume set name*>.<*dump level name*>.<*tape index*> where *volume set name* is the name of the volume set associated with the initial dump in the dump set of which this tape is a part; *dump level name* is the name of the dump level at which the initial dump was backed up; and *tape index* is the numerical position of the tape in the dump set.
- `creationTime` indicates the date and time at which the Backup System started executing the dump operation that created the dump set.
- `cell` indicates the cell in which the dump set was created. This is the cell whose Backup Database contains a record of the dump set.
- `size` indicates the tape's capacity in kilobytes (not the amount of data on the tape). The value comes from the tape label and is derived from the **-size** argument specified with **backup labeltape** or from the *tapeconfig* file, rather than from a measurement of the tape.
- `dump path` names the dump level of the initial dump in the dump set.

- `dump id` indicates the dump ID number of the initial dump in the dump set.
- `useCount` indicates the number of times a dump has been written to the tape.

If the tape is unlabeled or if the specified tape drive is empty, the output reads

```
Failed to read tape label.
```

EXAMPLES

The following shows the output for the tape with label *guests.monthly.3* and capacity 2 megabytes, expressed in kilobyte units (2097152 equals 2 times 1024^2).

% backup read 6

```
Tape read was labelled: guests.monthly.3 (697065340)
     size: 2097152 Kbytes
```

The following detailed output is displayed in the Tape Coordinator window associated with port offset *6*:

```
Tape label
----------
name =           guests.monthly.3
creationTime =    Fri Feb  7 05:31:32 1992
cell =           transarc.com
size =           2097152 Kbytes
dump path = /monthly
dump id = 697065340
useCount = 44
-- End of tape label --
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup labeltape

butc

backup restoredb — restore a backup copy of the Backup Database.

backup restoredb [-portoffset <*TC port offset*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup res [-p <*TC port offset*>**] [-h]**

DESCRIPTION

Restores a backup copy of the entire Backup Database from tape. If the Backup Database is damaged, delete the old database and use this command to restore an entirely new version from its backup tape.

When the Backup Database is restored, the following will occur:

- Existing dump records in the old database are not overwritten.
- Volume sets and entries, host entries, and dump level entries are overwritten.

The **backup savedb** command must have been used to dump the Backup Database to tape before this command can be used. The tape that contains the backup version of the database has the label *Ubik_db_dump.1* (*dump ID*). Subsequent tapes are labeled *Ubik_db_dump.2* (*dump ID*) and so on.

Ubik_db_dump.1. The *1* indicates the tape contains the first dump of the database; subsequent tapes required for larger databases are labeled *2*, *3*, and so on.

By default, when you execute this command, the **butc** process will prompt you to insert the appropriate tape in the correct drive and press **Return**. However, if that **butc** process was executed using the **-noautoquery** flag, you will not be prompted in this manner. Instead, you are expected to insert the tape in the drive prior to executing this **backup** command; if you do not insert the appropriate tape, you will be prompted to insert a tape via an error message on the Tape Coordinator window.

WARNING

Do not attempt to recover information from a corrupted database. Instead, stop all *buserver* processes and remove the old Backup Database from each machine on which it is installed.

Once the database is removed, restart all *buserver* processes on the machines on which they were running. Use **backup addhost** to add a tape host for the Tape Coordinator from which you will restore the Backup Database. Then use **backup restoredb** to restore the database from the Tape Coordinator just reinstalled with **backup addhost**.

ARGUMENTS

-portoffset specifies the port offset number of the Tape Coordinator/tape drive that you intend to use. See section 1.2 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following shows an example of the Backup Database being restored from the Tape Coordinator with port offset *0*:

```
% backup res
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup dbverify

backup savedb

butc

backup savedb — create a backup copy of the Backup Database.

backup savedb [-portoffset <*TC port offset*>**] [-archive** <*date time*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup sa [-p <*TC port offset*>**] [-a** <*date time*>**] [-h]**

DESCRIPTION

Creates a backup copy of the entire Backup Database. A tape to which the Backup Database is saved must be labeled *Ubik_db_dump.1* or *NULL* or it must be unlabeled. A record of the dumped Backup Database is created in the Backup Database and a dump ID is associated with that record. If the **-archive** argument is included, all dump records created prior to the specified *date time* will be deleted from the Backup Database (except for Backup Database dump records).

By default, when you execute this command, the **butc** process will prompt you to insert the appropriate tape in the correct drive and press **Return**. However, if that **butc** process was executed using the **-noautoquery** flag, you will not be prompted in this manner. Instead, you are expected to insert the tape in the drive prior to executing this **backup** command; if you do not insert the appropriate tape, you will be prompted to insert a tape via an error message on the Tape Coordinator window.

ARGUMENTS

-portoffset specifies the port offset number of the Tape Coordinator/tape drive that you intend to use. See section 1.2 for more details.

-archive specifies a deletion date and time cut off. After the Backup Database is dumped, all dump records created before the specified date and time are deleted from the Backup Database. If a dump set contains one or more dumps whose creation dates fall after the specified date, none of the dumps in the dump set are deleted.

The instance must have the following format: *mm/dd/yy hh:MM*, where *mm/dd/yy* indicates the month (mm), day (dd), and year (yy) and *hh:MM* indicates the hour (hh) and minutes (MM) in 24-hour format. The date is required. The time is optional; if it is excluded, the time defaults to *12:00:59 am*.

You can also use **NOW** for this instance to specify the current date and time. Using **NOW** will delete all database records in the Backup Database except for Backup Database dump records.

Note: Although the usage message for this command indicates that multiple instances can be included for this argument, it is not possible to specify more than one archive date and time. The plus sign follows the argument because a multi-word argument is required to define an archive date, not because multiple dates can be specified.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following shows an example of the Backup Database being saved to the Tape Coordinator with port offset *1*:

```
% backup sa 1
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup dbverify

backup restoredb

butc

backup scantape — extract dump set information from tape.

backup scantape [-dbadd] [-portoffset <*TC port offset*>] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup sc [-d] [-p <*TC port offset*>**] [-h]**

DESCRIPTION

Reads the tape in the drive controlled by the Tape Coordinator associated with the specified **-portoffset**, extracting information from the dump labels and volume headers on the tape. The Tape Coordinator displays the information about each dump on its window as soon as the information is extracted. If the **-dbadd** flag is provided, the **backup scantape** command will also record the scanned information in the Backup Database (assuming that the information is not already recorded).

If the **backup scantape** command determines that there is another tape, it prompts for it. If the command cannot determine if another tape exists, it will query the user with the following message:

```
Are there more tapes? (y/n)
```

The command displays a warning message and fails if you attempt to scan a tape containing a backup version of the Backup Database (that is, a tape created with the **backup savedb** command or labeled with the name *Ubik_db_dump.1*).

By default, when you execute this command, the **butc** process will prompt you to insert the appropriate tape in the correct drive and press **Return**. However, if that **butc** process was executed using the **-noautoquery** flag, you will not be prompted in this manner. Instead, you are expected to insert the tape in the drive prior to executing this **backup** command; if you do not insert the appropriate tape, you will be prompted to insert a tape via an error message on the Tape Coordinator window. Of course, if the operation requires multiple tapes, operator intervention will be required to make tape changes.

WARNING

If a tape is damaged, complete information about all volumes prior to the damaged spot can be scanned from that tape; however, data after the damaged spot cannot be scanned back into the Backup Database. Any information on subsequent tapes in a dump set, after a bad spot, is also lost for purposes of recovering data for the Backup Database.

If a tape in a dump set is relabeled (with **backup labeltape**), that and any subsequent tapes in the dump set are lost for purposes of recovering data for the Backup Database.

ARGUMENTS

-dbadd adds the information extracted from the tape to the Backup Database (but only if the database does not already contain an entry with the same dump ID number).

-portoffset specifies the port offset number of the Tape Coordinator/tape drive that you intend to use. See section 1.2 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

For every dump on a tape, this command displays the dump label and the volume header of each volume in the corresponding dump. If a dump spans more than one tape, the dump label will not appear at the beginning of subsequent tapes.

A dump label contains the following fields:

- `tape name` is the tape name, in the format <*volume set name*>.<*dump level name*>.<*tape index*> where *volume set name* is the name of the volume set associated with the initial dump in the dump set; *dump level name* is the name of the dump level at which the initial dump was backed up; and *tape index* is the numerical position of the tape in the dump set.
- `creationTime` indicates the date and time at which the Backup System started executing the dump operation that created this dump.
- `cell` indicates the cell in which the dump was created. This is the cell whose Backup Database contains the dump record.
- `size` indicates the tape's capacity in kilobytes (not the amount of data on the tape). The value comes from the tape label and is derived from the **-size** argument specified with **backup labeltape** or from the *tapeconfig* file, rather than from a measurement of the tape.
- `dump path` names the dump level of the dump.
- `dump id` indicates the dump ID number of the dump.
- `useCount` indicates the number of times a dump was written to the tape.

The volume header contains the following fields:

- `volume name` names the volume complete with a `.backup` or `.readonly` extension, if appropriate.
- `volume ID` is the volume's volume ID.
- `dumpSetName` indicates the dump to which the volume belongs. The dump name is of the form<*volume set name*>.<*dump level name*>, which should match the name displayed from the dump label.
- `dumpID` is the dump ID of the dump named by `dumpSetName`.
- `level` indicates the depth in the dump hierarchy of the dump level used in creating the dump. A value of `0` indicates a full dump. A value of `1` or greater indicates an incremental dump made at the indicated depth in the hierarchy. The value reported is for the entire dump, not necessarily for the volume itself; for example, it is possible for an individual volume to be dumped at a higher level if it was omitted from a previous dump.

- `parentID` is the dump ID number of `dumpSetName`'s parent dump. This should be `0` if `level` is `0`.
- `endTime` should always be `0`; it is for internal use only.
- `cloneDate` indicates the date and time at which the volume was created. For a Backup or ReadOnly volume, this represents the time at which it was cloned from its ReadWrite source. For a ReadWrite volume, it indicates the time at which the Backup System accessed the volume for purposes of including it in `dumpSetName`.

The following sequence (the second line in particular) indicates that the tape was made with the AFS 3.0 version of the Backup System. All such tapes lack the special "endpoint" markers the command interpreter expects to find, but in this case the command interpreter interprets the "`read error`" as the end of the tape rather than as damage. After the last tape containing the dump set is read (that is, the operator responds "`n`" to the "`Are there more tapes?`" prompt), the command interpreter prompts the operator to provide a complete pathname to be recorded as the dump set's dump level name.

```
file_tm: I/O error Reading file mark
read error - end of tape inferred
Are there more tapes? (y/n) n
Please supply the dump schedule path for dump name:
```

where *dump name* is the initial dump in the dump set of which the tape is a part. The final element in the pathname provided by the operator must match the *dump level name* currently found in the second field of `name` in the dump label. The entire pathname must be correct if this dump is to be used as the parent reference point for later incremental dumps.

EXAMPLES

The following shows the output for the first two volumes on a tape in the drive with port offset *0*:

% backup scantape

```
Dump label
----------
name =           guests.monthly.1
creationTime =    Fri Feb  7 05:31:32 1992
cell =           transarc.com
size =           2150000 Kbytes
dump path = /monthly
dump id = 697065340
useCount = 44
-- End of dump label --

-- volume --
volume name: user.guest10.backup
volume ID 1937573829
dumpSetName: guests.monthly
dumpID 697065340
level 0
parentID 0
endTime 0
clonedate Fri Feb  7 05:03:23 1992

-- volume --
volume name: user.guest11.backup
volume ID 1938519386
dumpSetName: guests.monthly
dumpID 697065340
level 0
parentID 0
endTime 0
clonedate Fri Feb  7 05:05:17 1992
   .       .   .   .     .     .
   .       .   .   .     .     .
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup dump

backup dumpinfo

butc

backup setexp — set expiration date of one or more existing dump levels.

backup setexp -dump <*dump level name*>+ **[-expires** <*expiration date*>**]** **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup se -d <*dump level name*>+ **[-e** <*expiration date*>**] [-h]**

DESCRIPTION

Sets the expiration date on each dump level specified with **-dump**. The dump level must already exist in the dump hierarchy stored in the Backup Database.

The expiration date is assigned to all dumps made at that dump level. A tape cannot be overwritten until all dumps in the associated dump set have expired. Attempts to overwrite an unexpired tape will fail until the issuer relabels the tape with **backup labeltape**. If no expiration date is defined for a dump level, the dumps backed up at that level are considered expired immediately.

Expiration dates may be either absolute or relative:

- Absolute expiration dates are defined as a specific month/day/year and, optionally, hour and minutes. A dump with an absolute expiration date expires at that date and time. If the expiration date predates the dump creation, the dump is immediately treated as expired.
- Relative dates are defined as a number of years, months, and/or days. When the Backup System creates a dump at the dump level, it calculates the dump's actual expiration date by adding the relative date to the start time of the dump operation.

ARGUMENTS

-dump specifies the full pathname of each dump level to be assigned the specified expiration date.

-expires defines the expiration date to be associated with all dump levels specified with the command. Absolute expiration dates have the format

[at] <*mm*/*dd*/*yy* **[***hh***:***mm***]>**

or

[at] NEVER

where the optional word **at** is followed by a date (month/day/year) and, optionally, a time (hours:minutes) or by the key word NEVER (which specifies that dumps at that dump level never expire). If provided, the time should be in 24-hour format (for example, *20:30* is 8:30 p.m.); if omitted, the time defaults to 12:00 a.m. (00:00). Valid values for *yy* run from *00* to *37*, which are interpreted as the years 2000 through 2037, and from *70* to *99*, which are interpreted as 1970 through 1999. (Values between *38* and *69* are not valid, because the years to which they correspond, 2038 through 2069, exceed the capacity of the standard UNIX representation of dates. Such values are reduced to 2038.)

Relative expiration dates have the format

[in] [<*integer*>**y] [**<*integer*>**m] [**<*integer*>**d]**

where the word **in** is followed by a number of years (maximum 9999), months (maximum 12), and/or days (maximum 31). At least one of the three must be provided, and the units abbreviations (**y**, **m**, or **d**) must always be used. If more than one of the three is provided, they must appear in the order shown. See the EXAMPLES section for illustrations of acceptable values. As with absolute dates, any dates later than the year 2038 are reduced to 2038.

Omit this argument to assign no expiration date to each specified dump level. Dumps created at a dump level that has no expiration date are assigned an expiration date of 00:00 on 1 January 1970. They are considered to be expired as soon as they are dumped. Use of expiration dates is strongly recommended to prevent accidental overwriting of backup data.

Note: It is not possible to specify more than one expiration date. A plus sign follows this argument because a multi-word argument is required to define an expiration date, not because multiple dates can be specified. An expiration date provided with the **-expires** argument applies to each dump level specified with the **-dump** argument.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following associates an absolute expiration date of 10:00 p.m. on 31 December 1994 with the dump level */92/december*:

% backup setexp /92/december -exp at 12/31/94 22:00

The following associates a relative expiration date of 7 days with the two dump levels */monthly/week1* and */monthly/week2*:

% backup set -d /monthly/week1 /monthly/week -e 7d

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup adddump

backup deldump

backup listdumps

backup status — report the operation a Tape Coordinator is executing.

backup status [-portoffset <*TC port offset*>] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup st [-p <*TC port offset*>] [-h]

DESCRIPTION

Displays the operation, if any, the indicated Tape Coordinator is currently executing. See the OUTPUT section for details on the information in the output.

ARGUMENTS

-portoffset	specifies the port offset number of the Tape Coordinator/tape drive whose status you want to examine. See section 1.2 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

If the indicated Tape Coordinator is not currently performing an operation, the output reports the following information:

```
Tape coordinator is idle
```

Otherwise, it reports the following information:

- A task number associated with each active operation for the specified Tape Coordinator.
- An operation name describing the operation. One of the following operation names is displayed:
  - `Dump (`*dump*`)` for a dump operation initiated with the **backup dump** command. In the operation name, *dump* is the name of the dump in the form <*volume set name*>.<*dump level name*>.
  - `Restore` for a restore operation initiated with the **backup volrestore** or **backup diskrestore** command.
  - `Labeltape (`*tape label*`)` for a tape labeling operation started with the **backup labeltape** command. In the operation name, *tape label* is the tape label specified with the **backup labeltape** command's arguments.
  - `Scantape` for a tape scanning operation initiated with the **backup scantape** command.
  - `SaveDb` for a **backup savedb** operation.

  - `RestoreDb` for a **backup restoredb** operation.

- For a dump operation, the number of kilobytes transferred from file system to tape. For a restore operation, the number of kilobytes transferred from tape to file system.
- For a dump operation, the string `volume` followed by the name of the volume currently being dumped. For a restore operation, the string `restore.volume`.
- One of the following messages reporting the status of the operation. No message is displayed if the operation is executing normally.
  - `[abort requested]` means the **(backup) kill** command was issued, but it has not been sent to the Tape Coordinator yet.
  - `[abort sent]` means the operation is canceled, but its execution is not yet stopped.
  - `[operator wait]` means the Tape Coordinator is waiting for the operator monitoring the command's execution to insert a tape in the drive.

EXAMPLES

The following shows that the Tape Coordinator with port offset *4* is currently dumping the volume named *user.pat.backup*:

```
% backup status 4
Task 4001:  Dump: volume user.pat.backup
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

butc

backup volinfo — query Backup Database about a volume.

backup volinfo -volume <*volume name*> [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

backup voli -v <*volume name*> **[-h]**

DESCRIPTION

Displays a dump history of the specified volume, detailing information such as the date on which the volume was dumped and the tape(s) on which it resides. If the dump was made of the Backup version (as recommended), *volume name* must include the `.backup` extension.

ARGUMENTS

-volume	names the volume for which the dump history is to be displayed. Include a `.backup` or `.readonly` extension if the Backup or ReadOnly version of the volume (rather than the ReadWrite version) was dumped.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The output contains a line of information for each dump of the specified *volume name* contained in the Backup Database. The output is displayed in six columns:

- `dumpID`: The dump ID containing *volume name*.
- `lvl`: The depth in the dump hierarchy of the dump level at which *volume name* was dumped. A value of `0` indicates a full dump. A value of `1` or greater indicates an incremental dump made at the specified level in the dump hierarchy.
- `parentid`: The dump ID of the dump's parent dump. A value of `0` indicates a full dump, which has no parent; in this case, `lvl` should also be `0`.
- `creation date`: The date and time at which the Backup System started the dump operation that created the dump.
- `clone date`: For a Backup or ReadOnly volume, this is the time at which it was cloned from its ReadWrite source. For a ReadWrite volume, this is the same as the `creation date`.
- `tape name`: The name of the tape containing the dump in the form <*volume set name*>.<*dump level name*>.<*tape index*> (*dump ID*) where *volume set name* is the name of the volume set associated with the initial dump in the dump set of which this tape is a part; *dump level name* is the name of the dump level at which the initial dump was backed up; *tape index* is the numerical position of the tape in the dump set; and *dump ID* is the ID of the initial dump in the dump set.

EXAMPLES

The following example shows part of the dump history of the Backup volume *user.smith.backup*:

% backup volinfo user.smith.backup

```
DumpID    lvl parentID  creation date  clone date     tape name
654998415 1   654925629 02/04/92 05:26 02/04/92 05:01 user.tue.1 (654912115)
654912115 1   654925629 02/03/92 05:25 02/03/92 05:05 user.mon.1 (654912115)
654825629 0           0 02/02/92 05:38 02/02/92 04:58 user.sun.1 (654825629)
    .     .      .         .       .       .      .         .
    .     .      .         .       .       .      .         .
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup dumpinfo

backup volrestore

backup volrestore — restore volume(s).

backup volrestore **-server** <*destination machine*>
-partition <*destination partition*>
-volume <*volume(s) to restore*>+
[-extension <*new volume name extension*>**]**
[-date <*date from which to restore*>**]**
[-portoffset <*TC port offsets*>+**] [-n] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

backup volr **-s** <*destination machine*> **-pa** <*destination partition*>
-v <*volume(s) to restore*>+ **[-e** <*new volume name extension*>**]**
[-d <*date from which to restore*>**] [-po** <*TC port offsets*>+**] [-n] [-h]**

DESCRIPTION

Restores the contents of the indicated volume(s) to the indicated site (file server machine and partition). Restores are "full," recreating the volume as it existed as of a given date. A full restore includes data from the last full dump and all subsequent incremental dumps (if any).

The issuer can also choose to do a "date-specific" restore by including the **-date** argument. A date-specific restore returns the volume to the state it was in at that date (and optionally time). In other words, rather than including all dumps up to the most recent one, it includes only the last full dump and any incremental dumps done before the indicated date.

If all full and incremental dumps of a volume set were made on the same type of tape drive, that drive can be used to complete the entire restore operation. If they were made on different types of tape drives, the different drives can be specified with the **-portoffset** argument.

In general, the issuer should provide the name of the volume's ReadWrite version, even if the definition in the volume set caused its Backup version to be dumped.

The precise effect of a restore depends on whether the volume currently exists in the file system and whether the issuer wants to preserve its current state:

- To overwrite a volume's current contents with data restored from tape, omit the **-extension** argument.
  - If **-server** and **-partition** specify the volume's current site, the restored data simply overwrites the current contents; there is no change in the VLDB entry for the volume.
  - If **-server** and/or **-partition** indicate a new site, the restored data goes into a new volume at that site, and the current volume is deleted from the system; the VLDB entry for the volume records the change in site.
- To preserve a volume's current contents but also introduce a restored version into the file system, use the **-extension** argument. Use the **-server** and **-partition** arguments to designate

the site of the new volume. The new volume has the same name as the current volume, with the addition of the distinguishing extension.

- It is also possible to restore a volume that no longer has an existing counterpart in the file system. Use the **-server** and **-partition** arguments to designate the site of the new volume.

The creation date of each restored volume is set to the date and time at which it is restored with this command.

It is possible to have the command interpreter print out a list of the tapes the Backup System would need to perform the indicated restore, without actually performing the restore. To do so, include the **-n** flag.

By default, when you execute this command, the **butc** process will prompt you to insert the appropriate tape in the correct drive and press **Return**. However, if that **butc** process was executed using the **-noautoquery** flag, you will not be prompted in this manner. Instead, you are expected to insert the tape in the drive prior to executing this **backup** command; if you do not insert the appropriate tape, you will be prompted to insert a tape via an error message on the Tape Coordinator window. Of course, if the restore requires multiple tapes, operator intervention will be required to make tape changes.

WARNING

By default, the **backup volrestore** command overwrites the existing volume. Use the **-extension** flag to prevent overwriting by restoring data into a new volume.

ARGUMENTS

-server	names the file server machine on which to restore the volume. If the volume currently exists at a site other than the one specified with this argument and **-partition**, the effect is to change the volume's location at the same time that it is restored (the restored volume goes to the specified machine, and the current volume is removed).
-partition	names the partition on which to restore the volume. If the volume currently exists at a site other than the one specified with this argument and **-server**, the effect is to change the volume's location at the same time that it is restored (the restored volume goes to the specified partition, and the current volume is removed).
-volume	names the volume to restore (the name recorded in the Backup Database). If the name of the ReadWrite version of a volume is provided, the Backup System automatically appends a *.backup* extension to the name. It then determines which version of the volume was dumped most recently, the Backup or the ReadWrite, and restores that version. If a *.backup* or *.readonly* extension is provided with the name of the volume, the Backup System restores only that version of the volume.
-extension	specifies an extension to add to the restored volume's name to distinguish it from a volume of the same name currently existing in the file system. This causes the Backup System to restore the data from tape into a new volume, distinct from the existing one. Any string is acceptable except *.readonly* or *.backup*. If a period is to precede the extension, include it in the string provided.

-date specifies the date before which a dump must have been done to be included in the restore (a date-specific restore). Only dumps dated before **-date** are restored. See the DESCRIPTION section for a more complete explanation of date-specific restores. The instance can have one of two legal values.

- *mm/dd/yy* specifies 12:00 a.m. (00:00) on the indicated date (month/day/year). A value of this type causes a date-specific restore containing data only from dumps done before 00:00 on the indicated date; for example, *1/23/94* or *10/7/94*.
- "*mm/dd/yy hh:mm*" specifies a time (hours:minutes) on the indicated date (month/day/year). A value of this type causes a date-specific restore containing data only from dumps done before the indicated date and time. The time must be in 24-hour format; for example, *20:30* for 8:30 p.m. Date format is the same as for a date alone. Surround the entire instance with double quotes because it contains a space; for example, *"1/23/94 22:30"* or *"10/7/94 3:45"*.

If this argument is omitted, the most recent dump is restored.

-portoffset specifies the port offset number(s) of the tape drive(s) to be used in the operation. If different tape drives were used to create the tapes that contain the dump from which data is to be restored, specify the port offset numbers of all of the drives required for the operation in the order in which they were used to create the dumps (that is, full dump first, next lower incremental, etc,).

Port offsets must be paired with dump levels. Note that a dump needs only to be restored from the same type of drive, not the exact same physical drive.

You must provide this argument unless the default value of *0* is appropriate for all required dump. However, the port offset of *0* must be provided if it is preceded or followed by a different port offset required for another dump.

Note: If you specify more than one volume to restore, the portoffset order must be the same for both volumes. That is, all full dumps must be done on the same portoffset, all level 1 incrementals on the same portoffset, etc.

-n indicates that the command interpreter is to display the list of tapes necessary to perform the indicated restore operation, without actually performing the operation.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

If the issuer includes the **-n** flag with the command, a "`Tapes needed:`" header appears, followed by a list of the tapes necessary to complete the restore operation.

EXAMPLES

The following restores the volume *user.pat* to */vicepa* on *fs5.transarc.com*:

% backup volrestore fs5.transarc.com a user.pat

The following restores the volumes *user.smith* and *user.terry* to */vicepb* on *fs4.transarc.com*, adding a *.restored* extension to each volume to preserve the volumes currently in the file system. The restore is date-specific, with 5:00 p.m. on 31 February 1994 specified as the date and time.

% backup volr fs4.transarc.com b -v user.smith user.terry -e .restored
-d "1/31/94 17:00"]

The following restores the volume *user.pat* to */vicepb* on *fs4.transarc.com*. The Tape Coordinator with port offset *1* is used to restore from the tape containing the full dump; the Tape Coordinator with port offset *0* is used to restore from all tapes containing incremental dumps.

% backup volrestore fs5.transarc.com a user.pat -po 1 0

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*.

MORE INFORMATION

backup dump

backup diskrestore

backup scantape

butc

2. The bos Commands

This chapter describes the **bos** commands that system administrators use to contact the Basic OverSeer Server (BOS Server). It assumes the reader is familiar with the concepts described in the *AFS System Administrator's Guide*.

The BOS Server, an instance of which runs on every file server machine, aids system administrators by monitoring the other server processes on each file server machine. The BOS Server knows how to restart processes automatically should they fail, taking interdependencies among processes into account. Thus, system administrators do not need to continually monitor the status of file server machines.

The BOS Server also provides an interface through which system administrators can

- Install new binaries for server processes.
- Maintain system configuration files.
- Start and stop processes at will.
- Check on server status.

2.1. Configuration Files Used by the bos Commands

The BOS Server and the **bos** commands use four main configuration files:

- */usr/afs/local/BosConfig*
- */usr/afs/etc/CellServDB*
- */usr/afs/etc/KeyFile*
- */usr/afs/etc/UserList*

The *BosConfig* file must be defined on each file server machine. The other three files can either be defined on

- The system control machine, if you are using *upserver* and *upclient* to distribute the contents of */usr/afs/etc*.
- All file server machines, if you are not using *upserver* and *upclient* to distribute the contents of */usr/afs/etc*.

2.1.1. The *BosConfig* File

The */usr/afs/local/BosConfig* file on each file server machine defines the processes relevant to that server and the status of each process. The system administrator uses **bos** commands to define the contents of the *BosConfig* file on each server and thus define the type and function of that server. Upon startup, the BOS Server accesses the local *BosConfig* file to determine which processes to start and monitor.

The system administrator creates entries for server processes in *BosConfig* with the **bos create** command. Many other **bos** commands allow the administrator to control the BOS Server's behavior with respect to a server process, mostly by altering the process' status flag. You should always use **bos** commands to define the contents of *BosConfig*. Never edit the file directly because its format is subject to change and an edited *BosConfig* file can cause unusual problems that are difficult to diagnose.

The *BosConfig* file defines two "restart times:"

- The time each week that the BOS Server should restart itself and then start a new instance of all the server processes that appear in *BosConfig*.
- The time each day that the BOS Server should restart any server process for which it finds new binaries in */usr/afs/bin*.

The **bos getrestart** and **bos setrestart** commands allow the administrator to inspect and set the restart times, respectively.

The file also defines each server process that the BOS Server should monitor. The information about each server process includes the following:

- Its ***type***. There are three types, ***simple***, ***fs***, and ***cron***. See section 2.1.1.1 for an explanation.
- Its ***name***. For simple processes, the name usually matches the filename of the binary executed to run the process (the last element in the command parameter pathname discussed just below), but this is not a requirement.

- Its ***status flag***, which tells the BOS Server two things:
  - Whether it should start up the process as it (the BOS Server) initializes.
  - Whether it should restart the process if it (the process) fails.

  If the flag is `Run`, it means the process should be running whenever possible—the BOS Server should start it initially and restart it automatically if it fails. If the flag is `NotRun`, it means that the BOS Server should not start it initially and not attempt to restart it. Only a system administrator, not the BOS Server, can change this flag.

  It is also possible for the system administrator to control a process's run status independently of this flag. For an explanation, see section 2.1.1.2.
- Its ***command parameters***, which are the commands the BOS Server should issue in order to run the process. The number of commands that appear here depends on the type of the process: "simple" processes have one, the "fs" process three and "cron" processes two. See section 2.1.1.1 just below.

2.1.1.1. The Types of Processes

Simple Process

A ***simple*** process is one that runs independently of any other on the file server machine. Should a simple process fail, the BOS Server can restart it without having to coordinate the restart with that of any other process. All standard AFS processes except the single one of type *fs* are simple (there are no standard cron processes, but the administrator may define them at will).

A simple process has a single command parameter, namely the pathname to its binary file. The *BosConfig* entry for a simple process would include information like the following, which is for the Protection Server:

- `ptserver` as the name.
- `simple` as the type.
- `Run` as the status flag.
- `/usr/afs/bin/ptserver` as the command parameter.

The fs Process

There is only one AFS server process of type ***fs***, the one that is standardly referred to as "*fs*" also. This process combines three components: the File Server, the Volume Server and the Salvager. There are two reasons for combining them into a single process:

- They all operate on the same data, namely the files and directories stored in the AFS file tree. Because they are combined they can coordinate their operations on the data, never attempting simultaneous operations on the same data, which would cause corruption.
- The BOS Server needs to stop and restart the processes in a certain order. Whenever the File Server fails, the BOS Server has to stop the Volume Server and run the Salvager to correct any corruption that may have resulted from the failure. (The Salvager runs only during this special circumstance or when the administrator invokes it using **bos salvage**.) If only the

Volume Server fails, it can restart as though it were simple. Ordering is not important when restarting the Volume Server and File Server at the same time.

The fs process has three command parameters, one for each component. Its *BosConfig* entry contains information like the following:

- `fs` as the name.
- `fs` as the type.
- `Run` as the status flag.
- Three command parameters
  - `/usr/afs/bin/fileserver`
  - `/usr/afs/bin/volserver`
  - `/usr/afs/bin/salvager`

Cron Process

A ***cron*** process is like a simple process in that it involves only one process that runs independently of any other. The difference is that the BOS Server executes *cron* processes only periodically, not continuously.

A cron process has two command parameters. The first is the command to be executed, the second the time at which the BOS Server should execute it.

There are no standard AFS cron processes, but an example of the information in *BosConfig* for one would be:

- `cron-proc` as the name.
- `cron` as the type.
- `Run` as the status flag.
- Two command parameters
  - /usr/afs/bin/cron-proc
  - wed 16:00

2.1.1.2. How the BOS Server Uses the Information in *BosConfig*

When the BOS Server starts or restarts, it reads *BosConfig* to learn about the processes it should monitor. The BOS Server transfers the information into its memory and does not read *BosConfig* again until its next restart.

As a result, there are two potential sources of information for BOS Server: the *BosConfig* file and the BOS Server's memory state. The two are identical when the BOS Server restarts but can diverge after that. For instance, it is possible for a process to stop running even though its status flag in *BosConfig* is `Run` – this happens when the administrator (or the BOS Server) changes the process' status in the BOS Server's memory state. Similarly, the administrator can start running a process without setting its *BosConfig* status flag to `Run`, by setting its memory state flag to `Run`.

2.1.1.3. Changing *BosConfig* and BOS Server Memory State

A number of **bos** commands allow the privileged administrator to alter the information about a process in either *BosConfig* or in the BOS Server's memory state. The **bos status** command allows the issuer to examine a process' status and entry in *BosConfig*.

The following commands alter information in *BosConfig*:

- **bos create**
- **bos start**
- **bos stop**

The commands listed below alter information in the BOS Server's memory state:

- **bos restart**
- **bos shutdown**
- **bos startup**

2.1.2. The *CellServDB* File

The */usr/afs/etc/CellServDB* file on the local disk of each file server machine contains the list of the local cell's database server machines – those machines that run the Authentication, Protection, and VL (Volume Location) Server processes. (Each of these processes maintains a database of administrative information, the Authentication, Protection, and Volume Location Databases, respectively).

The database server processes use the *CellServDB* file to coordinate replication of changes across the multiple copies of each database and to maintain a database server quorum. The non-database server processes use the file to get information from the databases when they need it. For more information, see the section in the *AFS System Administrator's Guide* on the server copy of *CellServDB*.

The initial line in *CellServDB* names the cell to which the server machine (and the listed database server machines) belong. Each additional line specifies the Internet address and name of one database server machine.

The initial version of *CellServDB* is created with **bos setcellname** when the cell's first file server machine is installed; see the *AFS Installation Guide*. The privileged administrator may add and remove database server machines in *CellServDB* with the **bos addhost** and **bos removehost** commands, respectively. No privilege is required to list the contents of *CellServDB* with **bos listhosts**.

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, you should make changes to *CellServDB* only on the cell's system control machine. If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc* and must make changes to each file server and database server machine's copy of *CellServDB* individually. In either case, always use the appropriate **bos** commands to change this file; a manually edited *CellServDB* file can cause unusual problems that are difficult to diagnose. (See chapter 9 for more information on the Update Server's two components, *upserver* and *upclient*.)

2.1.3. The *KeyFile* File

The */usr/afs/etc/KeyFile* file on the local disk of each file server machine contains the list of AFS ***server encryption keys*** that the server processes on that machine can use to decrypt the tickets presented by clients. A complete discussion of server encryption keys is beyond the scope of this chapter, but it is worth noting that it plays a central role in AFS's "mutual authentication" security procedures. AFS server processes will deal only with clients possessing tickets encrypted with server encryption keys found in *KeyFile*. For more detailed information, see the sections on mutual authentication and keys in the *AFS System Administrator's Guide*.

Each key in *KeyFile* is marked with a ***key version number*** that distinguishes it from the other keys. All tickets presented by clients are also marked with a key version number, so that the server process knows which key to use to decrypt the ticket. If a client's ticket is marked with a key version number that does not appear in *KeyFile*, the server process cannot decrypt it.

The privileged administrator may examine, add, and remove keys in *KeyFile* with the **bos listkeys**, **bos addkey**, and **bos removekey** commands, respectively. Note that **bos listkeys** requires privilege even though it only lists information; this is necessary because server encryption keys are so vital to AFS security.

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, you should make changes to *KeyFile* only on the cell's system control machine. If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc* and must change each file server and database server machine's *KeyFile* individually. In either case, always use the appropriate **bos** commands to change this file; a manually edited *KeyFile* file can cause unusual problems that are difficult to diagnose. (See chapter 9 for more information on the Update Server's two components, *upserver* and *upclient*.)

2.1.4. The *UserList* File

The */usr/afs/etc/UserList* file on the local disk of each file server machine contains the user name of each administrator privileged to issue **bos** (and **vos**) commands. The privileged administrator may add and remove other privileged users in *UserList* with the **bos adduser** and **bos removeuser** commands, respectively. No privilege is required to list the contents of *UserList* with **bos listusers**.

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, you should make changes to *UserList* only on the cell's system control machine. If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc* and must change each file server and database server machine's *UserList* individually. In either case, always use the appropriate **bos** commands to change this file; a manually edited *UserList* file can cause unusual problems that are difficult to diagnose. (See chapter 9 for more information on the Update Server's two components, *upserver* and *upclient*.)

2.2. Common Arguments and Flags

Most **bos** commands accept the following optional arguments and flags. They are listed in the command descriptions where they apply, and are described in detail below:

[-cell <*cell name*>]

This argument specifies that the command should be run in the indicated *cell name*. The issuer may abbreviate *cell name* to the shortest for that distinguishes it from the other cells listed in */usr/vice/etc/CellServDB* his or her client machine. By default, commands are executed in the cell given by the environment variable AFSCELL, if it is defined. Otherwise, commands are executed in the cell specified by */usr/vice/etc/ThisCell* on the local client machine.

With the **-cell** argument, authentication to the bosserver is performed using the issuer's token for *cell name*. If the server machine is not is the same cell as the issuer's client, this argument must be used or an incorrect token will be used, and authentication will fail.

Note: Do not use the **-cell** argument and the **-localauth** flag on the same **bos** command because they are functionally incompatible. The **-cell** argument instructs a **bos** command to execute with respect to a specific cell; the **-localauth** flag instructs a **bos** command to use always use the cell name given in */usr/afs/etc/ThisCell*.

[-noauth]

This flag instructs the BOS Server not to authenticate the issuer of the command, and thus establishes an unauthenticated connection between the user and BOS Server (the user is recognized as the unprivileged user *anonymous*). It is useful only when authorization checking is disabled on the file server machine (during the installation of a file server machine or when **bos setauth** has been used during other unusual circumstances). In normal circumstances, the BOS Server allows only authorized (privileged) users to issue commands that change the status of a server or configuration file, and will refuse to perform such an action even if the **-noauth** flag is used.

[-localauth]

This flag instructs the **bos** command interpreter running on the local machine to construct a server ticket using the server encryption key with the highest key version number in */usr/afs/etc/KeyFile* on the local machine. The command interpreter presents the ticket to the BOS Server for use in mutual authentication.

This argument is only useful for commands issued on file server machines, since client workstations do not have a *KeyFile*. It is intended for ***cron***-type processes or jobs included in the machine's */usr/afs/local/BosConfig* file. It can also be used if the user is unable to authenticate but is logged into the local UNIX file system as "root".

Note: Do not use the **-localauth** flag and the **-cell** argument on the same **bos** command because they are functionally incompatible. The **-cell** argument instructs a **bos** command to execute with respect to a specific cell; the **-localauth** flag instructs a **bos** command to use always use the cell name given in */usr/afs/etc/ThisCell*.

[-help]

This flag has the same function as the **bos help** command: it prints a command's online help message on the screen. No other arguments or flags should be provided at the same time. Even if they are, this flag overrides them, and the only effect of issuing the command is that the help message appears.

2.3. The Privilege Required for bos Commands

Users who appear in the file */usr/afs/etc/UserList* on a file server machine's local disk are privileged to issue all **bos** commands to the BOS Server on that machine; see the chapter on privileged users in the *AFS System Administrator's Guide* for more information.

Only privileged users can execute commands that change the status of a server or the information in a configuration file. Any user may execute commands that simply list information (with the exception of **bos listkeys**). Consult the PRIVILEGE REQUIRED section of each command description.

bos addhost — add a database server machine to */usr/afs/etc/CellServDB*.

bos addhost -server <*machine name*> **-host** <*host name*>+
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos addh -s <*machine name*> **-ho** <*host name*>+ **[-c** <*cell name*>**] [-n] [-l] [-he]**

DESCRIPTION

Adds each database server machine specified as *host name* to the file */usr/afs/etc/CellServDB* on *machine name*. See section 2.1.2 for more information on the *CellServDB* file.

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, change *CellServDB* only on the cell's system control machine (specify it as *machine name*). By default, it takes up to five minutes for the changed file to propagate to the other file server and database server machines via *upserver* and *upclientetc*.

If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc*. Instead, repeat this command for all of your cell's file server and database server machines, specifying each one as *machine name* in turn. To avoid possible inconsistency problems, finish issuing the commands within a fairly short time (within about five minutes, the default time period within which the Update Server distributes the changes in cells running the United States edition of AFS).

WARNING

After executing this command, the issuer must restart the database server processes on all database server machines. Otherwise, the lightweight process that each database server process runs to handle database replication (Ubik) will not notice the change. See the section in the *AFS Installation Guide* on adding and removing database server machines.

ARGUMENTS

-server names the file server machine on which to change *CellServDB*. With the United States edition of AFS, it should be the cell's system control machine so that the changes propagate to all file server and database server machines in the cell. If the specified machine is not the system control machine but is running an *upclientetc* process, then the changes will be overwritten when that process next brings over *CellServDB* from the system control machine (by default, within five minutes).

With the international edition of AFS, specify the name of each file server and database server machine in the cell in turn.

-host names the new database server machines to be registered in *CellServDB*. The file must record each machine's complete Internet-style host name (such as

fs1.transarc.com), so it is best to provide the complete name explicitly. Otherwise, the name that ends up in the file depends on the state of the cell's naming service at the time the command is issued.

-cell specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.

-noauth tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following adds the database server machine *db3.transarc.com* to the *CellServDB* kept on *fs1.transarc.com* (the system control machine).

```
% bos addhost fs1.transarc.com db3.transarc.com
```

The following adds the database server machines *db2.transarc.com* and *db3.transarc.com* to the *CellServDB* kept on *fs1.transarc.com* (the system control machine). Note that the arguments are used for clarity, not because they are actually necessary.

```
% bos addh -server fs1.transarc.com -host db2.transarc.com db3.transarc.com
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos listhosts

bos removehost

AFS Installation Guide

bos addkey — add a new server encryption key to */usr/afs/etc/KeyFile*.

bos addkey -server <*machine name*> **-key** <*key*>
-kvno <*key version number*> **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

bos addk -s <*machine name*> **-ke** <*key*> **-kv** <*key version number*>
[-ce <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Adds a new server encryption key, derived from *key*, to the file */usr/afs/etc/KeyFile* on *machine name*. It assigns *key version number* as the new key's key version number. Section 2.1.3 discusses *KeyFile* and keys further. Be sure to add this same key (with the appropriate key version number) to the "afs" entry in the Authentication Database, using the **kas setpassword** or **kas setkey** command. (See chapter 4 for information on the **kas** commands.)

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, change *Keyfile* only on the cell's system control machine (specify it as *machine name*). By default, it takes up to five minutes for the changed file to propagate to the other file server and database server machines via *upserver* and *upclientetc*.

If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc*. Instead, repeat this command for all of your cell's file server and database server machines, specifying each one as *machine name* in turn. To avoid possible inconsistency problems, finish issuing the commands within a fairly short time (within about five minutes, the default time period within which the Update Server distributes the changes in cells running the United States edition of AFS).

ARGUMENTS

-server names the file server machine on which to change *KeyFile*. With the United States edition of AFS, it should be the cell's system control machine so that the changes propagate to all file server and database server machines in the cell. If the specified machine is not the system control machine but is running an *upclientetc* process, then the changes will be overwritten when that process next brings over *KeyFile* from the system control machine (by default, within five minutes).

With the international edition of AFS, specify the name of each file server and database server machine in the cell in turn.

-key should define a character string just like a password; the BOS Server calls a DES conversion function to encode it into a form appropriate for use as an encryption key.

-kvno defines the new key's key version number. It must be an integer in the range *0* through *255*. In addition, *999* is reserved for the "bcrypt" key used in AFS 2.0 authentication.

-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following adds a new server encryption key with key version number 14 to the *KeyFile* kept on *fs1.transarc.com* (the system control machine). The string "*fourteenth_new_key*" will be converted into a DES-encrypted key before being placed in the *KeyFile*.

% bos addk fs1.transarc.com fourteenth_new_key 14

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos listkeys

bos removekey

bos adduser — add a privileged user to */usr/afs/etc/UserList*.

bos adduser -server <*machine name*> **-user** <*user names*>+
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos addu -s <*machine name*> **-u** <*user names*>+ **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Adds each specified *user name* to the */usr/afs/etc/UserList* file on *machine name*. The BOS Server does not check that an entry exists in the Protection or Authentication Database for the user being added. It is the issuer's responsibility to verify that the user actually exists. See section 2.1.4 for more information on the *UserList* file.

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, change *UserList* only on the cell's system control machine (specify it as *machine name*). By default, it takes up to five minutes for the changed file to propagate to the other file server and database server machines via *upserver* and *upclientetc*.

If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc*. Instead, repeat this command for all of your cell's file server and database server machines, specifying each one as *machine name* in turn. To avoid possible inconsistency problems, finish issuing the commands within a fairly short time (within about five minutes, the default time period within which the Update Server distributes the changes in cells running the United States edition of AFS).

ARGUMENTS

-server names the file server machine on which to change *UserList*. With the United States edition of AFS, it should be the cell's system control machine so that the changes propagate to all file server and database server machines in the cell. If the specified machine is not the system control machine but is running an *upclientetc* process, then the changes will be overwritten when that process next brings over *KeyFile* from the system control machine (by default, within five minutes).

With the international edition of AFS, specify the name of each file server and database server machine in the cell in turn.

-user specifies the user names of the users who are to become privileged with respect to **bos** and **vos** commands.

-cell specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.

-noauth tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following adds the user names *pat* and *smith* to the *UserList* file kept on *fs1.transarc.com* (the system control machine).

% bos addu fs1.transarc.com pat smith

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos listusers

bos removeuser

bos apropos — show each help entry containing keyword.

bos apropos -topic <*help string*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos ap -t <*help string*> **[-h]**

DESCRIPTION

Displays the first line of the online help entry for any **bos** command that has *help string* in its name or short description.

ARGUMENTS

-topic specifies the keyword string to search for. If it is more than a single word, surround it with double quotes or other delimiters. Type all *help string*s for **bos** commands in all lowercase letters.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

The first line of a command's online help entry names the command and briefly describes what it does. The **bos apropos** command displays that first line for any **bos** command where *help string* is part of the command name or first line.

To see the remaining lines in a help entry, which provide the command's alias (if any) and syntax, use the **bos help** command.

EXAMPLES

The following lists all **bos** commands that have the word "*restart*" in their operation code or short online description.

```
% bos apropos restart
getrestart: get restart times
restart: restart all processes
setrestart: set restart times
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

bos help

bos create — create new process in */usr/afs/local/BosConfig* and start it running.

bos create **-server** <*machine name*> **-instance** <*server process name*> **-type** <*server type*> **-cmd** <*command lines*>+ **[-notifier** <*Notifier program*>**]** **[-cell** <*cell name*>**]** **[-noauth]** **[-localauth]** **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos c **-s** <*machine name*> **-i** <*server process name*> **-t** <*server type*> **-cm** <*command lines*>+ **[-not** <*Notifier program*>**]** **[-ce** <*cell name*>**]** **[-noa]** **[-l]** **[-h]**

DESCRIPTION

Creates a new server process on the file server machine *machine name* by creating an entry in */usr/afs/local/BosConfig* on its local disk. The status field in the new *BosConfig* entry is set to Run, and the process starts running on *machine name*. See section 2.1.1 for more information on the *BosConfig* file.

ARGUMENTS

-server names the file server machine on which to create the new process.

-instance names the process to create. The issuer may choose any name for a process, but for the sake of simplicity, it is best to give the process the same name as its binary file (and to choose the same name on every machine running that process). The recommended names, and those specified in the *AFS Installation Guide*, are

buserver	for the Backup Server process.
fs	for the process combining the File Server, Volume Server, and Salvager.
kaserver	for the Authentication Server process.
ptserver	for the Protection Server process.
runntp	for the controller process for the Network Time Protocol Daemon.
upclientbin	for the client portion of the Update Server process that retrieves binary files from */usr/afs/bin* on the binary distribution machine for this machine's CPU/operating system type.
upclientetc	for the client portion of the Update Server process that retrieves configuration files from */usr/afs/etc* on the system control machine. Do not run this process in cells that use the international edition of AFS.
upserver	for the server portion of the Update Server process.
vlserver	for the Volume Location Server process.

-type specifies the process's type. Legal values are *simple*, *fs*, and *cron*. Specify *simple* for independent processes, *fs* only for the "fs" process (combining the File Server, Volume Server, and Salvager), or *cron* for processes that are to run only at specified times.

-cmd specifies the command(s) the BOS Server is to run to start the process.

For a *simple*-type process, this should be the complete pathname to the binary file for the process (for example, */usr/afs/bin/ptserver* for the Protection Server). The commands for some simple processes (for instance, *runntp*, *upclient*, *upclientbin*, and *upclientetc*) take arguments, in which case the entire instance must be surrounded with double quotes; see the EXAMPLES section. Also consult the entries in chapter 9 for **runntp** and **upclient**, as well as the *AFS Installation Guide* and the chapter on process management in the *AFS System Administrator's Guide*.

For the *fs*-type process, this should be three complete pathnames to the binary files for the File Server, Volume Server, and Salvager, along with any command options.

For a *cron*-type process, provide two parameters:

- The first parameter should be either the pathname to a binary file to be executed, or a command from one of the AFS suites complete with all the necessary arguments. Surround this parameter with quotes or other delimiters if it contains spaces.
- The second parameter specifies the time at which the BOS Server is to execute the command specified by the first parameter. Specify either a day and time or a time alone. Surround this parameter with quotes if it contains spaces.

  If a day is specified, it must appear first, in lowercase letters. Either type the whole name or the first three letters as an abbreviation (*sunday* or *sun*, *thursday* or *thu*, and so on).

  The time of day may be expressed in different ways. In all cases, separate the hours from the minutes with a colon. Twenty-four hour time is understood by default, but the issuer can also use *1:00* through *12:00* if *am* (or *a.m.*) or *pm* (or *p.m.*) is added.

  To execute the command only once, specify *now* instead of a day and/or time. It would, however, be more sensible simply to issue the command directly.

-notifier specifies the pathname of a program to execute when the process terminates. When a process with a corresponding *Notifier program* terminates, BOS Server will execute the program. Transarc does not provide notifier programs; each site must create its own. See NOTES ON NOTIFIER PROGRAMS below for some information on creating a notifier program.

-cell specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.

-noauth tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following creates the simple-type process *kaserver* on *fs3.transarc.com*:

% bos create fs3.transarc.com kaserver simple /usr/afs/bin/kaserver

The following creates the simple-type process *upclientbin* on *fs4.transarc.com*. It specifies *fs1.transarc.com* as the source for updates to */usr/afs/bin*. Consistency with *fs1.transarc.com* is checked every *120* seconds.

% bos create fs4.transarc.com upclientbin simple
"/usr/afs/bin/upclient fs1.transarc.com -clear -t 120 /usr/afs/bin"

The following creates the fs-type process *fs* on *fs4.transarc.com*. The issuer types the command on a single line.

% bos create fs4.transarc.com fs fs /usr/afs/bin/fileserver /usr/afs/bin/volserver
/usr/afs/bin/salvager

The following creates the cron-type process *userbackup* on *fs5.transarc.com*. The effect is that the BOS Server issues the indicated **vos backupsys** command each day at 3:00 a.m. (the command causes Backup versions to be created for all volumes in the file system whose names begin with "user"). Note that it is necessary to provide the complete pathname of the **vos** command and to include the **-localauth** flag (see section 8.3 for an explanation of its function). The issuer types the command on a single line.

% bos cr fs5.transarc.com userbackup cron
-cmd "/usr/afs/bin/vos backupsys -prefix user -localauth" 3:00

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

NOTES ON NOTIFIER PROGRAMS

This section provides information for writing a notifier program. It is meant for use by developers of notifier programs.

The format of the input to the notifier program may vary slightly depending on the type of process to which the notifier program is attached. In general, the format will be one line of ASCII for each field in the bnode and bnode_proc structures, terminated by a new line. Each structure will be preceded and followed by tokens identifying the enclosed data. These tokens have the form `BEGIN struct bnode` and `END struct bnode` and will immediately follow the preceding newline with no intervening spaces or other characters. If the notifier program is not prepared to parse a particular structure, it may simply scan ahead in the input stream for the `END` token.

Below is the C code that will explode the data structures. Some of the fields are not printed out, either because they are not used by **bos** or because they are used only for internal housekeeping.

struct bnode contents

```
struct bnode {
    struct bnode *next;      /* next pointer in top-level's list */
    char *name;              /* instance name */
    long nextTimeout;        /* next time this guy should be awakened */
    long period;             /* period between calls */
    long rsTime;             /* time we started counting restarts */
    long rsCount;            /* count of restarts since rsTime */
    struct bnode_type *type; /* type object */
    struct bnode_ops *ops;   /* functions implementing bnode class */
    long procStartTime;      /* last time a process was started */
    long procStarts;         /* number of process starts */
    long lastAnyExit;        /* last time a process exited for any reason */
    long lastErrorExit;      /* last time a process exited unexpectedly */
    long errorCode;          /* last exit return code */
    long errorSignal;        /* last proc terminating signal */
    char *lastErrorName;     /* name of proc that failed last */
    short refCount;          /* reference count */
    short flags;             /* random flags */
    char goal;               /* 1=running or 0=not running */
    char fileGoal;           /* same, but to be stored in file */
};
```

format of struct bnode explosion

```
printf("name: %s\n",tp->name);
printf("rsTime: %ld\n", tp->rsTime);
printf("rsCount: %ld\n", tp->rsCount);
printf("procStartTime: %ld\n", tp->procStartTime);
printf("procStarts: %ld\n", tp->procStarts);
printf("lastAnyExit: %ld\n", tp->lastAnyExit);
printf("lastErrorExit: %ld\n", tp->lastErrorExit);
printf("errorCode: %ld\n", tp->errorCode);
printf("errorSignal: %ld\n", tp->errorSignal);
printf("lastErrorName: %s\n", tp->lastErrorName);
printf("goal: %d\n", tp->goal);
```

struct bnode_proc contents

```
struct bnode_proc {
    struct bnode_proc *next; /* next guy in top-level's list */
    struct bnode *bnode;     /* bnode creating this process */
    char *comLine;           /* command line used to start this process */
    char *coreName;          /* optional core file component name */
    long pid;                /* pid if created */
    long lastExit;           /* last termination code */
    long lastSignal;         /* last signal that killed this guy */
    long flags;              /* flags giving process state */
};
```

format of struct bnode_proc explosion

```
printf("comLine: %s\n", tp->comLine);
printf("coreName: %s\n", tp->coreName);
printf("pid: %ld\n", tp->pid);
printf("lastExit: %ld\n", tp->lastExit);
printf("lastSignal: %ld\n", tp->lastSignal);
```

The BOS Server will close the standard input file descriptor to the notifier process when it has completed delivery of the data. It is the responsibility of the notifier process to terminate. The notifier process should robustly handle the presence of unexpected data structure contents on standard input. The notifier process should read input data until EOF is detected.

bos delete — delete server process from */usr/afs/local/BosConfig*.

bos delete -server <*machine name*> **-instance** <*server process name*>+
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos d -s <*machine name*> **-i** <*server process name*>+ **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Removes each *server process name* entry from */usr/afs/local/BosConfig* on the file server machine *machine name*.

Before issuing this command, the issuer must execute **bos stop** to stop the actual process running on the file server machine. An error message results if the process' status flag is Run when this command is issued.

ARGUMENTS

-server	names the file server machine from which to delete the process(es).
-instance	names the process(es) to delete. Use the name assigned with the **-instance** argument in the **bos create** command; if the issuer has forgotten the appropriate name, it appears in the output from the **bos status** command.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following removes the *kaserver*, *ptserver* and *vlserver* entries from the *BosConfig* file on *db3.transarc.com*, a database server machine being decommissioned.

% bos d db3.transarc.com kaserver ptserver vlserver

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos create

bos status

bos exec — execute command on file server machine from remote machine.

bos exec -server <*machine name*> **-cmd** <*command to execute*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos e -s <*machine name*> **-cm** <*command to execute*> **[-ce** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Executes the command *command to execute* on the file server machine *machine name*. Its intended use is to reboot the machine using */etc/reboot*.

ARGUMENTS

-server	names the file server machine on which to execute the command.
-cmd	specifies the complete pathname of the command to execute. (for example, */etc/reboot*). Surround this argument with quotes if the command contains spaces.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following causes *fs2.transarc.com* to reboot. Note that before rebooting the issuer has used **bos shutdown** to shutdown all processes safely.

% bos exec fs2.transarc.com /etc/reboot

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

bos getdate — list time stamps of version(s) of a binary file in */usr/afs/bin*.

bos getdate -server <*machine name*> **-file** <*files to check*>$^{+}$
[-dir <*destination dir*>**] [-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

bos getd -s <*machine name*> **-f** <*files to check*>$^{+}$ **[-d** <*destination dir*>**]**
[-c <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Displays the time stamps for the current version, *.BAK* version (if any) and *.OLD* version (if any) of the binary *files to check* found in the */usr/afs/bin* directory (or an alternate directory provided as *destination dir*) on the file server machine *machine name*. (The BOS Server automatically creates *.BAK* and *.OLD* versions when new binaries are installed with **bos install**.)

ARGUMENTS

-server	names the file server machine on which to inspect the specified binary files. All machines should show the same information if the binaries were installed properly on the cell's binary distribution machine(s), and if all non-distribution machines are running the client portion of the Update Server.
-file	names the binary files to check. If more than one is named, they must all reside in the same directory.
-dir	specifies a complete directory pathname, and is necessary only if the indicated file(s) are not in */usr/afs/bin*.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

For each instance of *files to check*, the output reports the time stamp on the current (unmarked), `.BAK`, and `.OLD` versions. If a version does not exist, the output reports so, rather than simply not mentioning the version.

EXAMPLES

The following examines the time stamps on the files with basename *kaserver* on *fs2.transarc.com*:

```
% bos getdate fs2.transarc.com kaserver
File /usr/afs/bin/kaserver dated Mon Jan 1 10:00:36 1990,
.BAK file dated Wed Dec 20 18:55:04 1989, no .OLD file.
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

bos create

bos install

bos prune

bos status

bos uninstall

bos getlog — examine log file for server process.

bos getlog -server <*machine name*> **-file** <*log file to examine*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos getl -s <*machine name*> **-f** <*log file to examine*> **[-c** <*cell name*>**] [-n] [-l] [-h]**

WARNINGS

If you do not periodically shutdown and restart the database server processes, the corresponding log files may grow unmanageably large. To prevent this, you may periodically delete the contents of the log files using the UNIX **rm** command.

Note: The space occupied by a log file may not actually be freed up until five minutes after deletion of that file.

DESCRIPTION

Displays, on standard output (stdout), the requested log file from the specified machine. Unless a pathname is specified, it is assumed that the log file resides in */usr/afs/logs*.

ARGUMENTS

-server	names the file server machine from which to retrieve the log file.
-file	names the log file to display. If a plain file name (one with no slashes) is provided, it is assumed to reside in */usr/afs/logs*; the standard choices from this directory are *AuthLog*, *BackupLog*, *BosLog*, *FileLog*, *SalvageLog*, and *VolserLog*. Relative pathnames are interpreted relative to */usr/afs/logs*; absolute pathnames are also allowed.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

The output is preceded by the line

```
Fetching log file 'filename'...
```

The remainder of the output depends on the particular log file.

EXAMPLES

The following displays the *FileLog* file that resides on *fs3.transarc.com*:

% bos getlog fs3.transarc.com FileLog

```
Fetching log file 'FileLog'...
Sun Nov 4 04:00:34 1990 File server starting
Sun Nov 4 04:00:39 1990 Partition /vicepa:  attached 21 volumes;
                        0 volumes not attached
Sun Nov 4 04:00:40 1990 File Server started Sun Nov 4 04:00:40
                        1990
Mon Nov 5 21:45:06 1990 CB: RCallBack (zero fid probe in host.c)
                        failed for host 28cf37c0.22811
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList* on **-server**.

bos getrestart — list automatic restart times for server processes.

bos getrestart -server <*machine name*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos getr -s <*machine name*> **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Displays two "restart times" from */usr/afs/local/BosConfig* on the file server machine *machine name*:

- The time each week that the BOS Server process automatically restarts itself and all processes that have status flag `Run` in *BosConfig*.
- The time each day that the BOS Server automatically restarts any process for which the time stamp on the binary file in */usr/afs/bin* is later than the last restart time for the process.

ARGUMENTS

-server	names the file server machine on which to check the restart times.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

The output consists of two lines:

```
Server machine name restarts at time
Server machine name restarts for new binaries at time
```

Possible values for *time* include:

- `never`, indicating that the BOS Server will never perform that type of restart.
- `now`, indicating that the BOS Server performs that type of restart each time it restarts and reads *BosConfig*, but not in between.
- A specified day and time, indicating that the BOS Server performs that type of restart once per week. Example: `sun 4:00 am`

- A specified time, indicating that the BOS Server performs that type of restart once per day. Examples: `11:00 pm`, `3:00 am`

The word `every` or `at` may precede a time or day/time definition. It does not change the function.

EXAMPLES

The following displays the restart times defined for *db2.transarc.com*:

% bos getr db2.transarc.com
```
Server db2.transarc.com restarts at sun 4:00 am
Server db2.transarc.com restarts for new binaries at 2:15 am
```

In the following example, the issuer abbreviates *fs1.transarc.com* to *fs1*, relying on the cell's name server to complete the name correctly. The output echoes the abbreviated form.

% bos getr fs1
```
Server fs1 restarts at sat 5:00 am
Server fs1 restarts for new binaries at 11:30 pm
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

bos setrestart

bos help — show syntax of specified **bos** command(s) or list functional description for all of them.

bos help [-topic <*help string*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos h [-t <*help string*>+**] [-h]**

DESCRIPTION

Displays the first line (name and short description) of every **bos** command's online help entry, if no *help string* is provided. For each operation code specified with **-topic**, it outputs the entire help entry. See the OUTPUT section below.

ARGUMENTS

-topic	specifies the operation code(s) for which the syntax is to be displayed. If it is omitted, the output instead provides a short description of all **bos** commands.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

The online help entry for each **bos** command consists of two or three lines.

- The first line names the command and briefly describes what it does.
- If the command has aliases, they will appear on the next line.
- The final line, which begins with "`Usage:`", lists the command's arguments and flags in the prescribed order. Online help entries use the same symbols (brackets, etc.) as the command definitions in this manual. For an explanation of their meaning, see page x of the introductory *About This Manual* chapter.

EXAMPLES

The following displays the online help entry for the **bos status** command.

```
% bos help status
bos status: show server instance status
Usage: bos status -server <machine name> [-instance <server
process name>+] [-long] [-cell <cell name>] [-noauth]
[-localauth] [-help]
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

bos apropos

bos install — install a new version of a binary file.

bos install -server <*machine name*> **-file** <*files to install*>+
[-dir <*destination dir*>**] [-cell** <*cell name*>**] [-noauth]**
[-localauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

bos i -s <*machine name*> **-f** <*files to install*>+ **[-d** <*destination dir*>**]**
[-c <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Installs the specified *files to install* as the new version of a process's binaries into the */usr/afs/bin* directory on the file server machine *machine name*, which should be the binary distribution machine for its CPU/operating system type. The UNIX mode bits of each newly installed file will be set to the setting of the *files to install*; in other words, the mode bit settings will be preserved in the transfer.

The command automatically saves the currently used version by adding a *.BAK* extension to its name. If there is a current *.BAK* version at least seven days old, it replaces the current *.OLD* version. If there is no current *.OLD* version, the current *.BAK* becomes *.OLD* automatically.

ARGUMENTS

-server	names the file server machine on which to install the new binaries. It should be the distribution machine for machines of its CPU/operating system type. If it names a machine other than a distribution machine, and the machine is running an *upclientbin* process, then the changes made will be overwritten the next time this command is directed to the distribution machine.
-file	gives the complete pathname for each binary file to be installed. Each pathname should specify a source directory other than */usr/afs/bin*, but the last element (the filename) should match the file in */usr/afs/bin* that this file is replacing (for example, it should be *ptserver* for the Protection Server). See the description of **bos create** for a list of the standard binary file names.
-dir	provides a complete directory pathname. It is necessary only if the binaries should be installed in a directory other than */usr/afs/bin*. All *files to install* will go into this one directory.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following would be appropriate in the Transarc cell. It installs the contents of the file */afs/transarc.com/rt_aix221/usr/afs/bin/vlserver* into */usr/afs/bin/vlserver* on *fs3.transarc.com*, which is the binary distribution machine for file server machines running AIX 2.2.1. The current version of */usr/afs/bin/vlserver* would be moved to */usr/afs/bin/vlserver.BAK* and so on.

% bos install fs3.transarc.com /afs/transarc.com/rt_aix221/usr/afs/bin/vlserver

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos getdate

bos getrestart

bos install

bos restart

bos setrestart

bos listhosts — display the contents of a machine's */usr/afs/etc/CellServDB*.

bos listhosts -server <*machine name*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos listh -s <*machine name*> **[-c** <*cell name*>**] [-n] [-l] [-h]**

bos getcell -s <*machine name*> **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Formats and displays information about a cell's database server machines from the */usr/afs/etc/CellServDB* file stored on the file server machine *machine name*. See section 2.1.2 for more information on this file.

ARGUMENTS

-server	names the file server machine from which to display the *CellServDB*. If changes to the file were made properly, then the output should be the same on every file server machine.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

The first line of the output names the cell to which the file server machine belongs. Each of the following lines names a database server machine for that cell.

The "`Host`" number assigned to each database server machine is for server-internal use only and is not the same as, nor necessarily related to, the machine's Internet address. It was assigned automatically by the **bos addhost** command.

EXAMPLES

The following displays the database server machines listed in *CellServDB* on *fs7.transarc.com*.

```
% bos listhosts fs7.transarc.com
Cell name is transarc.com
    Host 1 is db1.transarc.com
    Host 2 is db2.transarc.com
    Host 3 is db3.transarc.com
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

bos addhost

bos removehost

bos listkeys — display the server encryption keys from */usr/afs/etc/KeyFile*.

bos listkeys -server <*machine name*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos listk -s <*machine name*> **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Formats and displays information about server encryption keys kept in the file */usr/afs/etc/KeyFile* stored on the file server machine *machine name*. See the OUTPUT section for details on the display format and type of information. See section 2.1.3 for more information on the *KeyFile* file.

ARGUMENTS

-server	names the file server machine from which to display the *KeyFile*. If changes to the file were made properly, then the output should be the same on every server machine.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

The output includes one line for each server encryption key found in the *KeyFile*, identified by key version number.

If authorization checking is disabled on *machine name*, the string of octal numbers making up the key appears; **bos setauth** is used to disable authorization checking. There are two reasons for this. First, it implies that only someone authorized to issue the **bos setauth** command or with "root" access to *machine name*'s local disk (presumably an administrator) is able to see keys from the Authentication Database. Second, it makes it clear that the system is in a compromised state of security while keys are being examined (both turning off authorization checking and displaying keys on a screen are serious security risks).

The string of octal numbers you see when authorization checking is disabled may also include regular ASCII characters, because some octal numbers correspond to the codes for ASCII characters. There should be a total of eight octal numbers and/or characters in each key; each octal number is represented as a backslash and 3 digits.

If authorization checking is enabled on *machine name* (the normal case), a "checksum" appears, which is a decimal number derived by encrypting a constant with the key. The output of **bos listkeys** should match that of **kas examine afs** whether authorization checking is on or off.

This number should match the checksum with the same key version number that appears when the **kas examine** is used to examine the "*afs*" entry (also with authorization checking enabled).

Following the list of keys or checksums, a line specifies when a key was last added to the *KeyFile*. The words `All done` indicate the end of the output.

EXAMPLES

The following shows the checksums for the keys stored in *KeyFile* on *fs3.transarc.com*. They appear instead of the actual keys because authorization checking has not been disabled.

```
% bos listk fs3.transarc.com
key 1 has cksum 972037177
key 3 has cksum 2825175022
key 4 has cksum 260617746
key 6 has cksum 4178774593
Keys last changed on Wed Jan 3 11:24:46 1990.
All done.
```

The following shows the keys found in *KeyFile*, because authorization checking has been disabled on *fs6.transarc.com*. Note the letter `j` in key 2.

```
% bos listk fs6.transarc.com
key 0 is '\040\205\211\241\345\002\023\211'
key 1 is '\343\315\307\227\255\320\135\244'
key 2 is '\310\310\255\253j\236\261\211'
Keys last changed on Wed Jan 3 11:24:46 1990.
All done.
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList* on **-server** to list checksums. This is the only **bos** listing-only command that is privileged.

To see the actual keys, no privilege is required, but authorization checking must be disabled with **bos setauth**, which also requires being listed in */usr/afs/etc/UserList* and having "root" access on **-server**.

MORE INFORMATION

bos addkey

bos removekey

kas examine

bos listusers — list the privileged users from */usr/afs/etc/UserList*.

bos listusers -server <*machine name*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos listu -s <*machine name*> **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Lists the user names of the privileged users found in the file */usr/afs/etc/UserList* stored on the file server machine *machine name*. See section 2.1.4 for more information on the *UserList* file.

ARGUMENTS

-server	names the file server machine from which to display the *UserList*. If changes to the file were made properly, then the output should be the same on every file server machine.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

The output lists the user name of each user entitled to issue privileged **bos** and **vos** commands.

EXAMPLES

The following lists the privileged users found in the *UserList* kept on *fs4.transarc.com*. The issuer chooses to abbreviate the machine name.

```
% bos listusers fs4
SUsers are: pat smith jones terry
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

bos adduser

bos removeuser

bos prune — remove obsolete versions of files from */usr/afs/bin* and */usr/afs/logs*.

bos prune -server <*machine name*> **[-bak] [-old] [-core] [-all] [-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos p -s <*machine name*> **[-b] [-o] [-co] [-a] [-ce** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Removes certain obsolete files, as specified by the flags provided on the command line:

- All files in */usr/afs/bin* with a `.BAK` extension (with **-bak** flag).
- All files in */usr/afs/bin* with a `.OLD` extension (with **-old** flag).
- All files in */usr/afs/logs* with a `core.` prefix (with **-core** flag).
- All three types at once (with **-all** flag).

ARGUMENTS

-server	names the file server machine from which to remove the indicated files.
-bak	removes all files with a `.BAK` extension from the */usr/afs/bin* directory. It may be combined with the other flags except **-all**.
-old	removes all files with a `.OLD` extension from the */usr/afs/bin* directory. It may be combined with the other flags except **-all**.
-core	removes all core files from the */usr/afs/logs* directory. It may be combined with the other flags except **-all**.
-all	combines the effect of the other three flags. Provide it by itself.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following removes all files with *.BAK* and *.OLD* extensions from the */usr/afs/bin* directory on *fs3.transarc.com*.

% bos p fs3.transarc.com -b -o

The following removes all files with *.BAK* and *.OLD* extensions from the */usr/afs/bin* directory, and all files that begin with "*core.*" from the */usr/afs/logs* directory, on *db2.transarc.com*. The issuer chooses to abbreviate the machine name.

% bos prune db2 -all

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos getdate

bos removehost — remove a database server machine from */usr/afs/etc/CellServDB*.

bos removehost -server <*machine name*> **-host** <*host name*>+
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos removeh -s <*machine name*> **-ho** <*host name*>+ **[-c** <*cell name*>**] [-n] [-l] [-he]**

DESCRIPTION

Removes each specified database server machine (*host name*) from the */usr/afs/etc/CellServDB* file on *machine name*. See section 2.1.2 for more information on the *CellServDB* file.

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, change *CellServDB* only on the cell's system control machine (specify it as *machine name*). By default, it takes up to five minutes for the changed file to propagate to the other file server and database server machines via *upserver* and *upclientetc*.

If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc*. Instead, repeat this command for all of your cell's file server and database server machines, specifying each one as *machine name* in turn. To avoid possible inconsistency problems, finish issuing the commands within a fairly short time (within about five minutes, the default time period within which the Update Server distributes the changes in cells running the United States edition of AFS).

WARNING

After executing this command, the issuer must restart the database server processes on all database server machines. Otherwise, the lightweight process that each such process runs to handle replication (Ubik) will not notice the change. See the section in the *AFS Installation Guide* on adding and removing database server machines.

ARGUMENTS

-server names the file server machine on which to change *CellServDB*. With the United States edition of AFS, it should be the cell's system control machine so that the changes propagate to all file server and database server machines in the cell. If the specified machine is not the system control machine but is running an *upclientetc* process, then the changes will be overwritten when that process next brings over *CellServDB* from the system control machine (by default, within five minutes).

With the international edition of AFS, specify the name of each file server and database server machine in the cell in turn.

-host specifies the complete machine name (such as *fs2.transarc.com*) of each database server machine to be removed from *CellServDB*.

-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following removes the former database server machine *db2.transarc.com* from the *CellServDB* file kept on *fs1.transarc.com* (the cell's system control machine).

% **bos removeh fs1.transarc.com db2.transarc.com**

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos addhost

bos listhosts

AFS Installation Guide

bos removekey — remove a server encryption key from */usr/afs/etc/KeyFile*.

bos removekey -server <*machine name*> **-kvno** <*key version number*>$^{+}$
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos removek -s <*machine name*> **-k** <*key version number*>$^{+}$ **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Removes each specified encryption key (*key version number*) from the */usr/afs/etc/KeyFile* file stored on *machine name*. See section 2.1.3 for more information on the *KeyFile* file.

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, change *KeyFile* only on the cell's system control machine (specify it as *machine name*). By default, it takes up to five minutes for the changed file to propagate to the other file server and database server machines via *upserver* and *upclientetc*.

If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc*. Instead, repeat this command for all of your cell's file server and database server machines, specifying each one as *machine name* in turn. To avoid possible inconsistency problems, finish issuing the commands within a fairly short time (within about five minutes, the default time period within which the Update Server distributes the changes in cells running the United States edition of AFS).

ARGUMENTS

-server names the file server machine on which to change *KeyFile*. With the United States edition of AFS, it should be the cell's system control machine so that the changes propagate to all file server and database server machines in the cell. If the specified machine is not the system control machine but is running an *upclientetc* process, then the changes will be overwritten when that process next brings over the *KeyFile* from the system control machine (by default, within five minutes).

With the international edition of AFS, specify the name of each file server and database server machine in the cell in turn.

-kvno specifies the key version number of each key to be removed from the *KeyFile*.

-cell specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.

-noauth tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following removes the keys with key version numbers 5 and 6 from the *KeyFile* file on *fs1.transarc.com* (the cell's system control machine). The issuer chooses to abbreviate the machine name.

% bos removek fs1 5 6

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos addkey

bos listkeys

bos removeuser — remove a privileged user from */usr/afs/etc/UserList*.

bos removeuser -server <*machine name*> **-user** <*user names*>+ **[-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos removeu -s <*machine name*> **-u** <*user names*>+ **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Removes each specified *user names* from the */usr/afs/etc/UserList* file stored on *machine name*. See section 2.1.4 for more information on the *UserList* file.

If your cell runs the United States edition of AFS and you use the Update Server to distribute the contents of */usr/afs/etc*, change *UserList* only on the cell's system control machine (specify it as *machine name*). By default, it takes up to five minutes for the changed file to propagate to the other file server and database server machines via *upserver* and *upclientetc*.

If you are running the international edition of AFS, you should not use the Update Server to distribute */usr/afs/etc*. Instead, repeat this command for all of your cell's file server and database server machines, specifying each one as *machine name* in turn. To avoid possible inconsistency problems, finish issuing the commands within a fairly short time (within about five minutes, the default time period within which the Update Server distributes the changes in cells running the United States edition of AFS).

ARGUMENTS

-server names the file server machine on which to change *UserList*. With the United States edition of AFS, it should be the cell's system control machine so that the changes propagate to all file server and database server machines in the cell. If the specified machine is not the system control machine but is running an *upclientetc* process, then the changes will be overwritten when that process next brings over the *UserList* from the system control machine (by default, within 5 minutes).

With the international edition of AFS, specify the name of each file server and database server machine in the cell in turn.

-user specifies the user names to be removed from the *UserList*.

-cell specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.

-noauth tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following removes the former privileged users *pat* and *jones* from *UserList* on *fs1.transarc.com* (the cell's system control machine).

% bos removeu fs1.transarc.com pat jones

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos adduser

bos listusers

bos restart — restart process(es) on a file server machine.

bos restart -server <*machine name*> **[-instance** <*instances*>+**] [-bosserver] [-all] [-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos res -s <*machine name*> **[-i** <*instances*>+**] [-b] [-a] [-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Instructs the BOS Server running on *machine name* to stop and immediately restart the appropriate process(es) when directed by one of the following:

- The **-instance** argument instructs the BOS Server to stop each specified process on *machine name* and then immediately restart each specified process (regardless of its status flag setting). Do not include *bosserver* in the list of processes; use the **-bosserver** flag instead.
- The **-all** flag instructs the BOS Server to stop all processes running on *machine name*, other than itself, and then immediately restart each stopped process that is marked with the status flag Run in */usr/afs/local/BosConfig*.
- The **-bosserver** flag instructs the BOS Server to stop all processes on *machine name*, including itself. A new BOS Server starts immediately, and it restarts a new instance of each stopped process that is marked with the status flag Run in */usr/afs/local/BosConfig*.

This command does not change a process' status flag in *BosConfig*. Remember, use only one of the above three argument or flags when specifying the processes that you want to stop and restart.

ARGUMENTS

-server	names the file server machine on which to restart the indicated processes.
-instance	specifies each process that should be stopped and then restarted (regardless of its status flag setting). This argument, the **-bosserver** flag, or the **-all** flag must be specified with this command. Specify a process using the same name that was used to create it with the **-instance** argument of the **bos create** command. If you cannot recall the appropriate name, run the **bos status** command.
-bosserver	indicates that the current BOS Server instance should stop all processes running on *machine name*, including itself. A new BOS Server instance immediately starts up, and restarts all processes marked with the status flag Run in the *BosConfig* file. This flag, the **-instance** argument, or the **-all** flag must be specified with this command.
-all	directs the BOS Server to stop all processes running on *machine name*, other than itself, and then immediately restart those processes marked with the status flag Run in the *BosConfig* file. This flag, the **-instance** argument, or the **-bosserver** flag must be specified with this command.

-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following stops all processes running on *fs3.transarc.com*, including the BOS Server. A new BOS Server instance starts up immediately and restarts all processes marked with status flag Run in *BosConfig*.

% bos res fs3.transarc.com -b

The following instructs the BOS Server to stop all processes running on *fs5.transarc.com*, excluding itself. The BOS Server then restarts all processes marked with status flag Run in *BosConfig*.

% bos restart fs5 -a

The following instructs the BOS Server to stop and restart the *ptserver* (Protection Server) and the *vlserver* (Volume Server) processes on *db3.transarc.com*:

% bos res db3.transarc.com ptserver vlserver

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos create

bos status

bos salvage — restore internal consistency to file system or volume.

bos salvage -server <*machine name*> **[-partition** <*salvage partition*>**]**
[-volume <*salvage volume number or volume name*>**]**
[-file <*salvage log output file*>**] [-all] [-showlog]**
[-parallel <*# of max parallel partition salvaging*>**]**
[-tmpdir <*directory to place tmp files*>**] [-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

bos sa -se <*machine name*> **[-part** <*salvage partition*>**]**
[-v <*salvage volume number or volume name*>**]**
[-f <*salvage log output file*>**] [-a] [-sh]**
[-para <*# of max parallel partition salvaging*>**]**
[-t <*directory to place tmp files*>**] [-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Salvages (restores internal consistency in) one or more volumes, depending on how the command's arguments are combined. For instance, to salvage

- All volumes on a file server machine, provide the **-server** argument and the **-all** flag. A file system outage results because the BOS Server stops the File Server and Volume Server components while the Salvager is running.
- All the volumes on one partition, provide **-server** and **-partition**. A file system outage results because the BOS Server stops the File Server and Volume Server components while the Salvager is running.
- Only one volume, combine **-server**, **-partition**, and **-volume**. Only the specified volume goes off-line during the salvage. The BOS Server does not need to shutdown the File Server and Volume Server during this type of salvage, so there is no outage.

By default, up to four Salvager processes execute to salvage four or more partitions; if fewer than four partitions are being salvaged, only that number of Salvager processes execute. To increase or decrease the number of Salvager processes that execute, use the **-parallel** argument to specify an integer value between *1* and *32*. Specify a value of *1* to run the Salvager processes sequentially (that is, one partition at a time). The Salvager will execute only the number of processes required to salvage the specified partitions; if a single volume is specified, only one Salvager process will be executed.

By default, the Salvager writes its output to the */usr/afs/logs/SalvageLog* file. By using the **-file** argument, you can specify the name and destination of the salvage log file. And by using the **-showlog** flag, you can also view this output on the screen (as it is written to the log file). However, **-file** and **-showlog** cannot be used together.

As a Salvager process salvages a partition, it can create a large number of *tmp* files. By default, these files are written to the partition that the process is currently salvaging. If there is not enough disk space

on a particular partition to store the *tmp* files, use the **-tmpdir** argument to redirect these tmp files to a directory that has enough storage space. Should a Salvager process run out of space to write its *tmp* files, it stops salvaging the partition and terminates; any remaining partitions are salvaged by the remaining Salvager processes. Any partitions that remain unsalvaged because of this problem can be salvaged by re-executing the command with the **-tmpdir** argument and specifying a directory that contains sufficient space to store the *tmp* files. The Salvager automatically removes its *tmp* files from the full partition.

During the salvage, the output of a **bos status** command on the *fs* process reports an auxiliary status of "`Salvaging file system`." When the salvage is complete, the BOS Server restarts the File Server and Volume Server components of the *fs* process.

Note: There is no limit to the number of volumes that the Salvager can salvage on a single partition.

WARNING

Running this command can result in loss of data if the Salvager can repair corruption only by removing the offending data. Before issuing this command, the issuer may wish to read the section in the *AFS System Administrator's Guide* about salvaging volumes.

ARGUMENTS

-server	names the file server machine on which to salvage volumes.
-partition	specifies a partition where volumes should be salvaged as necessary. Provide it alone or in combination with **-volume**. See page xv to learn the legal ways to refer to partitions.
-volume	specifies the name or volume ID number of the volume to be salvaged. The volume may be of type ReadWrite, ReadOnly, or Backup. The partition housing the volume must be indicated with **-partition**.
-file	specifies the complete pathname of the file into which a log of the salvage will be written. If this is omitted, the output goes into the file */usr/afs/logs/SalvageLog*. If salvaging a partition or entire machine, indicate a pathname of a file on the machine's local disk, since the file system is not accessible during the salvage. Do not use this argument with the **-showlog** flag.
-all	indicates that all of the volumes on all of the partitions of the file server machine should be salvaged.
-showlog	instructs the Salvager to display on the screen all log data that is being written to the */usr/afs/logs/SalvageLog* file. Do not use this flag with the **-file** argument.
-parallel	specifies the maximum number of Salvager processes to be run in parallel. By default, four Salvager processes run if four or more partitions require salvaging; if fewer than four partitions require salvaging, a Salvager process will be run for each partition. Use this argument to increase or decrease this default number to a value between *1* and *32*. Specifying *1* runs the Salvager process sequentially; specifying a number greater than the number of partitions requiring salvage is unnecessary, because the Salvager runs only the number of processes required to salvage the specified number of partitions.

-tmpdir	redirects the write destination of the *tmp* files created by the command to the specified *directory to place tmp files* on the machine being salvaged. If this argument is not used, the *tmp* files are placed on the partition being salvaged. If the *directory to place tmp files* is incorrectly specified or is not on the machine being salvaged, the *tmp* files are placed on the partition being salvaged.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

The Salvager produces a log or trace of its actions, which it normally sends to */usr/afs/logs/SalvageLog*. To change the name or destination of the trace file, use the **-file** argument; to direct the log to standard output (in addition to */usr/afs/logs/SalvageLog*) use **-showlog**. Do not use **-file** and **-showlog** together.

EXAMPLES

The following salvages all volumes on all partitions of *fs4.transarc.com*. The issuer chooses to abbreviate the machine name as well as the operation code and **-all** flag.

% bos sa fs4 -a

The following salvages all volumes on the */vicepd* partition of *db3.transarc.com*:

% bos salvage db3.transarc.com /vicepd

The following salvages the volume with volume ID number 536870988 on */vicepb* of *fs2.transarc.com*:

% bos sa fs2 b 536870988

The following salvages all volumes on all partitions of *fs4.transarc.com*. To speed up the salvage task, the issuer executes six Salvager processes in parallel (the issuer knows that there are 12 partitions on *fs4.transarc.com*), rather than rely on the four that would normally execute.

% bos sa fs4 -a -para 6

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

salvager

bos setauth — set authorization checking requirements for all server processes.

bos setauth -server <*machine name*>
-authrequired <*on or off: authentication required for admin requests*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos seta -s <*machine name*>
-a <*on or off: authentication required for admin requests*>
[-c <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Determines whether or not all of the server processes running on *machine name* check to see if the issuer of a command is authorized to execute it (has the required privilege). If this "authorization checking" is disabled, then the server will perform any action for anyone, even *anonymous*, which is a grave security risk. Authorization checking on a machine should almost always be enabled, except during installation or emergencies.

To indicate to other AFS server processes that authorization checking has been disabled, the BOS Server creates the zero-length file */usr/afs/local/NoAuth* on its local disk. All AFS server processes constantly monitor for *NoAuth*'s presence and do not check for authorization when it is present. To indicate that authorization checking is enabled (the normal state), BOS Server removes the file.

WARNINGS

After executing this command, the issuer must restart all server processes on the file server machine. The server processes automatically obey the new authorization checking requirements in making connections with clients. However, the server processes also have established connections among themselves. For instance, the File Server keeps open a connection with the Protection Server so that it can request a current protection subgroup (CPS) for new users without having to establish a new connection each time. The Ubik lightweight processes for a database server process all have connections open to one another as well. The issuer needs to restart the servers to make sure that they destroy these connections and open new ones that obey the current authorization requirements.

Administrators should not create *NoAuth* directly, except when directed by instructions for dealing with emergencies (doing so requires logging into the machine's UNIX file system as "root"). Use this command instead.

ARGUMENTS

-server is the file server machine on which the state of authorization checking is to change.

-authrequired sets whether authorization checking is enabled or disabled. The legal values are *on*, which enables checking, and *off*, which disables it.

-cell specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.

-noauth tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following turns off authorization checking for all server processes on *fs7.transarc.com*:

% bos seta fs7.transarc.com off

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos restart

bos setcellname — sets the cell's name in */usr/afs/etc/ThisCell* and */usr/afs/etc/CellServDB*.

bos setcellname -server <*machine name*> **-name** <*cell name*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos setc -s <*machine name*> **-n** <*cell name*> **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Sets the name of the cell to which the file server machine *machine name* belongs, by recording *cell name* (the one following the **-name** argument) in */usr/afs/etc/ThisCell* on *machine name*'s local disk.

This command also creates the file */usr/afs/etc/CellServDB* on *machine names*'s local disk. It puts the specified *cell name* on the first line of the file, and also creates an entry naming *machine name* as the cell's first database server machine.

WARNING

This command should be issued only during the installation of a cell's first file server machine (the system control machine), and not at any other time. The other file server machines will use the same cell name automatically because *ThisCell* and *CellServDB* are in the */usr/afs/etc* directory exported by server portion of the Update Server (*upserver* process) running on the system control machine.

Changing a cell's name after setting it on the first installed machine is not recommended. It requires that the system administrator change every password in the Authentication Database. If you change your cell's name, contact your AFS Product Support Representative for complete instructions.

ARGUMENTS

-server	names the file server machine on which to set the cell name in *ThisCell* and *CellServDB*. It should always be the first machine installed in a cell (the system control machine).
-name	defines the cell's name. It should agree with the name specified as the **-cell** argument, if that is provided. The cell name should follow Internet format for domain names. Examples are "transarc.com" for the Transarc Corporation cell and "andrew.cmu.edu" for the main Andrew cell at Carnegie Mellon University. If the new cell is not an Internet site, the choice is not as restricted, but must not duplicate an existing official Internet domain name.
-cell	specifies the cell to which **-server** belongs. It should agree with the name specified as the **-name** argument. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.

-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following defines the cell name *transarc.com* in the *ThisCell* and *CellServDB* files on *fs1.transarc.com* as it is installed as the cell's first file server machine. The machine itself is recorded in *CellServDB* as the cell's first database server machine.

% bos setcellname fs1.transarc.com transarc.com

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server** (unless authorization checking is turned off, as it normally is during installation).

MORE INFORMATION

See the *AFS Installation Guide* for more information on choosing a cell name.

bos setrestart — set the date/time at which BOS Server restarts all processes or those with new binaries.

bos setrestart -server <*machine name*> **-time** <*time to restart server*> **[-general] [-newbinary] [-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos setr -s <*machine name*> **-t** <*time to restart server*> **[-g] [-ne] [-c** <*cell name*>**] [-no] [-h]**

DESCRIPTION

Sets the times at which the BOS Server running on the file server machine *machine name* performs two types of restarts, by recording the time settings in */usr/afs/local/BosConfig*. The two types of restart times are:

1. the time each week at which the BOS Server restarts itself and then any AFS process marked with status flag Run in *BosConfig*. This is equivalent to executing **bos restart** with the **-bosserver** flag. The default setting is 4:00 a.m. each Sunday morning.
2. the time each day at which the BOS Server restarts any process currently running for which the binary file in */usr/afs/bin* was modified since the process last re/started. The default is 5:00 a.m. each day.

A single issue of the command can change only one of the times, so changing both times requires issuing the command twice.

WARNINGS

Restarting a process makes it unavailable for a period of time. The longest outage results from restarting the *fs* process; it can be up to a half hour, depending on how many volumes the File Server and Volume Server have to reattach. The issuer will probably want to set the restart times at times of typically low usage so that the outage inconveniences as few users as possible; this is the reasoning behind the defaults as well.

If the specified time is within one hour of the current time, the BOS Server will not restart the process(es) until that time the next day.

ARGUMENTS

-server	specifies the file server machine on which */usr/afs/local/BosConfig* should change to reflect the new restart times.
-time	specifies the restart time.

When setting the weekly restart time, it is standard to specify both a day and a time, in that order. (If the issuer wants to increase the frequency of automatic restarts to every day, he or she may specify only a time.) Separate the day and the time with a space and enclose the entire definition in quotes.

Type the name of the day in all lowercase letters, and give either the whole name or the first three letters as an abbreviation (*sunday* or *sun*, *thursday* or *thu*, and so on). It is not possible to include more than one day.

The time of day may be expressed in several ways. In all cases, separate the hours from the minutes with a colon (for instance, *5:00*). Twenty-four hour time is assumed, but twelve hour time (using only *1:00* through *12:00*) is acceptable if *a.m.* (or *am*) and *p.m.* (or *pm*) are used to distinguish times before and after noon.

When setting the daily binary checking time, it is standard to specify only a time of day. (If the issuer wants to reduce the frequency of binary checking down to once per week, he or she may specify both a day and a time.) Use the conventions described in the previous paragraph to express the time of day, and enclose it in quotes.

The issuer may precede either type of time definition with the word *every* or *at*. These words add no function, but may make the output of **bos getrestart** more understandable.

Also, the issuer may use two special-case definitions instead of a day/time:

- *never* indicates that the BOS Server should never perform the indicated type of restart.
- *now* indicates that the BOS Server should perform the indicated restart immediately and not again.

-general sets the weekly restart time.

-newbinary sets the daily restart time.

-cell specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.

-noauth tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following defines a restart time in *BosConfig* on *fs4.transarc.com* that causes all processes on that machine to stop and restart each Saturday morning at 3:30 am. The issuer chooses to abbreviate the **-general** flag.

% bos setr fs4.transarc.com "sat 3:30" -g

The following defines a restart time in *BosConfig* on *fs6.transarc.com*. Each evening at 11:45 pm, the BOS Server on that machine will check for new binary files in */usr/afs/bin* and restart any processes for which it finds a new file. The issuer chooses to shorten the machine name and the **-newbinary** flag.

% bos setr fs6 "23:45" -ne

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos getrestart

bos restart

bos shutdown — stop process(es) without changing status flag(s) in *BosConfig*.

bos shutdown -server <*machine name*> **[-instance** <*instances*>+**] [-wait] [-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos sh -s <*machine name*> **[-i** <*instances*>+**] [-w] [-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Instructs the BOS Server running on the file server machine *machine name* to stop either all AFS processes (except itself) running on the machine or only the processes listed as *instances*. The BOS Server cannot be stopped using this command. It is appropriate to run this command before halting or rebooting a file server machine.

This command does not change a process' status flag in */usr/afs/local/BosConfig*, but only in the BOS Server's memory state. Use the **bos stop** command to change the status flag.

Processes stopped with this command will not run again until a system administrator starts them using **bos start**, **bos startup**, or **bos restart**, or until the BOS Server itself restarts.

ARGUMENTS

-server	names the file server machine on which processes should stop.
-instance	specifies which process(es) should be stopped. If omitted, then all processes running on the file server machine except the BOS Server are stopped. Refer to a process by the name assigned with the **-instance** argument in the **bos create** command; if the issuer has forgotten the appropriate name, it appears in the output from the **bos status** command.
-wait	indicates that the command shell prompt should not return until the shutdown is complete (all processes actually stop running). Otherwise, the prompt returns almost immediately, and the shutdown may not be complete at that point.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following instructs the BOS Server running on *fs3.transarc.com* to stop running all processes except itself.

% bos shutdown fs3.transarc.com

The following instructs the BOS Server running on *fs5.transarc.com* to stop running the *upserver* (server portion of the Update Server). The issuer chooses to shorten the machine name.

% bos sh fs5 upserver

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos create

bos status

bos start — start process(es) after setting status flag(s) to Run in *BosConfig*.

bos start -server <*machine name*> **-instance** <*server process name*>+
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos start -s <*machine name*> **-i** <*server process name*>+ **[-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Sets the status flag for each specified process *server process name* to Run in */usr/afs/local/BosConfig* on the file server machine *machine name*, then starts each process running on that machine. If an instance of the process is already running, the only effect of this command is to guarantee that the status flag gets set to Run; it does not start up a new instance of the process.

ARGUMENTS

-server	names the file server machine where processes should be started.
-instance	specifies for which process(es) the *BosConfig* flag should be set to Run before the process starts running on the machine. Refer to a process by the name assigned with the **-instance** argument in the **bos create** command; if the issuer has forgotten the appropriate name, it appears in the output from the **bos status** command.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following changes the status flag for the *upclientbin* and *upclientetc* processes to Run in *BosConfig* on *fs6.transarc.com* and starts the processes running.

% bos start fs6.transarc.com upclientbin upclientetc

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos create

bos status

bos startup — start process(es) without changing status flag(s) in *BosConfig*.

bos startup -server <*machine name*> **[-instance** <*instances*>+**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos startu -s <*machine name*> **[-i** <*instances*>+**] [-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Instructs the BOS Server running on the file server machine *machine name* to start *either* all AFS processes marked with status flag `Run` in */usr/afs/local/BosConfig* that are not running *or* each process specified with **-instance**, even if its status flag in *BosConfig* is `NotRun`.

This command does not change a process' status flag in *BosConfig*. Use the **bos start** command for that.

ARGUMENTS

-server	names the file server machine on which to start process(es)
-instance	specifies which process(es) to start running. This flag is useful for testing server processes without enabling them permanently. Refer to a process by the name assigned with the **-instance** argument in the **bos create** command; if the issuer has forgotten the appropriate name, it appears in the output from the **bos status** command.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following causes the BOS Server on *fs3.transarc.com* to start all processes marked with status flag `Run` in *BosConfig* that are not currently running.

% bos startup fs3.transarc.com

The following causes the BOS Server on *db2.transarc.com* to start the *kaserver*, *ptserver* and *vlserver* processes even if their status flags *BosConfig* are `NotRun`. The issuer chooses to abbreviate the machine name.

% bos startup db2 vlserver ptserver kaserver

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos create

bos status

bos start

bos restart

bos status — list statuses of processes on a file server machine.

bos status -server <*machine name*> **[-instance** <*server process name*>**+] [-long] [-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos stat -s <*machine name*> **[-i** <*server process name*>**+] [-lon] [-c** <*cell name*>**] [-n] [-loc] [-h]**

DESCRIPTION

Lists the status of either all the processes on the file server machine called *machine name* or only the processes listed with **-instance**.

The BOS Server also notes whether certain files and directories under */usr/afs* on the local disk have the mode bits it expects. If they do not and the **-long** flag is used, a warning message appears.

ARGUMENTS

-server	names the file server machine for which process status is to be shown.
-instance	specifies the process(es) for which status information is to be shown. Refer to a process by the name assigned with the **-instance** argument in the **bos create** command; if the issuer has forgotten the appropriate name, the **bos status** command can be used to display it.
-long	gives detailed information about the status of each specified process. See the OUTPUT section for details.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

OUTPUT

If the **-long** flag is used and the BOS Server has found that the mode bits on certain files and directories under */usr/afs* deviate from what it expects, it prints the following warning message:

```
Bosserver reports inappropriate access on server directories
```

The correct protections for the directories and files under */usr/afs* are

/usr/afs	`drwxr-xr-x`
/usr/afs/backup	`drwx------`
/usr/afs/bin	`drwxr-xr-x`
/usr/afs/db	`drwx------`
/usr/afs/etc	`drwxr-xr-x`
/usr/afs/etc/KeyFile	`-rw-------`
/usr/afs/etc/UserList	`-rw-------`
/usr/afs/local	`drwx------`
/usr/afs/logs	`drwxr-xr-x`

See the *AFS Installation Guide* for more information about setting the protections on these files and directories.

The command's output contains one entry for each process found in */usr/afs/local/BosConfig*. The first line of an entry shows the current status of the process. The possible statuses are

- `currently running normally`. The process's status flag in *BosConfig* is `Run`. For cron entries, this message indicates only that the command is still scheduled to be run, not necessarily that it was executing when the **bos status** command was issued.
- `disabled`. Indicates that the process is not and should not be running, because a system administrator has set its status flag in *machine name*'s *BosConfig* to `NotRun`.
- `temporarily disabled`. This indicates one of two things: either a system administrator stopped the process with the **bos shutdown** command, or the BOS Server ceased trying to restart the process after numerous failed attempts. In the second case, the auxiliary message "`stopped for too many errors`" appears. The status flag in *BosConfig* for the process is still `Run`.
- `temporarily enabled`. Indicates that a system administrator has started the process running using **bos startup**. Its status flag in *BosConfig* is still `NotRun`.

The following messages may appear on the next line of a process's status entry, but only in special circumstances:

- `has core file`. Indicates the process failed and created a core file in */usr/afs/logs*. The BOS Server may nevertheless have been able to restart it, so this message may appear with "`currently running normally`."
- `stopped for too many errors`. Indicates that the reason for the primary status of "`temporarily disabled`" is that the BOS Server tried several times to restart the process and always failed.

The output for the *fs* process always includes an "`Auxiliary status is:`" message, which may be one of

- `file server running`. The File Server and Volume Server components of the File Server process are running normally.
- `salvaging file system`. The Salvager is running, so the File Server and Volume Server are temporarily disabled. The BOS Server restarts them as soon as the Salvager is finished.

The output for a cron process includes an "`Auxiliary status`" that reports when the command will next execute.

If the **-long** flag is used, the output includes the following additional information:

- Each instance's type (`simple`, `fs`, or `cron`).
- How often the current BOS Server instantiation has started the process (the output refers to these as "`proc starts`").
- The time of the last proc start.
- The "exit" time when the process (or one of the processes in the *fs* process) last failed. This line does not appear if the process has not failed during the current instantiation of the BOS Server.
- The "error exit" time when the process (or one of the processes in the *fs* process) last failed due to an error. A further explanation such as "`due to shutdown request`" may appear. This line does not appear if the process has not failed during the current instantiation of the BOS Server.
- The commands the BOS Server runs to start the process (as specified on the parameter line of the command's *BosConfig* entry).

EXAMPLES

The following displays the run status of each process on *fs3.transarc.com*:

```
% bos stat fs3.transarc.com
Instance buserver, currently running normally.
Instance kaserver, currently running normally.
Instance ptserver, currently running normally.
Instance vlserver, currently running normally.
Instance fs, has core file, currently running normally.
    Auxiliary status is: file server running.
Instance upserver, currently running normally.
Instance runntp, currently running normally.
```

The following displays information from the *BosConfig* entries for the *fs* and *ptserver* processes on *fs1.transarc.com*, in addition to their run statuses. The issuer includes all arguments and flags, some in shortened form.

% bos stat -s fs1.transarc.com -i fs ptserver -long

```
Instance fs, (type is fs), currently running normally.
   Auxiliary status is: file server running.
   Process last started at Tue Jan 2 8:29:19 1990 (3 proc starts)
   Last exit at Tue Jan 2 8:29:19 1990
   Last error exit at Tue Jan 2 8:29:19 1990, due to shutdown
       request
   Command 1 is '/usr/afs/bin/fileserver'
   Command 2 is '/usr/afs/bin/volserver'
   Command 3 is '/usr/afs/bin/salvager'
Instance ptserver, (type is simple) currently running normally.
   Process last started at Wed Jan 3 5:34:49 1990 (1 proc starts)
   Command 1 is '/usr/afs/bin/ptserver'
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

bos create

bos shutdown

bos status

bos stop — stop process(es) after changing status flag(s) to `NotRun` in *BosConfig*.

bos stop -server <*machine name*> **-instance** <*server process name*>+ **[-wait] [-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos sto -s <***machine name***> **[-i** <*server process name*>+**] [-w] [-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Sets the status flag for each process *server process name* to `NotRun` in */usr/afs/local/BosConfig* on the file server machine *machine name*, then stops the process(es) from running on that machine.

ARGUMENTS

-server	names the file server machine on which to stop processes.
-instance	specifies for which process(es) the *BosConfig* flag should be set to `NotRun` before the process stops running on the machine. Refer to a process by the name assigned with the **-instance** argument in the **bos create** command; if the issuer has forgotten the appropriate name, it appears in the output from the **bos status** command.
-wait	indicates that the command shell prompt should not return until the stop is complete (all processes actually stop running). Otherwise, the prompt returns almost immediately, and the stop may not be complete at that point.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following sets the status flag for the *upserver* and *runntp* processes to `NotRun` in *BosConfig* on *fs7.transarc.com*, then stops them from running.

% bos sto fs7.transarc.com upserver runntp

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos create

bos status

bos uninstall — revert to using former version of binaries for a process.

bos uninstall -server <*machine name*> **-file** <*files to uninstall*>+ **[-dir** <*destination dir*>**] [-cell** <*cell name*>**] [-noauth] [-localauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

bos u -s <*machine name*> **-f** <*files to uninstall*>+ **[-d** <*destination dir*>**] [-c** <*cell name*>**] [-n] [-l] [-h]**

DESCRIPTION

Replaces the specified binary *files to uninstall* found in the */usr/afs/bin* directory with their *.BAK* versions. It also changes the extension on the current *.OLD* version (if any) to *.BAK*.

To make *machine name* start using the reverted binaries immediately, issue the **bos restart** command. Otherwise the binaries will not be used until the BOS Server restarts the affected process at the daily time specified in */usr/afs/local/BosConfig* (use the **bos getrestart** and **bos setrestart** commands to inspect and set this time).

ARGUMENTS

-server	names the file server machine on which to revert to the *.BAK* version of binaries. It should be the distribution machine for machines of its CPU/operating system type. If it names a machine other than a distribution machine, and the machine is running the *upclient* or *upclientbin* process, then the changes made will be overwritten the next time this command is directed to the distribution machine.
-file	names each binary file to be replaced by its *.BAK* version.
-dir	provides a complete directory pathname. It is necessary only if the binaries to be replaced reside in a directory other than */usr/afs/bin*.
-cell	specifies the cell to which **-server** belongs. Do not use this argument with the **-localauth** flag. See section 2.2 for more details.
-noauth	tells the BOS Server to assign the unprivileged identity *anonymous* to the issuer. See section 2.2 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. The command interpreter presents the ticket to the BOS Server, which uses it for mutual authentication. Do not use this flag with the **-cell** argument. See section 2.2 for more details.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 2.2 for more details.

EXAMPLES

The following causes the contents of */usr/afs/bin/kaserver.BAK* to overwrite the current contents of */usr/afs/bin/kaserver* on *fs4.transarc.com*. If there is a *.OLD* version of this file, it becomes the *.BAK* version. Since *fs4.transarc.com* is the binary distribution machine for its CPU/operating system type, the change will propagate to all other machines of that type (by default within 5 minutes). The issuer should repeat this command on all binary distribution machines.

% bos u fs4.transarc.com kaserver

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

bos getdate

bos getrestart

bos install

bos restart

bos setrestart

3. The fs Commands

This chapter defines the **fs** commands that users and system administrators employ to contact the File Server and to configure the Cache Manager. It assumes the reader is familiar with the concepts described in the *AFS System Administrator's Guide*.

Some **fs** commands extend UNIX file system semantics by invoking file-related functions that UNIX does not provide (setting access control lists, for example). Other **fs** commands help users control the performance of the Cache Manager running on their local client workstation. When using **fs** commands, pay particular attention to the kind of privilege required, as it varies from command to command.

Refer to the *Command Summary* at the end of this document for a complete list of **fs** commands and their syntax.

3.1. Common Arguments and Flags

All **fs** commands accept the following optional flag. It is listed in the command descriptions and is described in detail here:

[-help]

This flag has the same function as the **fs help** command: It prints a command's online help message on the screen. No other arguments or flags should be provided at the same time as this flag. If they are, this flag overrides them, and the only effect of issuing the command is that the help message appears.

3.2. The Privileges Required for fs Commands

The privileges required for **fs** commands vary more than those required for commands in other suites. Pay special attention to the PRIVILEGE REQUIRED section of each command description.

The various types of necessary privilege include

- Having certain rights on a directory's access control list. For example, creating and removing mount points requires ADMINISTER, INSERT, and DELETE rights for the directory in which the mount point resides. Setting a directory's access control list requires certain rights, too.
- Being logged in as the super-user "root" in the UNIX file system of the machine on which the command is being issued. This is necessary when issuing commands that affect Cache Manager configuration.
- Belonging to the *system:administrators* group in the Protection Database. See the **fs setvol** command for an example.
- No privilege. Many **fs** commands simply list information and so do not require any special privilege.

fs apropos — show each help entry containing keyword.

fs apropos -topic <*help string*> [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs ap -t <*help string*> **[-h]**

DESCRIPTION

Displays the first line of the online help entry for any **fs** command that has *help string* in its name or short description.

ARGUMENTS

-topic specifies the keyword string for which to search. If it is more than a single word, surround it with double quotes or other delimiters. This argument is case-sensitive; type *help string*s for **fs** commands in lowercase letters.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The first line of a command's online help entry names the command and briefly describes what it does. The **fs apropos** command displays that first line for any **fs** command where *help string* is part of the command name or first line.

To see the remaining lines in a help entry, which provide the command's alias (if any) and syntax, use the **fs help** command.

EXAMPLES

The following lists all **fs** commands that have the word "*cache*" in their operation codes or short online descriptions:

% fs apropos -topic cache

```
setcachesize: set cache size flush: flush file from cache
getcacheparms: get cache usage info monitor: set cache monitor host
address
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs help

fs checkservers — check status of file server machines.

fs checkservers [-cell <*cell to check*>**] [-all] [-fast]**
[-interval <*seconds between probes*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs checks [-c <*cell to check*>**] [-a] [-f] [-i** <*seconds between probes*>**] [-h]**

DESCRIPTION

Lists any file server machines in the indicated cell(s) that meet two conditions:

1. The Cache Manager has been in contact with the *fileserver* process running on the machine and/or may need to contact it in future. (The Cache Manager may need to contact any of the database server machines in its cell.)
2. The *fileserver* process on the machine is not currently responding to Cache Manager probes (implying that it is not responding to Cache Manager file requests either).

The Cache Manager constantly maintains a list of file server machines that meet the first condition, updating it every three minutes (by default) by attempting to contact the *fileserver* process on each machine in the list. When a process does not respond to the probe, the Cache Manager marks it as non-functioning. If a machine that previously did not respond begins to respond again, the Cache Manager erases the non-functioning flag. The interval between probes can be adjusted using the **-interval** argument.

This command forces the Cache Manager to update its information immediately (rather than waiting the standard interval). The Cache Manager probes the *fileserver* process on the machines in the specified cell that meet the first condition above, records those that do not respond, and reports the result. If the issuer includes the **-fast** flag, the Cache Manager outputs the list it already has at the time the command is issued instead of probing the machines again.

By default, the Cache Manager probes machines in the local cell only. If the **-all** flag is used, it probes all machines (from all cells) that meet the first condition. If a cell name is specified with **-cell**, The Cache Manager probes the machines in that cell only. The **fs checkservers** command will not use the AFSCELL environment variable.

WARNING

It can take quite a while for this command to produce its entire output if a number of machines in the Cache Manager's list are in fact down when the command is issued. The delay is because after issuing the probe the Cache Manager waits a standard timeout period before concluding that the *fileserver* is not responding; this allows for the possibility of slow cross-network communication. If it is important that the command shell prompt return quickly, the issuer may wish to put this command in the background. It is harmless to interrupt the command (with **Ctrl-C** or another interrupt signal).

This command is not guaranteed to check the status of all file server machines in a cell. The Cache Manager probes only those machines that meet the first condition mentioned above.

ARGUMENTS

-cell specifies the complete name of the cell whose file server machines the Cache manager should probe (shortened forms are not acceptable). Provide this argument or **-all**; it may be combined with **-fast**.

-all causes the Cache Manager to probe all machines that meet the first condition mentioned above. Provide this argument or **-cell**; it may be combined with **-fast**.

-fast tells the Cache Manager to display its current list of down machines, rather than probing any machines. The displayed output may be up to 10 minutes old.

-dir is obsolete, but can still be provided on the command-line. Previous versions of this command required a directory argument. If the issuer includes it by accident, a warning message appears, but the command still executes correctly.

-interval sets the file server probe interval in seconds used by the client machine. By default, the interval is 180 seconds (3 minutes). Only the user "root" can use this argument. Once set, the probe interval will remain set until it is changed via this command or the client machine is re-booted (at which time it will return to the default setting). Do not use any other argument or flag with this argument.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

If the Cache Manager gets a response from all of the machines that it probes (that is, all such machines are functioning normally), the output is

```
All servers are running.
```

(Remember that this message does not imply that all file server machines in the cell are running. It reports the status of only those that the Cache Manager tries to probe.)

If a machine fails to respond to the Cache Manager's probe within the timeout period, the output displays its name. The format of a machine name (name in uppercase, name in lowercase, or Internet address in four-field decimal form) depends on the state of the local cell's name server at the time the command is issued.

When issued with the **-interval** argument, no output is provided.

EXAMPLES

In the following example, the issuer chooses to see the Cache Manager's current list of down machines that belong to the local cell, rather than waiting for it to probe them again. The output indicates that all machines responded to the previous probe.

% **fs checks -f**
```
All servers are running.
```

The following example checks file server machines in all cells that the Cache Manager has previously contacted. It reports that the machines *fs1.transarc.com* and *vice3.andrew.cmu.edu* did not respond to the machine's probe.

% **fs checkservers -all &**
```
These servers are still down:
     fs1.transarc.com
     VICE3.ANDREW.CMU.EDU
```

The following example checks machines that the Cache Manager has previously contacted in the *athena.mit.edu* cell only:

% **fs checks athena.mit.edu &**
```
%All servers are running.
```

PRIVILEGE REQUIRED

None. The user must be logged in as "root" to set the length of the prove interval with the **-interval** option.

fs checkvolumes — force Cache Manager to update volume-related information.

fs checkvolumes [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs checkv [-h]

DESCRIPTION

Forces the Cache Manager to discard its table of mappings between volume names and volume ID numbers. The Cache Manager needs the information in the table to fetch files, so this command will force it to fetch the most current information available at the File Server about a volume's contents before it can fetch any more files.

This command is most useful if the issuer knows that a volume's name has changed, or that there has been a release of new ReadOnly replicas, because issuing it forces the Cache Manager to reference the volume with the new name, or the new ReadOnly replica.

Normally the Cache Manager flushes the table and constructs a new one once per hour anyway.

ARGUMENTS

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

PRIVILEGE REQUIRED

None.

fs cleanacl — remove obsolete entries from access control list.

fs cleanacl [-path <*dir/file path*>$^{+}$**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs cl [-p <*dir/file path*>$^{+}$**] [-h]**

DESCRIPTION

Removes from the access control list of each specified directory or file any entries that specify a user or group no longer found in the Protection Database. When a user/group is removed from the Protection Database, its AFS UID appears on access control lists rather than its name. This command removes such "abandoned" AFS UIDs from access control lists.

Cleaning access control lists in this way not only keeps them from becoming crowded with irrelevant information, but also prevents the new possessor of a recycled AFS UID from obtaining access intended for the former possessor of the ID. (Note that recycling IDs is not recommended in any case.)

ARGUMENTS

-path specifies a file or directory for which the associated access control list is to be cleaned. If a filename is specified, the ACL of the file's parent directory is cleaned. If the issuer omits this argument, the current working directory is assumed.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

If there are no obsolete AFS UIDs on the ACL, the following message appears:

```
Access list for dir/file path is fine.
```

Otherwise, the output reports the resulting state of the ACL, following the header

```
Access list for dir/file path is now
```

EXAMPLES

In the following example, the user *pat* cleans the ACL on the current directory and its subdirectories called *reports* and *sources*. The ACLs for the first two have no obsolete AFS UIDs on them, but *sources* does.

```
% fs cl . ./reports ./sources
Access list for . is fine.
Access list for ./reports is fine.
Access list for ./sources is now
Normal rights:
   system:authuser rl
   pat rlidwka
```

PRIVILEGE REQUIRED

Issuer must have ADMINISTER rights to the directory; by default, the owner of the directory and members of *system:administrators* do.

MORE INFORMATION

fs listacl

fs copyacl — copy access control list from one directory to one or more other directories.

fs copyacl -fromdir <*source directory*> **-todir** <*destination directory*>$^{+}$ **[-clear] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs co -f <*source directory*> **-t** <*destination directory*>$^{+}$ **[-c] [-h]**

DESCRIPTION

Copies the access control list (ACL) from the *source directory* to each *destination directory*. The command does not affect entries on the ACL of the *source directory*. It affects entries on the ACL of each *destination directory* as follows:

- If an entry is unique to the ACL of the *source directory*, it is copied to the ACL of the *destination directory*.
- If an entry exists on the ACLs of both directories, it is changed on the ACL of the *destination directory* to match the rights granted on the ACL of the *source directory*.
- If an entry is unique to the ACL of the *destination directory* and the **-clear** flag is omitted, the entry is not affected.
- If an entry is unique to the ACL of the *destination directory* and the **-clear** flag is included, the entry is removed.

Use the **-clear** flag to completely replace the ACL of each *destination directory* with that of the *source directory*.

ARGUMENTS

-fromdir specifies the *source directory* whose ACL is to be copied to each *destination directory*. Abbreviated pathnames are interpreted relative to the directory in which the command is issued. If a filename is provided, the file's parent directory is used as the *source directory*.

-todir specifies one or more *destination directories* to receive the ACL from the *source directory*. Abbreviated pathnames are interpreted relative to the directory in which the command is issued. A filename cannot be specified with this argument.

-clear removes all existing entries from the ACL of each *destination directory* before copying the ACL from the *source directory*. The ACL of each *destination directory* is thus completely replaced with the ACL of the *source directory*. If the issuer omits this flag, entries that exist on the ACL of a *destination directory* but not on the ACL of the *source directory* are not affected.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following example uses the **fs copyacl** command to copy the ACL from the current directory to the subdirectory named *reports*. Entries on the ACL of the current directory are not affected. Because the **-clear** option is not used, entries on the ACL of the *reports* directory that are not on the ACL of the current directory remain unaffected as well.

```
% fs la . reports
Access list for . is
Normal rights:
   pat rlidwka
   smith rlidwk

Access list for reports is
Normal rights:
   pat rl
   pat:friends rl
Negative rights
   jones rlidwka
```

```
% fs co . reports
```

```
% fs la . reports
Access list for . is
Normal rights:
   pat rlidwka
   smith rlidwk

Access list for reports is
Normal rights:
   pat rlidwka
   pat:friends rl
   smith rlidwk
Negative rights
   jones rlidwka
```

PRIVILEGE REQUIRED

Issuer must have LOOKUP right to the *source directory* and ADMINISTER right to each *destination directory*. To issue the command with a filename used for *source directory*, the issuer must have both the LOOKUP and READ rights on the ACL of the file's parent directory.

MORE INFORMATION

fs listacl

fs setacl

fs debug — enable/disable Cache Manager debugging trace.

fs debug -debug <*'on' or 'off'*> **[-dafs** <*afs debug level*>**]**
[-dnet <*network debug level*>**] [-syslog] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs de -de <*'on' or 'off'*> **[-da** <*afs debug level*>**] [-dn** <*network debug level*>**] [-s] [-h]**

DESCRIPTION

Determines whether the Cache Manager records information about its activities that may prove helpful in debugging or other trouble-shooting. The output goes into the file */usr/vice/etc/AFSLog* (unless an alternate directory or name is specified for the file with the **-logfile** argument of the **afsd** command). See the ARGUMENTS section for information about the different types of debugging output that can be written to the file.

You can use the **more** command (or an equivalent command such as the **pg** command on AIX systems) to read the debugging output recorded in the *AFSLog* file. You must be logged in as **root** on the machine on which the *AFSLog* file resides to read the file. Interpreting the output requires familiarity with the AFS source code.

ARGUMENTS

-debug controls whether debugging information is produced. The legal values are *on*, which directs debugging information into the *AFSLog* file, and *off*, which stops the recording of information in the file.

-dafs determines the types of debugging information the Cache Manager produces about its activities. The following list describes the legal values for this argument and the type of debugging output each causes the Cache Manager to write to the *AFSLog* file:

- *1*, which causes the Cache Manager to write standard debugging information. Using this value provides a good deal of general output.
- *2*, which causes the Cache Manager to write low-level debugging information about the AFS network. Use this value only if you are convinced that network problems exist.
- *4*, which causes the Cache Manager to write debugging information about the RX protocol.
- *8*, which causes the Cache Manager to write debugging information about the interface layer to AFS. This value is not useful on machines running a Sun operating system.

In addition, if a value of *1*, *4*, or *8* is specified, the Cache Manager also records in the *AFSLog* file the AFS UID of each user who accesses data from a file server machine. It records the appropriate AFS UID with each operation that accesses data.

The legal values can be added to specify different combinations of output. For example, a value of *15* specifies that all possible types of output are to be provided. The default value of *1* is used if no value is specified.

Note: The *AFSLog* file also records the type of volume (ReadWrite, ReadOnly, or Backup) accessed from a file server machine. The type of the volume is displayed along with the volume ID in the "state" flag in bitmap form. If a ReadWrite volume is accessed, the bits are clear; if a ReadOnly volume is accessed, the 1 bit is set; if a Backup volume is accessed, the 4 bit is set.

-dnet is not currently implemented and should not be used.

-syslog specifies that debugging output is to be redirected to the **syslogd** daemon. This flag can be used only on machines running Sun OS 4.1 or higher.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following turns on debugging using the default debugging level of *1*:

% fs de on

PRIVILEGE REQUIRED

None.

MORE INFORMATION

afsd

fs diskfree — show information about the partition housing a directory/file.

fs diskfree [-path <*dir/file path*>+] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs df [-p <*dir/file path*>+**] [-h]**

DESCRIPTION

Provides information about the partition that houses the volume containing the specified directory or file. See the OUTPUT section for a complete explanation of the information provided. To learn more about the volume itself, use the **fs examine** command.

ARGUMENTS

-path specifies a file or directory about whose host partition information is desired. If the issuer omits this argument, the current working directory is assumed.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

Note: The numbers that appear in this output may not always agree with the corresponding numbers in the output of the standard UNIX **df** command. The main reason is that the **df** output reflects the state of partitions exactly when the command is issued. The numbers in this command's output may be up to 5 minutes old, as the Cache Manager polls the File Server for partition information at that frequency. Another potential difference: the partition size reported by the UNIX **df** command includes some reserved space that does not show up in this report of partition size, and so is likely to be about 10% larger.

The output reports the following information about each partition that houses a specified directory or file:

- The name of the volume that contains the directory or file.
- The total size in kilobyte blocks of the partition that stores the named volume.
- The number of kilobyte blocks used on the partition.
- The number of kilobyte blocks available on the partition.
- The percentage of the partition's total space used.

EXAMPLES

The following shows the output for the partition housing the volume `user.smith` in the Transarc Corporation cell:

```
% fs df /afs/transarc.com/usr/smith
Volume Name    kbytes  used     avail     %used
user.smith     333305  286710   46595       86%
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs examine

fs examine — show information about volume containing specified directory.

fs examine [-path <*dir/file path*>+] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs exa [-p <*dir/file path*>+] [-h]

fs listvol [-p <*dir/file path*>+] [-h]

fs lv [-p <*dir/file path*>+] [-h]

DESCRIPTION

Displays information about the volume containing each specified directory or file. The information includes the file's quota and current size. See the OUTPUT section for a complete explanation of the information provided. While this command provides the most information about a volume, the **fs listquota** and **fs quota** commands are also available to display information about a volume.

ARGUMENTS

-path specifies each file and/or directory for which information about the host volume is desired. Omit this argument to display information about the volume that contains the current working directory.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

Note: The partition-related numbers that appear in this output may not always agree with the corresponding numbers in the output of the standard UNIX **df** command. The main reason is that the **df** output reflects the state of partitions exactly when the command is issued. The numbers in this command's output may be up to 5 minutes old, as the Cache Manager polls the File Server for partition information at that frequency. Another potential difference: the partition size reported by the UNIX **df** command includes some reserved space that does not show up in this report of partition size, and so is likely to be about 10% larger.

The output reports the following information about each volume that contains a specified directory or file:

- The volume ID number (abbreviated in the output as "`vid`") of the volume.
- The volume's name.
- The current "offline" message associated with the volume, as set by a system administrator using the **fs setvol** command.
- The current "message of the day" associated with the volume, as set by a system administrator using the **fs setvol** command.
- The volume's maximum size quota, in kilobyte blocks.
- Its current size, in kilobyte blocks.
- The number of kilobyte blocks still available on the disk partition that houses the volume and the partition's total size.

EXAMPLES

The following shows the output for the volume `user.smith` (and the partition housing it) in the Transarc Corporation cell:

```
% fs exa /afs/transarc.com/usr/smith
Volume status for vid = 50489902 named user.smith
Current maximum quota is 15000
Current blocks used are 5073
The partition has 46383 blocks available out of 333305
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs listquota

fs quota

fs setquota

fs exportafs — report or set whether machine can export AFS to clients of alternate file systems.

fs exportafs -type <*exporter name*> **[-state** <*'on' or 'off'*>**]**
[-noconvert] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs exp -t <*exporter name*> **[-s** <*'on' or 'off'*>**] [-n] [-h]**

DESCRIPTION

This command performs one of the following, depending on whether the issuer provides the **-state** argument:

- It sets whether the machine is accessible as a server of the non-AFS file system *exporter name*, able to be mounted by clients of that file system.
- It reports on the current status of the machine.

The command's **-noconvert** flag can be used to indicate whether mode bits of exported directories and files are to be converted. By default, the group and other mode bits of exported directories and files are changed to match the user bits.

ARGUMENTS

-type names the alternate file system for which the setting is to be changed or reported. Only lowercase letters are acceptable. The only legal value is *nfs*.

-state controls whether the workstation is accessible as a server of the non-AFS file system or not. The legal values are *on*, which enables the workstation as a server, and *off*, which makes it inaccessible as a server. If the issuer omits this argument, the output reports the current setting.

-noconvert
determines whether the group and other bits on exported files and directories are converted to match the user bits. By default, the group and other bits on exported files and directories are made to match the user bits. Specify this flag to leave the bits as they are in AFS.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

When the **-state** argument is omitted, the output reports the name of the non-AFS file system and whether the workstation is enabled as a server of it.

EXAMPLES

The following shows that this machine is enabled as an NFS server (that is, it is running the AFS/NFS Translator):

```
% fs exportafs nfs
Exporter type: nfs is currently enabled for AFS
```

The following shows that the machine is not enabled as an NFS server:

```
% fs exportafs nfs
Sorry, the nfs-exporter type is currently not supported on
    this AFS client
```

The following prevents the machine from acting as an NFS server:

```
% fs exp nfs off
```

PRIVILEGE REQUIRED

Issuer must be logged in as "root" on the machine on which the command is being issued.

fs flush — force Cache Manager to discard a cached file/directory.

fs flush [-path <*dir/file path*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs flush [-p <*dir/file path*>+**] [-h]**

DESCRIPTION

Forces the Cache Manager to remove each specified directory or file from its cache of data and status information. The result is that the next time data from a flushed directory or file is requested, the Cache Manager contacts the File Server for the most current version, along with a new callback (if necessary) and associated status information. This command does not discard data from application program buffers or data that has been altered in the cache but not yet written back to the central copy maintained by the File Server.

The **fs flushvolume** command can be used to flush all data that resides in the same volume as a specified file or directory.

ARGUMENTS

-path specifies each file or directory to be flushed. In the case of a directory element, only the element itself is flushed, not data cached from files or subdirectories that reside in it. If this argument is omitted, the current directory is flushed.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following flushes from the cache the file *projectnotes* in the current working directory and all data from the subdirectory *plans*:

```
% fs flush projectnotes ./plans/*
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs flushvolume

fs flushvolume — force Cache Manager to discard any cached data from the volume containing specified file/directory.

fs flushvolume [-path <*dir/file path*>+] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs flushv [-p <*dir/file path*>+] [-h]

DESCRIPTION

Forces the Cache Manager to remove cached data (but not the cached status information) for all files and directories that reside in the same volume as each specified directory or file. The result is that the next time the Cache Manager needs anything from a flushed volume, it contacts the File Server for the most current version, along with a new callback (if necessary). This command does not discard data from application program buffers or data that has been altered in the cache but not yet written back to the central copy maintained by the File Server.

The **fs flush** command can be used to flush individual files and directories.

ARGUMENTS

-path specifies one file or directory from each volume that the Cache Manager is to flush completely from its cache. If this argument is omitted, all data from the volume that contains the current directory is flushed.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following flushes from the cache all data that comes from the volume that contains the current working directory and the directory *reports* at the same level in the file tree:

```
% fs flushv . ../reports
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs flush

fs getcacheparms — show current size of data cache and amount being used.

fs getcacheparms [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs getca [-h]

DESCRIPTION

Displays the current size of the cache that the Cache Manager has at its disposal, and the amount it is using at the moment the command is issued. The command works both on machines using a memory cache and on machines using a disk cache.

This information comes from the kernel of the workstation on which the command is issued. On machines using a disk cache, the current cache size may disagree with the default setting specified in the file */usr/vice/etc/cacheinfo*, if someone has set it with the **fs setcachesize** command.

ARGUMENTS

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output is of the form

```
AFS using <amount> of the cache's available <size> 1K byte blocks.
```

where <*amount*> is the number of 1K byte blocks the Cache Manager is currently using, and <*size*> the total number of blocks available to the Cache Manager (the current cache size).

EXAMPLES

The following shows the output on a machine with a 25000 kilobyte cache.

```
% fs getca
AFS using 22876 of the cache's available 25000 1K byte blocks.
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs setcachesize

fs getcellstatus — show whether workstation can run setuid programs from specified cell(s), and whether cell is using the old VLDB.

fs getcellstatus -cell <*cell name*>+ **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs getce -c <*cell name*>+ **[-h]**

DESCRIPTION

Reports whether the workstation allows programs fetched from the specified cell(s) to run with setuid privilege. System administrators set a cell's setuid status on a per-workstation basis with the **fs setcell** command.

If a cell is using the AFS 2.0 method for tracking volume location rather than the VLDB, the output reports this also (see the OUTPUT section).

ARGUMENTS

-cell names the cell(s) for which setuid status is desired. Provide the complete Internet-style name for each cell (unlike the common **-cell** argument in other command suites, it is not possible to abbreviate this one).

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

Possible output values are

- `no setuid allowed`, indicating that programs from the cell may not run with setuid privilege.
- `setuid allowed`, indicating that programs from the cell may run with setuid privilege.
- `using old VLDB`, indicating that the cell is still using the AFS 2.0 volume location method.

EXAMPLES

The following indicates that programs from the cell *oldcell.com* may not run with setuid privilege and that the cell is still using the old volume location method:

```
% fs getce oldcell.com
Cell oldcell.com status: no setuid allowed, using old VLDB
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs setcell

fs getserverprefs — display Cache Manager's preferences for file server machines.

fs getserverprefs [-file <*dir/file path*>] [-numeric] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs gets [-f <*dir/file path*>] [-n] [-h]

fs gp [-f <*dir/file path*>] [-n] [-h]

DESCRIPTION

Displays the Cache Manager's preferences for file server machines. A preference consists of the name or IP address of a file server machine followed by its "rank." The rank is a positive integer in the range from 1 to 65,534.

A file server machine's rank determines the Cache Manager's preference for selecting it when the Cache Manager must access a ReadOnly replica that resides on it. The Cache Manager compares the rank of the server machine with the ranks of other server machines that house the replica. It then attempts to access the replica on the server machine that has the lowest integer rank.

If it cannot access the replica on the machine with the lowest rank (possibly because the machine or the network on which the machine is located is down), the Cache Manager attempts to access the replica from the server machine with the next lowest rank. It continues in this manner until it either accesses the replica or determines that all of the file server machines on which the replica resides are unavailable.

The Cache Manager records addresses and ranks for all local file server machines. It also records addresses and ranks for all foreign file server machines that house a volume it has accessed or for which a rank has been specified with the **fs setserverprefs** command. It stores the addresses and ranks in the kernel of the client machine.

Information displayed with this command is sent to stdout by default. The **-file** argument can be used to direct the output to a file.

ARGUMENTS

-file specifies the pathname of a file to which the file server machine names and ranks are to be written. Omit this argument to display the machine names and ranks on stdout.

-numeric specifies that the IP addresses of the file server machines are to be displayed. Omit this flag to display the names of the file server machines. Because including this flag skips the resolution of IP addresses to machine names, information is displayed more quickly than if the option is omitted. (This flag is especially useful if the output is intended to be used as input to the **fs setserverprefs** command, in which case it does not matter whether names or addresses are used.)

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output displays a separate line for each file server machine that has a rank in the kernel of the machine on which the command is issued. Each line displays the name of a file server machine followed by its rank, as follows:

first machine name *rank*
second machine name *rank*
. . . .

If the **-numeric** flag is included with the command, the output displays the IP addresses of the file server machines instead of their names. The address of a machine is also displayed if the Cache Manager cannot resolve a file server machine's name based on the machine's address at the time the command is issued.

EXAMPLES

The following displays the preferences (the list of file server machines and their respective ranks) associated with a Cache Manager. The output in the example truncates the complete list of server machine names and ranks. Note that the IP addresses, not the names, of some machines are displayed because their addresses cannot be resolved.

```
% fs gets
fs5.transarc.com                20000
fs1.transarc.com                40000
fs3.transarc.com                20001
fs4.transarc.com                40001
fs2.transarc.com                25000
121.86.3.37                     40002
fserver1.andrew.cmu.edu         40000
121.86.3.34                     40001
server1.athena.mit.edu          1000
   .   .   .   .   .             . .
```

The following displays the same Cache Manager's preferences, but the **-numeric** flag is included to display only the IP addresses of the file server machines, not their names. The example output again truncates the complete list of server machine names and ranks.

```
% fs gets -n
128.21.6.214                    20000
128.2.11.9                      40000
128.2.11.12                     20001
128.2.11.13                     40001
128.2.11.11                     25000
121.86.3.37                     40002
121.86.3.31                     40000
121.86.3.34                     40001
145.2.50.121                    1000
  .  .  .                        . .
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs setserverprefs

fs help — show syntax of specified **fs** commands or list functional descriptions of all **fs** commands.

fs help [-topic <*help string*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs h [-t <*help string*>+**] [-h]**

DESCRIPTION

Displays the first line (name and short description) of every **fs** command's online help entry if no *help string* is provided. For each operation code specified with **-topic**, it outputs the entire help entry. See the OUTPUT section.

ARGUMENTS

-topic	specifies the operation codes for which syntax is to be provided. If the issuer omits this argument, the output instead provides a short description of all **fs** commands.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The online help entry for each **fs** command consists of two or three lines:

- The first line names the command and briefly describes what it does.
- The second line displays any aliases the command has (this line does not appear for every command).
- The final line, which begins with "`Usage:`", lists the command's arguments and flags in the prescribed order. Online help entries use the same symbols (for example, brackets) as the command definitions in this manual. For an explanation of their meaning, see page x of the introductory *About This Manual* chapter.

EXAMPLES

The following displays the online help entry for the **fs setacl** command:

```
% fs help setacl
fs setacl: set access control list
aliases: sa
Usage: fs setacl -dir <directory>+
-acl <access list entries>+ [-clear] [-negative] [-help]
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs apropos

fs listacl — show access control list.

fs listacl [-path <*dir/file path*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs la [-p <*dir/file path*>+**] [-h]**

DESCRIPTION

Displays the access control list (ACL) associated with each *directory*. It is legal to provide a filename rather than a directory name for *directory*, in which case the ACL of the file's parent directory is displayed (because it is not possible to set an ACL for an individual file, the file is inheriting the ACL from its parent directory). Omit this argument to display the ACL of the current working directory.

Users who possess the ADMINISTER right on an ACL may change the ACL with the **fs setacl** command or copy the ACL from a different directory to it with the **fs copyacl** command.

WARNING

The appearance of a user/group on the `Negative rights` list does not guarantee that the person is denied those rights. If *system:anyuser* is granted any rights on the `Normal rights` list, a user need only unlog to obtain those rights.

ARGUMENTS

-path	specifies each file and/or directory for which to display the associated ACL. If this argument is omitted, the output displays the ACL associated with the current working directory. If it is a filename, the ACL displayed is associated with the file's parent directory.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The first line of the output names the directory associated with the access control list. If the issuer used shorthand notation (such as "." for the current directory) when indicating the *directory*, it may appear here rather than the full pathname of the directory.

The "`Normal rights:`" header indicates the list of users who have normal rights to the directory. Each following line lists a user/group name and the set of rights the user/group may exercise. The possible rights and their meanings are

- `r` = READ the contents of files in the directory
- `w` = WRITE (modify) the contents of files in the directory
- `l` = LOOKUP status information about the files in the directory
- `d` = DELETE files from the directory
- `i` = INSERT new files into the directory
- `k` = LOCK; set read or write locks on the files in the directory
- `a` = ADMINISTER; change the rights on the access control list
- `A`, `B`, `C`, `D`, `E`, `F`, `G`, `H`;
  by default, these have no meaning to AFS server processes. Administrators and application programs may assign meanings to them and place them on ACLs to control access to the directory's contents in new ways. The letters must be uppercase.

A "`Negative rights:`" header may appear next, if any negative rights have been specified for this directory. The format of this list is the same as that of the `Normal rights` list. The difference is that the user(s)/group(s) listed are denied rather than granted the specified rights.

EXAMPLES

The following displays the ACL associated with user *pat*'s home directory and its *private* subdirectory when the **fs listacl** command is issued in the home directory:

```
% fs la . private
Access list for . is
Normal rights:
   system:authuser rl
   pat rlidwka
   pat:friends rlid
Negative rights:
   smith rlidwka

Access list for private is
Normal rights:
   pat rlidwka
```

PRIVILEGE REQUIRED

To issue this command with a directory name argument, issuer must have the LOOKUP right on the directory's ACL. To issue this command with a filename argument, the issuer must have both the LOOKUP and READ rights on the ACL of the file's parent directory.

MORE INFORMATION

fs cleanacl

fs copyacl

fs setacl

fs listcells — show database server machines in cell(s) known to Cache Manager.

fs listcells [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs listc [-h]

DESCRIPTION

Formats and displays the Cache Manager's kernel-resident list of the database server machines in its home cell and foreign cells.

At each reboot of the workstation, the Cache Manager copies the contents of */usr/vice/etc/CellServDB* into the kernel. It is possible to modify the kernel-resident list between reboots using **fs newcell**.

ARGUMENTS

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output contains a line for each cell for which the kernel has a list of database server machines. The cell name is followed by a list of its database server machines (referred to as "`hosts`").

The format of each machine name (name in uppercase, name in lowercase, or Internet address in four-field decimal form) depends on the state of the local cell's name server at the time the command is issued.

EXAMPLES

The following shows output for several cells as illustrations of the different formats for machine names:

```
% fs listc
Cell transarc.com on hosts fs1.transarc.com fs2.transarc.com
Cell andrew.cmu.edu on hosts VICE11.FS.ANDREW.CMU.EDU
   VICE2.FS.ANDREW.CMU.EDU VICE7.FS.ANDREW.CMU.EDU.
Cell athena.mit.edu on hosts 18.80.0.2 orf.mit.edu
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs newcell

fs listquota — show quota information for the volume containing a file/directory.

fs listquota [-path <*dir/file path*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs lq [-p <*dir/file path*>+**] [-h]**

DESCRIPTION

Displays information about the size and quota of the volume containing each specified directory or file. See the OUTPUT section for a complete explanation of the information provided.

ARGUMENTS

-path specifies each file and/or directory for which information about the host volume is desired. If the issuer omits this argument, the current directory is assumed.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output reports the following information about each volume that contains a specified directory or file:

- The name of the volume.
- Its maximum size quota, in kilobytes.
- Its current size, in kilobytes.
- The percentage of its quota that its current size represents.
- The percentage of the volume's disk partition that is full. This is usually unrelated to how much of the user's quota is used, since it depends on all the volumes on the partition. A large value may nevertheless prevent a user from being able to store more data on the partition.

EXAMPLES

The following shows the output for the volume `user.smith` in the Transarc Corporation cell:

```
% fs lq /afs/transarc.com/usr/smith
Volume Name      Quota    Used    % Used   Partition
user.smith       15000    5071       34%         86%
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs diskfree

fs examine

fs quota

fs setquota

fs setvol

fs lsmount — show volume for which directory is a mount point.

fs lsmount -dir <*directory*>+ **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs ls -d <*directory*>+ **[-h]**

DESCRIPTION

For each *directory*, displays the name of the volume for which it is a mount point. If *directory* is not a mount point or is not in AFS, an error message appears. The association between *directory* and a volume name is created with the **fs mkmount** command.

ARGUMENTS

-dir names the directory that serves as a mount point for a volume. The last element in the pathname that the issuer provides must be an actual name, not "dot" (.) or "dot dot" (. .), which the **fs** command interpreter does not understand in this case.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output is of the form:

```
'directory' is a mount point for volume 'volume name'
```

A hash sign (#) preceding *volume name* indicates that *directory* is a regular mount point.

A percent sign (%) preceding *volume name* indicates that *directory* is a ReadWrite mount point.

If *directory* is a cellular mount point, then a cell name and colon precede *volume name* in addition to the hash sign or percent sign.

If *directory* is not a mount point, the output reads:

```
'directory' is not a mount point.
```

EXAMPLES

The following shows the mount point for the home directory of user *smith* in the Transarc Corporation cell:

```
% fs ls /afs/transarc.com/usr/smith
'/afs/transarc.com/usr/smith' is a mount point for
    volume '#user.smith'
```

The following shows both the regular and ReadWrite mount points for the Transarc Corporation cell's `root.cell` volume.

```
% fs ls /afs/transarc.com
'/afs/transarc.com' is a mount point for volume '#root.cell'

% fs ls /afs/.transarc.com
'/afs/.transarc.com' is a mount point for volume
    '%root.cell'
```

The following shows a cellular mount point: the Andrew cell's `root.cell` volume as mounted in the Transarc Corporation cell's tree.

```
% fs ls /afs/andrew.cmu.edu
'/afs/andrew.cmu.edu' is a mount point for volume
    '#andrew.cmu.edu:root.cell'
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs mkmount

fs rmmount

fs messages — log and disable Cache Manager messages.

fs messages [-show <*user*|*console*|*all*|*none*>**] [-log** <*user*|*console*|*all*|*none*>**] [-writethru** <*on*|*off*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs me [-s <*user*|*console*|*all*|*none*>**] [-l** <*user*|*console*|*all*|*none*>**] [-w** <*on*|*off*>**] [-h]**

DESCRIPTION

Logs and/or disables messages generated by the Cache Manager. The messages can be logged to the file *AFSLog* and/or partially or entirely disabled (though entirely disabling Cache Manager messages is not recommended because they provide useful status and warning information.)

There are two types of Cache Manager messages:

- User messages, which provide user status and warning information.
- Console messages, which provide system-level status and warning information.

The **-show** and **-log** arguments are used to specify the destination of these messages. The **-show** argument is used to direct messages to the user screens and/or client machine console and the **-log** argument is used to direct messages to the *AFSLog* file. These arguments can be combined to send the same messages to both locations, to send some messages to the screens or console and others to the log file, or to totally disable messaging. (For example, user messages can be sent to user screens and logged to the *AFSLog* file). By default, user messages are sent to user screens, console messages are sent to the designated console, and no messages are logged in *AFSLog*.

Note: This command cannot be used to redirect user messages to the console or console messages to user screens.

By default, the Cache Manager writes all log-bound messages into a two-kilobyte buffer and when that buffer is filled its contents are written to the *AFSLog* file. For this reason, logged messages may not be immediately visible in the log file. If you want log-bound messages to be written directly to *AFSLog* so that the messages are immediately available, use the **-writethru** flag. Note, however, that using this flag imposes a performance penalty on the affected Cache Manager.

ARGUMENTS

-show specifies which types of messages are to be sent to the screens/console. The valid values are listed below:

- *user*: Send user messages to user screens.
- *console*: Send console messages to the console.

- *all*: Send user messages to user screens and console messages to the console. This is the default.
- *none*: Do not send any messages to the screens/console.

-log specifies which types of messages are to be logged to the file *AFSLog* stored in the directory specified by **afsd** (by default */usr/vice/etc*). The valid values are listed below:

- *user*: Log user messages to *AFSLog*.
- *console*: Log console messages to *AFSLog*.
- *all*: Log user messages and console messages to *AFSLog*.
- *none*: Do not log any messages to *AFSLog*. This is the default.

-writethru
indicates that messages being logged should be written directly to *AFSLog* rather than being stored in the log buffer. By default, logged messages are stored in a two kilobyte buffer; when the buffer is filled, the contents are written to *AFSLog*.

EXAMPLES

The following command instructs the local Cache Manager to display and log all Cache Manager messages:

% fs messages -sh all -log all

PRIVILEGE REQUIRED

Issuer must be logged in as "root" on the machine on which the command is issued.

MORE INFORMATION

fs listcells

fs mkmount — create a mount point for a volume.

fs mkmount -dir <*directory*> **-vol** <*volume name*> **[-cell** <*cell name*>**]**
[-rw] [-fast] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs mk -d <*directory*> **-v** <*volume name*> **[-c** <*cell name*>**] [-r] [-f] [-h]**

DESCRIPTION

Creates a mount point named *directory* for the volume *volume name*. The volume's root directory is also named *directory*. Mount points look and act just like standard UNIX directories; when the Cache Manager encounters a mount point named *directory* in a pathname, it knows to look in the volume indicated by the mount point for the elements contained in *directory*. It is possible, although not recommended, to create more than one mount point to a volume.

Volume Access

The Cache Manager follows three basic rules as it traverses AFS mount points:

Rule 1: Access Backup and ReadOnly Volumes When Specified

When the Cache Manager encounters a mount point that specifies a ReadOnly or Backup volume (a volume name that includes a `.readonly` or `.backup` extension), it accesses only the specified type of volume. For example, the Cache Manager never accesses the ReadWrite version of a volume from a mount point that names the Backup version of the volume. If the specified version is inaccessible, the access attempt fails. If a mount point does not specifically name a ReadOnly or Backup volume, the Cache Manager uses Rules 2 and 3.

Rule 2: Follow the ReadOnly Path When Possible

When the Cache Manager encounters a mount point located in a ReadOnly volume, it attempts to access a ReadOnly copy of the volume to which the mount point refers; if the indicated volume is not replicated, the Cache Manager accesses the ReadWrite version of the volume. When it encounters a mount point in a ReadOnly volume, the Cache Manager attempts to stay on a "ReadOnly path," accessing the ReadOnly version of the indicated volume; it accesses the ReadWrite version of the volume only if the volume is not replicated. Accessing ReadOnly versions of volumes as much as possible increases system performance and improves file availability.

The Cache Manager's built-in bias to access ReadOnly volumes also applies to the `root.afs` volume, which corresponds to the top-level */afs* directory. If a ReadOnly copy of `root.afs` is available, the Cache Manager accesses the ReadOnly version instead of the ReadWrite version. Note that if the Cache Manager cannot access a ReadOnly version of a replicated volume, the access attempt fails; the Cache Manager does not attempt to access the ReadWrite version of a replicated volume for which no ReadOnly versions are currently available.

Note: You must replicate both `root.afs` and `root.cell` (corresponding to */afs* and */afs/***cellname**, respectively) to ensure the Cache Manager's access to other replicated volumes in your cell.

Rule 3: Once on a ReadWrite Path, Stay There

When the Cache Manager encounters a mount point in a ReadWrite volume, it attempts to access only the ReadWrite volume named by the mount point; if the Cache Manager cannot access the ReadWrite volume, the access attempt fails. While the Cache Manager traverses ReadWrite volumes, it is said to be following the "ReadWrite path." Once on a ReadWrite path, the Cache Manager cannot return to a ReadOnly path. (One exception to this rule concerns cellular mount points, a description of which follows.)

Types of Mount Points

There are three types of mount points, each of which is appropriate for different purposes. Each type of mount point interprets the rules of mount point traversal in a slightly different manner.

- ***Regular mount points*** — When the Cache Manager encounters a regular mount point, it follows all rules of mount point traversal.
- ***ReadWrite mount points*** — When the Cache Manager encounters a ReadWrite mount point, it always accesses the ReadWrite version of the indicated volume; thereafter, it remains on the ReadWrite path (unless a mount point specifies the ReadOnly or Backup version of a volume, in which case the Cache Manager accesses that version of the volume).
- ***Cellular mount points*** — When the Cache Manager encounters a cellular mount point, it knows that it must access a volume in a different cell. In this case, it automatically follows a ReadOnly path, regardless of the path it was on prior to encountering the cellular mount point.

The **fs lsmount** command is used to display mount points. The command uses various symbols to distinguish among the different types of mount points; see the description of the command for more information.

The Purposes of the Three Types of Mount Points

The three types of mount points have the following purposes:

Regular mount points

The vast majority of your mount points should be regular mount points that name the ReadWrite versions of volumes. Using regular mount points as much as possible promotes the most efficient use of both replicated and nonreplicated volumes and, assuming your volumes are distributed in a balanced manner, the most efficient use of your system resources.

Because there are normally multiple ReadOnly copies of a volume but only one ReadWrite version, accessing one of the ReadOnly copies rather than the single ReadWrite version distributes volume access among the multiple ReadOnly copies. Because a regular mount point is not a "ReadOnly" mount point, the Cache Manager can still access a ReadWrite volume from a regular mount point when it is on a ReadWrite path or when it must access a volume that is not replicated.

Note: You must replicate all volumes that lead to a replicated volume; the Cache Manager cannot access a replicated volume that is accessible via only a ReadWrite path.

ReadWrite mount points

By convention, the only ReadWrite mount point in your filespace should be the one used to mount your cell's `root.cell` volume just below */afs*. If you use a ReadWrite mount point for this purpose, you should also mount `root.cell` with a regular mount point at the same level. You typically distinguish the two mount points by placing a period at the start of the ReadWrite mount point's name.

The existence of a ReadWrite mount point for `root.cell` is sufficient to put the Cache Manager on a ReadWrite path all the way down the pathnames in your cell when a user needs to access a ReadWrite volume specifically. The existence of a regular mount point for `root.cell` allows the Cache Manager to follow a ReadOnly path whenever possible.

Mounting a ReadOnly or Backup volume with a ReadWrite mount point is possible, but it is unnecessary. If a mount point explicitly names a ReadOnly or Backup version, the Cache Manager accesses the named version, regardless of whether the mount point is regular or ReadWrite.

Cellular mount points

A cellular mount point tells the Cache Manager that the indicated volume resides in a specified foreign cell. If a mount point is not cellular, the Cache Manager assumes that the indicated volume resides in the same cell as the mount point. Normally, you use cellular mount points only at the second level in your cell's file tree, just below */afs*, to mount the `root.cell` volumes of foreign cells that you wish to make visible to your cell's users. It is possible to create cellular mount points (mount foreign volumes) at other levels in your file tree, but doing so is not recommended; users can become confused if the Cache Manager switches from cell to cell at different points in a pathname.

Cellular mount points can be either regular or ReadWrite:

- A regular cellular mount point tells the Cache Manager to cross into a foreign cell and access the ReadOnly version of the indicated volume if possible. The Cache Manager traverses a "ReadOnly path" in the foreign cell even if the cellular mount point resides in a ReadWrite volume in the local cell.
- A ReadWrite cellular mount point tells the Cache Manager to cross into a foreign cell and access the ReadWrite version of the indicated volume (assuming the mount point does not explicitly name the ReadOnly or Backup version of the volume). Using this type of mount point is discouraged because accessing a ReadWrite volume requires the File Server to issue callbacks, an extra load to impose on the servers of a foreign cell. In general, only a cell's own administrators need to access the ReadWrite versions of replicated volumes that reside in that cell.

ARGUMENTS

-dir names the directory to be created as a mount point to the named volume. The specified directory must not already exist. If you do not specify a full pathname, the mount point is created as a subdirectory of the current working directory.

-vol names the volume to be mounted. Add the `.readonly` or `.backup` extension, if appropriate. You can also specify a volume ID number.

-cell names the cell in which the volume resides. By default, the volume is assumed to reside in the same cell as the volume in which the mount point is to be created.

-rw designates the mount point as ReadWrite, which forces the Cache Manager to access only the ReadWrite copy of the indicated volume (unless the mount point names the ReadOnly or Backup version of the volume). Omit this flag to create a regular mount point.

-fast indicates that the VL Server is not to check that a VLDB entry exists for the volume to be mounted; by default, the VL Server checks and prints a warning message if the volume has no VLDB entry. The mount point is created regardless of whether this flag is specified.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

Note: These examples illustrate only the recommended combinations and use of arguments. The description of the **fs lsmount** command shows what each mount point looks like.

The following creates a regular mount point, mounting `user.smith` at */afs/transarc.com/usr/smith*:

% cd /afs/transarc.com/usr
% fs mk smith user.smith

The following creates both a ReadWrite mount point and a regular mount point for the Transarc Corporation cell's `root.cell` volume in that cell's file tree. It follows the convention of putting a period at the beginning of the ReadWrite mount point's name.

% fs mk /afs/transarc.com root.cell
% fs mk /afs/.transarc.com root.cell -rw

The following mounts the `root.cell` volume belonging to Carnegie Mellon University's Andrew cell in the Transarc Corporation cell's file tree, creating a regular cellular mount point called *andrew.cmu.edu*. When a Transarc Corporation Cache Manager encounters this mount point, it crosses into the Andrew cell on a ReadOnly path.

% fs mk /afs/andrew.cmu.edu root.cell -c andrew.cmu.edu

PRIVILEGE REQUIRED

Issuer must have INSERT and ADMINISTER access for the directory that is to contain the mount point.

MORE INFORMATION

fs lsmount

fs rmmount

fs monitor — direct reports on file system activity to a specified machine or report current monitoring machine.

fs monitor [-server <*host name*> or <*off*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs mo [-s <*host name*> or <*off*>**] [-h]**

DESCRIPTION

Performs one of the following actions, depending on the value provided for the **-server** argument:

- If the **-server** argument names a machine, the command makes that machine the destination of the Cache Manager's messages about file system activity.
- If the **-server** argument specifies the value *off*, the command disables message sending.
- If the **-server** argument is omitted, the command reports the current destination for messages, if monitoring is enabled.

The messages are of a less technical nature than those generated by the **fs debug** command. They are at the level of file fetches and stores. For the messages to be displayed, the specified destination machine must be running a monitoring program that "listens" to the correct UDP socket. If the destination machine is not running such a program, the messages are lost.

WARNING

The effect of this command continues even after the issuer logs out. For example, suppose a user working on the machine named *machineA* issues the command, providing the machine named *machineB* as the value for the **-server** argument, and then logs out. Subsequent users of *machineA* see no messages about file system activity, but users of *machineB* see messages from *machineA*. To stop messages from being sent to *machineB*, a user on *machineA* should issue the **fs monitor** command with the value *machineA* or *off* for the **-server** argument.

Transarc Corporation does not provide a monitoring program appropriate for use with this command, but such a program, called "Console," is available as part of the Andrew Toolkit developed at Carnegie Mellon University's Information Technology Center. If no monitoring program is available, it is best to provide a value of *off* for the **-server** argument.

ARGUMENTS

-server has two legal values:

- *off* directs the Cache Manager not to generate any reports on its role in file system activities. This setting is recommended if no machine is running a monitoring program capable of intercepting and displaying the messages that are produced.

- *host name* names the machine to which the Cache Manager is to send messages. The *host name* must be a complete Internet-style machine name. The specified machine should be running a monitoring program; if no such program is running on the machine, the messages are lost.

Omit this argument to display the current monitor setting.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

If the **-server** argument is omitted, the command displays information about the status of monitoring for the local machine. If monitoring is enabled, the command displays the name of the machine to which monitoring messages are being sent, as follows:

```
Using host host name for monitor services.
```

If monitoring is disabled, the command displays the following output:

```
Cache monitoring is currently disabled.
```

EXAMPLES

The following shows that monitoring messages are being sent to the machine named *machineQ.transarc.com*:

```
% fs mo
Using host machineQ.transarc.com for monitor services.
```

The following sets the Cache Manager's monitoring machine to *machineB.transarc.com*:

```
% fs monitor machineB.transarc.com
fs: new monitor host set.
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs debug

fs newcell — change the list of a cell's database server machines in the kernel.

fs newcell -name <*cell name*> **-servers** <*primary servers*>+ **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs n -n <*cell name*> **-s** <*primary servers*>+ **[-h]**

DESCRIPTION

Removes the Cache Manager's kernel-resident list of database server machines for the cell *cell name*, replacing the list with *primary servers*. The **fs newcell** command does not make permanent changes to the local */afs/vice/etc/CellServDB* file, the contents of which are transferred into the kernel at each reboot, so rebooting the machine causes the contents of the *CellServDB* file to overwrite changes made with this command. Changes made with this command appear in the output of the **fs listcells** command because that command consults the kernel-resident list rather than the *CellServDB* file.

You can use this command to introduce a completely new cell into the kernel-resident list, but you cannot use the command to make a cell inaccessible (remove it from the kernel-resident list by providing no instances for the **-server** argument). To make a cell inaccessible, you must alter the *CellServDB* file and reboot the machine.

WARNING

Some commands, such as the **klog** command, work correctly only when both the *CellServDB* file and the kernel-resident list correctly name a cell's database server machines.

ARGUMENTS

-name	is the complete Internet-style name of the cell for which the kernel-resident list of database server machines is to change. You can name the local cell or a foreign cell.
-servers	names each database server machine for the indicated cell. Provide the complete Internet-style name of each machine.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following changes the machine's kernel-resident list of database server machines for the Transarc Corporation cell to *fs1.transarc.com* and *fs2.transarc.com*:

% fs n transarc.com fs1.transarc.com fs2.transarc.com

PRIVILEGE REQUIRED

Issuer must be logged in as "root" in the UNIX file system of the machine on which the command is issued.

MORE INFORMATION

fs listcells

fs quota — show percent of quota used for volume containing directory/file.

fs quota [-path <*dir/file path*>$^{+}$**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs q [-p <*dir/file path*>$^{+}$**] [-h]**

DESCRIPTION

Displays the percent of maximum quota currently used by the volume that contains each specified directory or file. This is the least informative but quickest **fs** command that provides quota information about a volume. The **fs examine** and **fs listquota** commands provide more complete information.

The system administrator may set quota with the **fs setquota** or **fs setvol** command.

ARGUMENTS

-path specifies each file and/or directory for which quota information about the host volume is desired. If the issuer omits this argument, the current directory is assumed.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output reports the percent of quota used. It does not name the host volume.

EXAMPLES

The following lists the percent quota used of the volume housing the current working directory:

```
% fs quota
17% of quota used.
```

The following lists the percent quota used of both the volume housing the current working directory's parent directory and the volume housing the directory named */afs/transarc.com/usr/smith*:

```
% fs quota .. /afs/transarc.com/usr/smith
43% of quota used.
92% of quota used.
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs examine

fs listquota

fs setquota

fs setvol

fs rmmount — destroy mount point.

fs rmmount -dir <*directory*>$^{+}$ **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs rm -d <*directory*>$^{+}$ **[-h]**

DESCRIPTION

Removes the mount point called *directory* from the file system. The corresponding volume remains in the system, but will be inaccessible if there are no other mount points for it.

ARGUMENTS

-dir names the mount point to be deleted from the file system. The last element in the pathname that the issuer provides must be an actual name, not "dot" (.) or "dot dot" (. .), which the **fs** command interpreter does not understand in this case.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following removes the mount points *jones* and *terry* from the current working directory (assume it is */afs/transarc.com/usr*).

% fs rm jones terry

PRIVILEGE REQUIRED

Issuer must have DELETE access for the directory containing the mount point.

MORE INFORMATION

fs lsmount

fs mkmount

fs setacl — sets access control list for a directory.

fs setacl -dir <*directory*>+ **-acl** <*access list entries*>+ **[-clear] [-negative] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs sa -d <*directory*>+ **-a** <*access list entries*>+ **[-c] [-n] [-h]**

DESCRIPTION

Puts the specified *access list entries* on the access control list (ACL) of each specified *directory*. Only user and group entries can be placed on an ACL. Machine entries (IP addresses) cannot be placed directly on an ACL; instead, the machine entry must be made a member of a group and the group can then be put on the ACL.

WARNING

If the ACL already grants certain rights to a user or group, the rights specified with *access list entries* replace them, rather than just being added to them.

Setting negative rights is generally unnecessary and not recommended. Simply omitting a user or group from the `Normal rights` list is normally adequate to prevent access. In particular, note that it is futile to deny rights that are granted to *system:anyuser* on the same ACL; all the user needs to do is issue the **unlog** command to receive the denied rights.

ARGUMENTS

-dir specifies each directory for which the access control list is to change. Abbreviated pathnames are interpreted relative to the directory in which the command is issued.

-acl defines a list of one or more entries, each of which specifies

- A user name or group name (letters all lowercase).
- The access right(s) to be associated with the user/group.

in that order, separated by a space. This argument is unusual in requiring two parts for each instance. The accepted abbreviation of each right and the meaning of the right follows:

r READ. Allows the possessor to read the contents of files in the directory and to "stat" (issue **ls -l** for) file and subdirectory elements in the directory.

w WRITE. Allows the possessor to modify the contents of files in the directory and to change their UNIX mode bits with **chmod**.

l LOOKUP. Allows the possessor to list the names of files and subdirectories in the directory (for example, by issuing **ls**). The possessor may "stat" (issue **ls -l** for) the directory itself (but not for files and subdirectories in it) and may examine the directory's ACL.

d DELETE. Allows the possessor to remove files from the directory.

i INSERT. Allows the possessor to create new files in the directory or move existing files into it.

k LOCK. Allows the possessor to run programs that need to issue the "flock" system call on files in the directory.

a ADMINISTER. Allows the possessor to change the directory's ACL.

A, B, C, D, E, F, G, H; by default, these have no meaning to AFS server processes. Administrators and application programs may assign meanings to them and place them on ACLs to control access to the directory's contents in new ways. The letters must be uppercase.

all all seven standard rights (*rlidwka*).

none no rights. Removes the user/group from the ACL, but may not guarantee they have no rights if they belong to groups that remain on the ACL.

read both *r* and *l*.

write all rights except ADMINISTER (*rlidwk*).

It is legal to mix the individual letters and the words within *access list entries*, but not within an individual pairing of user/group and rights.

-clear removes all existing entries on each access control list before placing *access list entries* on it. This should be used with caution: if *access list entries* does not grant all rights to the owner of the directory, it can become awkward for the owner to access items in the directory. In particular, not having the LOOKUP right makes it impossible to resolve the "dot" (.) and "dot dot" (. .) shorthand from within the directory.

-negative puts the specified *access list entries* in the `Negative rights` section of each access control list. The user/group is thus explicitly denied the indicated rights, even if entries on the accompanying `Normal rights` section of the access control list grant them rights. However, it is possible to unlog to obtain rights granted to *system:anyuser* on the `Normal rights` section of the same ACL; see the WARNING above.

This flag affects all directories and access list entries specified. Its use is not recommended; see the WARNING section above. If the issuer omits this flag, the *access list entries* go into the `Normal rights` section of the access control list.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following example adds two entries to the `Normal rights` part of the current working directory's ACL: the first entry grants READ and LOOKUP rights to *pat:friends*, while the other (using the **write** shorthand) gives all rights except ADMINISTER to *smith*.

% fs sa . pat:friends rl smith write

The following shows the effect of the **-clear** flag on the ACL of the subdirectory *reports* by showing the ACL before and after the command is issued:

```
% fs la reports
Access list for reports is
Normal rights:
   system:authuser rl
   pat:friends rlid
   smith rlidwk
   pat rlidwka
Negative rights:
   terry rl

% fs sa -clear reports pat all smith write system:anyuser rl
% fs la reports
Access list for reports is
Normal rights:
   system:anyuser rl
   smith rlidwk
   pat rlidwka
```

The following shows how the **-dir** and **-acl** switches are necessary when more than one directory is specified. The new entry granting READ, LOOKUP, and INSERT rights to *pat:friends* is added to the ACL for both the current directory and its *public* subdirectory.

```
% fs sa -d . public -a pat:friends rli
```

PRIVILEGE REQUIRED

Issuer must have ADMINISTER rights to the directory; the directory's owner and members of *system:administrators* always do.

MORE INFORMATION

fs copyacl

fs listacl

fs setcachesize — set size of disk cache.

fs setcachesize [-blocks <*size in 1K byte blocks (0 => reset*>] [-reset] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs setca [-b <*size in 1K byte blocks (0 => reset)*>] [-r] [-h]

fs cachesize [-b <*size in 1K byte blocks (0 => reset)*>] [-r] [-h]

DESCRIPTION

On machines using a disk cache, changes the amount of local disk space that the Cache Manager may use for its data cache. Specify the number in kilobyte blocks. This command is not operative on machines using memory caching.

To return the cache size to the default value specified in */usr/vice/etc/cacheinfo* on the client's local disk, specify *0* as the number of kilobyte blocks. The *cacheinfo* file is human-readable and visible with the **cat** command. The third and final field is the number of kilobyte blocks allocated to the cache at reboot. The chapter in the *AFS System Administrator's Guide* on client machine configuration further describes the contents of *cacheinfo*.

To return the cache size to the value set when the machine was last booted, use the **-reset** flag instead of the **-blocks** argument. This is normally the amount specified in *cacheinfo*, unless the **-blocks** argument was used on **afsd** to override the *cacheinfo* value.

The **fs getcacheparms** command displays the current actual cache size and the amount of space in use, both for disk and memory caches.

WARNINGS

This command is not operative on machines using memory caching, and will result in an error message.

On machines using a disk cache, do not set the cache size to exceed 90% of the actual disk space available for the cache directory. The cache implementation itself requires a small amount of room on the partition.

ARGUMENTS

-blocks specifies the number of 1 kilobyte blocks the Cache Manager may devote to the cache. Specifying a value of "*0*" sets cache size to the default specified in *cacheinfo*. This implies that the smallest possible cache size is 1 kilobyte, not 0.

-reset returns the cache size to the value set when the machine was last booted. This agrees with the value in *cacheinfo* unless the **-blocks** argument was used on **afsd**.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following sets the disk cache size to 25000 kilobyte blocks.

```
% fs setca 25000
```

Both of the following reset the disk cache size to the value in *cacheinfo*, assuming that the **-blocks** argument on **afsd** was not used.

```
% fs setcachesize 0
% fs setca -r
```

PRIVILEGE REQUIRED

Issuer must be logged in as "root" on the machine on which the command is being issued.

MORE INFORMATION

fs getcacheparms

fs setcell — allow or disallow running of setuid programs from specified cells.

fs setcell -cell <*cell name*>$^{+}$ **[-suid] [-nosuid] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs setce -c <*cell name*>$^{+}$ **[-s] [-n] [-h]**

DESCRIPTION

Determines whether the workstation allows programs whose binary files reside in the indicated cells to execute with setuid privilege. By default, no programs may run with setuid privilege.

Include the **-suid** flag with the command to allow programs from the each specified cell to execute with setuid privilege; include the **-nosuid** flag with the command to prohibit programs from each specified cell from executing with setuid privilege. Omit both flags to prevent programs from the each specified cell from executing with setuid privilege.

When you issue **fs setcell**, you directly alter the client's kernel-resident record of the cell's setuid status, so there is no need to reboot the machine. However, changed settings do not survive reboots of the machine (the settings return to their defaults at reboot). You can alter the reboot defaults by placing the appropriate **fs setcell** command in the machine's initialization file (*/etc/rc* or equivalent).

ARGUMENTS

-cell	names each cell from which to allow or disallow programs to execute with setuid privilege. Provide the complete Internet-style cell name of each cell (unlike the **-cell** argument common to many commands, the cell argument of this command does not accept abbreviated cell names).
-suid	allows programs from *cell name* to execute with setuid privilege. Provide it or provide **-nosuid**. Omit both flags to prevent programs from *cell name* from executing with setuid privilege.
-nosuid	prevents programs from *cell name* from executing with setuid privilege. Provide it or provide **-suid**. Omit both flags to prevent programs from *cell name* from executing with setuid privilege.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following enables programs whose binary files reside in the Transarc Cell to execute with setuid privilege in the local cell:

% fs setc transarc.com -s

PRIVILEGE REQUIRED

Issuer must be logged in as "root" on the machine on which the command is issued.

MORE INFORMATION

fs getcellstatus

fs setquota — sets maximum quota for volume containing specified directory.

fs setquota [-path <*dir/file path*>**] -max** <*max quota in kbytes*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs sq [-p <*dir/file path*>**] -m** <*max quota in kbytes*> **[-h]**

DESCRIPTION

Sets the maximum size quota for the ReadWrite volume that contains the specified directory or file. The **fs examine** and **fs listquota** commands show the current maximum quota. The **fs quota** command shows the percent of maximum quota used.

The **fs setvol** command can be used to set the quota on multiple volumes at once. It can also be used to create messages associated with the volumes.

ARGUMENTS

-path	names the directory or file for which quota on the host volume is to be set. If this argument is omitted, the current working directory is used; in this case, the **-max** argument must be used.
-max	specifies the maximum amount of disk space the volume can use. Express it in kilobyte blocks (a value of *1024* is one megabyte). A value of *0* grants an unlimited quota, but the size of the disk partition that houses the volume places an absolute limit on the volume's maximum size.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following imposes a maximum quota of 3000 kilobytes on the volume that houses the directory */afs/transarc.com/usr/smith*:

% fs sq /afs/transarc.com/usr/smith 3000

PRIVILEGE REQUIRED

Issuer must belong to the *system:administrators* group in the Protection Database.

MORE INFORMATION

fs examine **fs quota**

fs listquota **fs setvol**

fs setserverprefs — set Cache Manager's preferences for file server machines.

fs setserverprefs [-servers <*machine name and rank*>+**]**
[-file <*dir/file path*>**] [-stdin] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs sets [-se <*machine name and rank*>+**] [-f** <*dir/file path*>**] [-st] [-h]**

fs sp [-se <*machine name and rank*>+**] [-f** <*dir/file path*>**] [-st] [-h]**

DESCRIPTION

Sets the Cache Manager's preference for one or more file server machines. Each Cache Manager stores a table of file server machines and their respective "ranks." A file server machine's rank is an integer in the range from 1 to 65,534 that determines the Cache Manager's preference for selecting the server machine when the Cache Manager must access a ReadOnly replica that resides on it. Ranks bias the Cache Manager to prefer to access replicas on "near" server machines rather than those on "distant" server machines.

When the Cache Manager needs to access a ReadOnly replica, it first contacts the Volume Location (VL) Server to ascertain the names of the file server machines on which the replica resides. It then checks its internal table to determine the rank associated with each of the file server machines. After comparing the ranks of the machines, it attempts to access the replica on the server machine that has the lowest integer rank.

If the Cache Manager cannot access the replica on the machine with the lowest rank (possibly because of a server process, machine, or network outage), it attempts to access the replica on the machine with the next lowest rank. It continues in this way until it either accesses the replica or determines that all of the file server machines on which the replica is housed are unavailable.

Each time it is initialized with the **afsd** command, the Cache Manager assigns preferences to any database server machines listed in the local */usr/vice/etc/CellServDB* file that are also file server machines. It stores the preferences as machine IP addresses and associated ranks in the kernel of the client machine. (See the ASSIGNING PREFERENCES section for more information about how the Cache Manager determines actual file server machine ranks.) Because they are stored in the kernel, the preferences are recalculated when the client machine is rebooted.

The Cache Manager assigns ranks to file server machines in the local cell and from foreign cells as necessary. When it needs to access a ReadOnly volume, it first determines the machines on which the replica resides. It then assigns ranks to any of the machines that do not already have them and stores the ranks in the kernel, after which it uses the ranks as the basis of its selection of the file server machine from which to access the replica.

The **fs setserverprefs** command can be used to define or change the rank associated with a local or foreign file server machine. If the Cache Manager has no rank for the machine, the command defines the machine's initial rank. If the Cache Manager already has a rank for the machine, the command changes the rank to match the one specified by the issuer; the old rank is overwritten.

Preferences are specified as pairs of values. The first value is the file server machine, the second the machine's rank. File server machines can be specified by name or by IP address. Depending on the naming service available at the time the command is issued, abbreviated forms of machine names may be allowed. See the introductory *About This Manual* chapter for more information.

Pairs of file server machines and their ranks can be specified

- On the command line with the **-servers** argument.
- From a file with the **-file** argument.
- From stdin with the **-stdin** flag.

The **-file** argument and **-stdin** flag are especially useful for configuring multiple Cache Managers in a cell with the same preferences. The **-file** argument can be used to indicate a file created manually or generated automatically with the **fs getserverprefs** command. Similarly, the **-stdin** flag can be used to accept preferences piped directly from another process (possibly from another Cache Manager with the **fs getserverprefs** command). The **-servers**, **-file**, and **-stdin** arguments and flag are not mutually exclusive, so multiple sources of preferences are permitted.

It is possible for the Cache Manager or a user to assign the same rank to multiple file server machines housing a replica of the same volume. In this case, the Cache Manager uses methods described in the following section, ASSIGNING PREFERENCES, to break the tie. It then increments the ranks of the file server machines from which it does not access the replica.

ASSIGNING PREFERENCES

When initially assigning preferences, the Cache Manager bases the ranks on IP addresses, rather than on actual physical considerations such as location or distance. It calculates file server machine ranks according to the following heuristic:

- If the client machine is also a file server machine, the machine receives a rank of 5000.
- If the client machine is in a subnet, all file server machines in the same subnet as the client machine receive an initial rank of 20000.
- All file server machines in the same network as the client machine receive an initial rank of 30000.
- All file server machines on the distant ends of point-to-point links from the client machine receive an initial rank of 30000.
- All file server machines on networks not directly connected to the client machine receive a rank of 40000.
- All file server machines for which no network locality information can be determined receive a default rank of 40000.

The Cache Manager also considers additional metrics associated with networks, subnets, and interfaces when it determines ranks.

If the same ReadOnly replica is stored on multiple file server machines that have the same rank, the Cache Manager employs the metrics mentioned previously to resolve the duplicate rank collisions. If necessary, the Cache Manager randomizes its ranking of the tied machines. It resolves the ties internally by incrementing by one the ranks of the machines from which it chooses not to access the replica.

Note: The Cache Manager consults preferences only when accessing ReadOnly replicas of volumes. It does not consider the preferences when contacting the VL Server on a database server machine to determine the location of a volume. Its access of database server machines is still random.

ARGUMENTS

-servers	specifies one or more pairs of file server machines and their respective ranks. Identify file server machines by name or by IP address. See the DESCRIPTION section for more information on specifying file server machines and their ranks.
-file	specifies the pathname of a file that contains pairs of file server machines and their respective ranks. Identify file server machines by name or by IP address. See the DESCRIPTION section for more information on specifying file server machines and their ranks.
-stdin	indicates that pairs of file server machines and their respective ranks are to be read from stdin. Identify file server machines by name or by IP address. See the DESCRIPTION section for more information on specifying file server machines and their ranks.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following sets preference ranks for three file server machines. In this example, the server machines have no replicas in common, so no potential collisions are associated with their all having the same rank.

% fs sets -se fs1.transarc.com 10000 fs2.transarc.com 10000 128.2.11.12 10000

The following defines a rank for one file server machine from the command line and reads ranks for additional file server machines from a file named *prefs.txt* in the current directory:

% fs sets -se fs4.transarc.com 10010 -f prefs.txt

PRIVILEGE REQUIRED

Issuer must be logged in as "root" on the machine on which the command is issued.

MORE INFORMATION

fs getserverprefs

fs setvol — set maximum quota and messages for each volume containing specified directory.

fs setvol [-path <*dir/file path*>$^{+}$ **] [-max** <*disk space quota in 1K units*>**]**
[-motd <*message of the day*>**] [-offlinemsg** <*offline message*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs sv [-p <*dir/file path*>$^{+}$**] [-ma** <*disk space quota in 1K units*>**]**
[-mo <*message of the day*>**] [-o** <*offline message*>**] [-h]**

DESCRIPTION

Sets maximum quota for the volumes that contain each specified directory or file. It is also possible to use **-motd** and **-offlinemsg** to create messages associated with the volume, which appear when the **fs examine** command is issued.

The **fs examine** command displays all the information that can be altered with this command. The **fs listquota** command displays maximum quota, and the **fs quota** command displays the percent quota used.

The **fs setquota** command sets maximum quota on one volume at a time.

ARGUMENTS

-path	names each file and/or directory for which quota and messages on the host volumes are to be set. Omit this argument to affect the volume that contains the current working directory.
-max	specifies the maximum amount of disk space the volume can use. Express it in kilobyte blocks (a value of *1024* is one megabyte). A value of *0* grants an unlimited quota, but the size of the disk partition housing the volume places an absolute limit on the volume's maximum size.
-motd	specifies a "message of the day" displayed with the **fs examine** command. It can be used to alert users to anything of interest concerning the volume.
-offlinemsg	specifies a message displayed with the **fs examine** command. It can be used to explain why the volume is currently offline.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

EXAMPLES

The following imposes a 6500 kilobyte quota on the volumes housing the */afs/transarc.com/usr/smith* and */afs/transarc.com/usr/pat* home directories:

```
% cd /afs/transarc.com/usr
% fs sv -p smith pat -ma 6500
```

PRIVILEGE REQUIRED

Issuer must belong to the *system:administrators* group in the Protection Database.

MORE INFORMATION

fs examine

fs listquota

fs quota

fs setquota

fs sysname — report or set CPU/operating system type.

fs sysname [-newsys <*new sysname*>] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs sy [-n <*new sysname*>**] [-h]**

DESCRIPTION

Depending on whether the issuer provides the **-newsys** argument, the command either sets the indicator of CPU/operating system type in the kernel of the machine on which the command is issued or reports the current setting.

If the command is issued on an AFS client machine, the value is set/reported for the machine itself.

If the command is issued on an NFS client machine accessing AFS via the NFS/AFS Translator, then the specified CPU/OS value is set/reported for the NFS client machine. The information is in a record maintained by the AFS client machine serving as the NFS client's NFS/AFS translator machine. The translator machine maintains a separate record for each user logged into the NFS client. This implies that if a user adopts a new identity (UNIX UID) during a login session on the NFS client—perhaps using **su**—he or she must issue this command again. Setting this indicator allows the translator machine to provide the NFS client with the proper version of program binaries when the user issues commands for which the binaries are kept in the AFS file tree.

The Cache Manager's main use of this indicator is as a value for the "*@sys*" variable which can occur in AFS pathnames. As the Cache Manager interprets pathnames, it substitutes the indicator's value for any occurrence of *@sys*. See the EXAMPLES section for an example. (Note that *@sys* should be used sparingly, as it can make the effect of changing directories unpredictable; see the *AFS System Administrator's Guide* for further information.)

ARGUMENTS

-newsys	specifies the new setting of the CPU/operating system indicator for the machine on which it is issued. If the issuer omits it, the output shows the current setting. Consult the *AFS System Administrator's Guide* for a complete list of the legal values and the CPU/OS types they represent.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output reports the machine's system type in the format

```
Current sysname is 'system type'
```

EXAMPLES

The following shows the output produced on a Sun SPARCStation running SunOS 4.1.1:

% fs sy
```
Current sysname is 'sun4c_411'
```

The following defines a machine to be a DECStation running Ultrix 4.3:

% fs sysname pmax_ul43

When the Cache Manager on the machine encounters a pathname with the *@sys* variable in it, it substitutes *pmax_ul43* for the variable. For instance, this machine would interpret the pathname

/afs/transarc.com/@sys/usr/bin

as

/afs/transarc.com/pmax_ul43/usr/bin

and would access the volume corresponding to that directory. A machine whose CPU/OS type was *rt_aos4* would interpret the same pathname as

/afs/transarc.com/rt_aos4/usr/bin

and so would access a volume different from that accessed by the DECStation.

PRIVILEGE REQUIRED

None, if the machine is an NFS client. If the machine is an AFS client, the issuer must be logged into the local UNIX file system as "root."

MORE INFORMATION

fs exportafs

fs whereis — report name of file server machine(s) housing specified file/directory.

fs whereis [-path <*dir/file path*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs whe [-p <*dir/file path*>+**] [-h]**

DESCRIPTION

Returns the name of the file server machines that house each specified directory or file. See the OUTPUT section for a description of the information displayed.

ARGUMENTS

-path specifies each file or directory whose location is to be returned. Each specified file or directory must reside in AFS (not on a local disk). If the issuer omits this argument, the location of the working directory is returned.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output includes a line for each specified directory or file. It names the file server machine on which the volume that houses the specified directory or file resides. A list of multiple machines indicates that the directory or file is in a replicated volume.

Machine names usually have a suffix indicating their cell membership. If some question remains, the **fs whichcell** command names the cell in which a directory or file resides.

EXAMPLES

The following indicates that the directory */afs/transarc.com* resides in a replicated volume located on both *fs1.transarc.com* and *fs3.transarc.com*:

```
% fs whe /afs/transarc.com
File /afs/transarc.com is on hosts fs1.transarc.com
  fs3.transarc.com
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs whichcell **fs wscell**

fs whichcell — return name of cell to which specified file/directory belongs.

fs whichcell [-path <*dir/file path*>$^{+}$**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fs whi [-p <*dir/file path*>$^{+}$**] [-h]**

DESCRIPTION

Returns the name of the cell in which the volume that houses each indicated directory or file resides. See the OUTPUT section for a description of the information displayed.

ARGUMENTS

-path specifies each file and/or directory whose cell membership is to be returned. Each specified directory or file must reside in AFS (not on a local disk). If the issuer omits this argument, the cell of the working directory is returned.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The output includes a line for each specified directory or file, naming the cell in which the directory or file resides.

EXAMPLES

The following shows that the current directory resides in a volume in the Transarc Corporation cell:

```
% fs whi
File . lives in cell 'transarc.com'
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs whereis

fs wscell

fs wscell — return name of cell to which workstation belongs.

fs wscell [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

fs ws [-h]

DESCRIPTION

Returns the name of the home cell for the workstation from which the command is issued.

ARGUMENTS

-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one. See section 3.1 for more details.

OUTPUT

The reported cell name comes from the file */usr/vice/etc/ThisCell* on the workstation's local disk.

EXAMPLES

The following results when the **fs wscell** is issued on a machine in the Transarc Corporation cell:

```
% fs wscell
This workstation belongs to cell 'transarc.com'
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

fs whereis

fs whichcell

4. The kas Commands

This chapter defines the **kas** commands that system administrators use to contact the Authentication Server. It assumes the reader is familiar with the concepts described in the *AFS System Administrator's Guide*.

The **kas** command interface allows system administrators to create, modify, examine and delete entries in the Authentication Database maintained by the Authentication Server. Individual users may use the **kas setpassword** command to change their own password (as well as the more standard, non-**kas** command, **kpasswd**).

Refer to the *Command Summary* at the end of this document for a complete list of **kas** commands and their syntax.

4.1. The kas Interface

The **kas** command interface differs slightly from the others described in this manual. The Authentication Server authenticates issuers of **kas** commands directly rather than accepting a ticket from the Ticket Granting Service as most other servers do. Thus most commands require that the issuer provide his or her password at the time of issue.

4.1.1. Interactive Mode

Providing the password each time could get tedious if the user needed to execute a whole set of commands, so **kas** provides an "interactive" mode in which it is not necessary to provide a password repeatedly.

4.1.1.1. Entering Interactive Mode

There are several ways to enter interactive mode:

- Use the **kas interactive** command.
- Type `kas` without any operation code. By default, the command interpreter establishes a connection with each Authentication Server in the local cell. They attempt to authenticate the user logged into the machine from which the command is issued, based on the password the issuer provides at the prompt. The issuer may specify an alternate identity, password, cell name and/or list of Authentication Servers by using the first four common arguments described in section 4.4.
- Type `kas` followed by a user name and cell name, separated by an "@" sign (example: `kas smith@transarc.com`). The Authentication Server attempts to authenticate the specified user in the specified cell, and prompts for his or her password in the specified cell. This method is most useful when the issuer wishes to enter interactive mode with a different identity in a different cell.

Note: If the Authentication Database entry associated with the authentication attempt specifies a limit to the number of consecutive failed authentication attempts, you must correctly type in the corresponding password within the specified number of times. If you exceed the defined number of consecutive failed authentication attempts, the corresponding account will be locked for the appropriate amount of time.

The maximum number of consecutive failed authentication attempts and the resulting lockout time if the number is exceeded are defined using the **kas setfields** command. You can unlock a locked account using the **kas unlock** command; however, you must be authenticated as a user with `ADMIN` privileges to unlock an account.

4.1.1.2. Effects of Entering Interactive Mode

While in interactive mode:

- The "`ka>`" prompt replaces the issuer's regular prompt.
- It is no longer necessary or legal to type **kas** at the beginning of a command. Type the operation code as the first part of the command.
- The Authentication Server does not prompt for the issuer's password at each command. This is the mode's main advantage.
- The issuer's identity and cell are set and cannot be changed without leaving interactive mode, so it is not legal to provide any of the common arguments described fully in section 4.4, except for **-help**.

4.2. Information in the Authentication Database

The Authentication Database stores information used to provide authentication services to both users and servers. The database contains the following fields:

- The ***name*** of the entry. For users, this is the login name; for servers, this is the hard-coded name used by convention.
- Four ***status flags*** which are set by default and can be changed using the **kas setfields** command. (The first two are used frequently, while the latter two are used infrequently.) The non-default values of these flags are
  - `ADMIN`. This flag grants the user the ability to issue privileged **kas** commands (Default: `NOADMIN`).
  - `NOCPW`. This flag prevents the user from changing his or her password (Default: `CPW`).
  - `NOTGS`. This flag directs the Ticket Granting Service to refuse to issue tickets to the user (Default: `TGS`).
  - `NOSEAL`. This flag prevents the Ticket Granting Service from using the contents of this entry's key field as an encryption key (Default: `SEAL`).
- the ***key*** and the key version number. A key is a scrambled form of a password.
- The ***date that the user changed his or her password***.
- The ***entry expiration date***. If this is a user entry, the user will be unable to authenticate with the Authentication Server after this date.
- The **expiration date of the password**. If an expiration date appears here, the user must change his or her password before that expiration date. If there is no expiration date on the password, the entry will say: `password will never expire`. The password expiration date is set using the **kas setfields** command.
- The **limit on the number of consecutive failed authentication attempts** permitted before the user account will lock. This number is set by the **kas setfields** command.
- The **lock out time** for the user account if the limit on the number of consecutive failed authentication attempts is exceeded. The lockout time is set by the **kas setfields** command.
- The ***maximum ticket lifetime***. The maximum amount of time after being issued that a ticket can still be valid.

- The ***last modification date*** and the user name of the person who issued the modifying command. A change to a user's own password is not considered a modification for the purposes of this field.

4.3. The Auxiliary Database

While the Authentication Database will contain the limit on the number of failed authentication attempts for each user account, each kaserver process will maintain an auxiliary database in local file space. The auxiliary database must be stored in a file named *kaauxdb* in the */usr/afs/local* directory or in the directory specified by the **kaserver**'s **-localfiles** argument.

The auxiliary database contains a record of each user account, the number of consecutive authentication failures for that account, and the time of the most recent failure. When the time between the most recent authentication failure exceeds the lockout time for a user account, the consecutive authentication failure counter is reset to zero.

Normally, you should unlock locked user accounts using **kas unlock**. However, if you need to unlock all locked user accounts at one time, shutdown all of the **kaserver** processes and remove all corresponding *kaauxdb* files. The **kaserver** process will recreate this file when it is restarted.

4.4. Common Arguments and Flags

When not in interactive mode, most **kas** commands accept the following optional arguments and flags. Some of these are unavailable in interactive mode because the information they provide is established while entering interactive mode, and cannot be changed from within interactive mode. They are listed in the command descriptions where they apply, and are described in detail below:

[-admin_username <*admin principal to use for authentication*>**]**

This argument allows the issuer to assume the identity of the specified user name (which is referred to as an "*admin principal*"). By default, the Authentication Server attempts to authenticate the user logged into the local workstation.

[-password_for_admin <*admin password*>]

This argument provides the Authentication Server with the password needed to prove that the issuer of the command (the *admin principal*) is legitimate (which it will if it matches the password stored in the Authentication Database for the issuer). Note that providing this argument on the command line reveals the password on the screen. Issuers may prefer to respond to the prompt that will appear if this argument is not provided, as the password does not echo visibly in that case.

Note: If the Authentication Database entry associated with the authentication attempt specifies a limit to the number of consecutive failed authentication attempts, you must correctly type in the corresponding password within the specified number of times. If you exceed the defined number of consecutive failed authentication attempts, the corresponding account will be locked for the appropriate amount of time.

The maximum number of consecutive failed authentication attempts and the resulting lockout time if the number is exceeded are defined using the **kas setfields** command. You can unlock a locked account using the **kas unlock** command; however, you must be authenticated as a user with `ADMIN` privileges to unlock an account.

[-cell <*cell name*>]

This argument specifies that the command should be run in the cell specified by *cell name*. The issuer may abbreviate *cell name* to the shortest form that distinguishes it from the other cells listed in */usr/vice/etc/CellServDB* on the client machine on which the command is issued. By default, commands are executed in the local cell, as defined:

- First, by the value of the environment variable AFSCELL.
- Second, in */usr/vice/etc/ThisCell* on the client machine on which the command is issued.

[-servers <*explicit list of authentication servers*>+]

This argument causes the **kas** command interpreter to establish a connection with the Authentication Server running on each specified database server machine. It then chooses one of these at random to execute each subsequent command. The issuer may abbreviate the machine name to the extent the cell's name server will accept.

By default, the **kas** command interpreter establishes a connection with each machine listed in the local workstation's copy of */usr/vice/etc/CellServDB*, and then chooses one of those at random for command execution.

This option is useful for testing specific servers if problems are encountered.

[-noauth]

This flag instructs indicated Authentication Server(s) not to authenticate the issuer of the command, and thus establishes an unauthenticated connection between the issuer and the Authentication Server (he or she is recognized as the unprivileged user *anonymous*). It is useful only when authorization checking is disabled on the file server machine (during the installation of a file server machine or when **bos setauth**

has been used during other unusual circumstances). In normal circumstances, the Authentication Server allows only authorized (privileged) users to issue most **kas** commands, and will refuse to execute the requested actions even if the **-noauth** flag is used.

[-help]

This flag has the same function as the **kas help** command: it prints the command's online help message on the screen. No other arguments or flags should be provided at the same time. Even if they are, this flag overrides them, and the only effect of issuing the command is that the help message appears.

4.5. The Privilege Required for kas Commands

The Authentication Server considers privileged those users who have the `ADMIN` flag turned on in their Authentication Database entry. See the **kas setfields** command to learn how to turn on the `ADMIN` flag.

Most **kas** commands require that the issuer be privileged. All commands will prompt for a password, unless the issuer has entered interactive mode.

kas apropos — show each help entry containing keyword.

kas apropos -topic <*help string*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

kas a -t <*help string*> **[-h]**

DESCRIPTION

Displays the first line of the help entry for any **kas** command that has *help string* in its name or short description.

ARGUMENTS

-topic specifies the keyword string to search for. If it is more than a single word, surround it with double quotes or other delimiters. Type all *help string*s for **kas** commands in all lowercase letters, except the word "*AuthServer*."

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

The first line of a command's online help entry names the command and briefly describes what it does. The **kas apropos** command displays that first line for any **kas** command where *help string* is part of the command name or first line.

To see the remaining lines in a help entry, which provide the command's alias (if any) and syntax, use the **kas help** command.

EXAMPLES

The following lists all **kas** commands that have the word "*key*" in their operation code or short online description.

```
% kas a key
getrandomkey: get a random key
setkey: set a user's key
stringtokey: convert a string to a key
```

PRIVILEGE REQUIRED

None. No password will be prompted for.

MORE INFORMATION

kas help

kas create — create an entry in the Authentication Database.

kas create -name <*name of user*> **[-initial_password** <*initial password*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth]**
[-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas c -na <*name of user*> **[-i** <*initial password*>**]**
[-a <*admin principal to use for authentication*>**] [-p** <*admin password*>**]**
[-c <*cell name*>**] [-s** <*explicit list of authentication servers*>+**] [-no] [-h]**

DESCRIPTION

Creates an entry in Authentication Database for *name of user*. The Authentication Server converts *initial password* into a form suitable for use as an encryption key, and places it in the entry's key field.

ARGUMENTS

-name	specifies the name of the new Authentication Database entry. It will be the user name under which the user logs in to the system.
-initial_password	specifies a character string that will become the user's password. The Authentication Server scrambles it into an octal key and places it in the key field of the Database entry. If the issuer does not provide it, it will be prompted for (if it is not provided at the prompt the create operation fails). The advantage of waiting for the prompt is that the password does not echo on the screen.
-admin_username	specifies the user name under which the issuer wishes to perform the command. See section 4.4 for more details.
-password_for_admin	specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.
-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

The following shows the prompts that appear when someone authenticated as *admin* creates an Authentication Database entry for the user *smith*, and does not specify *smith*'s initial password on the command line. The passwords typed at the prompts do not echo visibly.

```
% kas cr smith
Password for admin:
initial_password:
Verifying, please re-enter initial_password:
```

PRIVILEGE REQUIRED

Issuer must have the `ADMIN` flag set in his or her Authentication Database entry.

MORE INFORMATION

kas examine

kas debuginfo — produce debugging trace for the Authentication Server.

kas debuginfo [-hostname <*authentication server host name*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>**+] [-noauth]**
[-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas deb [-ho <*authentication server host name*>**]**
[-a <*admin principal to use for authentication*>**] [-p** <*admin password*>**]**
[-c <*cell name*>**] [-s** <*explicit list of authentication servers*>**+] [-n] [-he]**

DESCRIPTION

Displays information on the standard output device (stdout) which is helpful in debugging the Authentication Server. This information is useful only for someone familiar with the implementation of the Authentication Server and the internal structure of the Authentication Database. Most system administrators will not find it helpful.

ARGUMENTS

-hostname	names the database server machine for which to display debugging information about the Authentication Server instance it is running.
-admin_username	specifies the user name under which the issuer wishes to perform the command. See section 4.4 for more details.
-password_for_admin	specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.
-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

PRIVILEGE REQUIRED

None. A password will be prompted for unless the **-noauth** flag is provided. The issuer must type some character string at the prompt, but even providing the wrong one does not prevent the command from being executed, despite possible error messages.

MORE INFORMATION

kas statistics

kas delete — delete entry from the Authentication Database.

kas delete **-name** <*name of user*>
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**]** **[-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**]** **[-noauth]**
[-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas del **-na** <*name of user*> **[-a** <*admin principal to use for authentication*>**]**
[-p <*admin password*>**]** **[-c** <*cell name*>**]**
[-s <*explicit list of authentication servers*>$^{+}$**]** **[-no]** **[-h]**

kas rm **-na** <*name of user*> **[-a** <*admin principal to use for authentication*>**]**
[-p <*admin password*>**]** **[-c** <*cell name*>**]**
[-s <*explicit list of authentication servers*>$^{+}$**]** **[-no]** **[-h]**

DESCRIPTION

Removes the entry *name of user* from the Authentication Database. If *name of user* is a user, he or she becomes unable to log in. If *name of user* is a server, it becomes unreachable, because the TGS no longer has anything with which to seal tickets for the server.

ARGUMENTS

-name	specifies the name of the Database entry to delete.
-admin_username	specifies the user name under which the issuer wishes to perform the command. See section 4.4 for more details.
-password_for_admin	specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.
-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

In the following the issuer *admin* enters interactive mode in order to make it more convenient to delete three accounts at once. The password typed at the prompt does not echo visibly.

```
% kas
Password for admin:
ka> del smith
ka> del pat
ka> del terry
```

PRIVILEGE REQUIRED

Issuer must have the ADMIN flag set in his or her Authentication Database entry.

MORE INFORMATION

kas create

kas examine — display information from an Authentication Database entry.

kas examine -name <*name of user*>
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>**+] [-noauth]**
[-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas e -na <*name of user*> **[-a** <*admin principal to use for authentication*>**]**
[-p <*admin password*>**] [-c** <*cell name*>**]**
[-s <*explicit list of authentication servers*>**+] [-no] [-h]**

DESCRIPTION

Formats and displays information from the Authentication Database entry for *name*. See the OUTPUT section for details.

ARGUMENTS

-name	specifies the Database entry from which to display information.
-admin_username	specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.
-password_for_admin	specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.
-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

The output reports, in this order:

- The name of the entry.
- One or more status flags, which will only appear if a system administrator has used the **kas setfields** command to change a flag from its default value. A plus sign (+) will separate the flags if more than one appears. The non-default values which may appear, and their meanings, are:
-
  - ADMIN. the user is allowed to issue privileged **kas** commands (Default: NOADMIN.)
  - NOTGS. the Ticket Granting Service will refuse to issue tickets to the user (Default: TGS.)
  - NOSEAL. the Ticket Granting Service cannot use the contents of this entry's key field as an encryption key (Default: SEAL.)
  - NOCPW. the user or server cannot change his/her/its own password or key (Default: CPW.)
- The word "key" followed by the key checksum in parentheses.

  The octal key itself appears only if authorization checking is disabled on the database server machine to which the **kas examine** command is directed with the **-servers** argument (see the EXAMPLES section). The reasoning behind this requirement is two-fold. First, it implies that only someone authorized to issue the **bos setauth** command or with "root" access to the database server machine's local disk is able to see actual keys from the Authentication Database. Second, it makes it clear that the system is in a compromised state of security while keys are being displayed on the screen. Both turning off authorization checking and displaying keys on a screen are serious security risks.

  In the normal cases when authorization checking is enabled on the database server machine, a "checksum" appears instead of the key. This is a decimal number derived by encrypting a constant with the key. In the case of the "*afs*" key, this number can be compared to the checksum with the corresponding key checksum in the output of the **bos listkeys** command.
- The date that the user changed his/her own password, indicated as "last cpw" (which stands for "last change of password")
- The expiration date of the password. If an expiration date appears here, the user must change his or her password before that expiration date. A regular user cannot authenticate using an expired password; a system administrator can get an admin ticket for up to 30 days after expiration. If there is no expiration date on the password, the entry will say: password will never expire. The password expiration date is set using the **kas setfields** command.
- The limit on the number of consecutive failed authentication attempts permitted before the user account will lock. This number is set by the **kas setfields** command.
- The lock out time for the user account if the limit on the number of consecutive failed authentication attempts is exceeded. (This entry will only appear if there is a limit on the number of failed authentication attempts.) The lockout time is set by the **kas setfields** command.

- The lock state of the user account. (This entry will only appear if there is a limit on the number of failed authentication attempts.) If the account is locked, the entry will say `User is locked until` *day/date/time*. If the account is not currently locked, the entry will say `User is not locked`. A system administrator can unlock a locked account by issuing the **kas unlock** command.
- The date on which the entry expires. If this is a user entry, the user will be unable to authenticate with the Authentication Server after this date.
- The maximum length of time that tickets issued for this entry may be valid
- The date of the last modification to the entry, indicated as "`last mod`," and the user name of the person who issued the modifying command. Password changes made by the user himself/herself are recorded as "`last cpw`" instead.

EXAMPLES

The following shows the privileged user *smith* examining her own Authentication Database entry. Note the `ADMIN` flag, which shows that *smith* is privileged.

```
% kas e smith
Password for smith:
User data for smith (ADMIN)
 key (0) cksum is 3414844392,  last cpw: Thu Dec 23 16:05:44 1993
 password will expire:  Fri Jul 22 20:44:36 1994
 5 consecutive unsuccessful authentications are permitted.
 The lock time for this user is 25.5 minutes.
 User is not locked.
 entry never expires. Max ticket lifetime 100.00 hours.
 last mod on Thu Jul 1 08:22:29 1993 by admin
```

In the following example, the user *erz* examines her Authentication Database entry to determine if her account is locked. Sure enough, it is.

```
% kas e erz
Password for erz:
User data for erz
 key (0) cksum is 73829292912,  last cpw: Wed Dec 8 11:23:01 1993
 password will expire:  Sat Jul 9 11:23:01 1994
 5 consecutive unsuccessful authentications are permitted.
 The lock time for this user is 25.5 minutes.
 User is locked until Tue Sep 20 12:25:07 1993
 entry expires on never. Max ticket lifetime 100.00 hours.
 last mod on Thu Jul 1 08:22:29 1993 by admin
```

In the following an administrator logged in as the privileged user *admin* uses **bos setauth** to turn off authorization checking on the database server machine *db1.transarc.com* so that he can look at the key in the *afs* entry. He enters interactive mode to open a connection with the Authentication Server on *db1.transarc.com* only and uses the **-noauth** flag to prevent that server from attempting to authenticate him.

```
% bos setauth db1.transarc.com off
% kas i -servers db1.transarc.com -noauth
ka> examine afs -servers db1.transarc.com
```

```
User data for afs
 key (12): \357\253\304\352a\236\253\352, last cpw: no date
 entry never expires. Max ticket lifetime 100.00 hours.
 last mod on Thu Jan 11 14:53:29 1990 by admin
```

PRIVILEGE REQUIRED

A user may examine his or her own entry. To examine others' entries, the issuer must have the ADMIN flag set in his or her Authentication Database entry.

To look at actual keys, authorization checking must be disabled on the database server machine with **bos setauth**, which implies being listed in */usr/afs/etc/UserList*; it is not necessary to have the ADMIN flag in addition.

MORE INFORMATION

bos listkeys

bos setauth

kas setfields

kas forgetticket — discard all tickets for the issuer.

kas forgetticket [-name <*name of server*>] [-all] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas f [-n <*name of server*>**] [-a] [-h]**

DESCRIPTION

Discards all tickets that the Authentication Server has registered for the issuer. This includes the AFS server ticket from each cell in which the user has authenticated, and any tickets that the user may have acquired during this **kas** session (either by virtue of entering the session or using **kas getticket**).

ARGUMENTS

-name specifies which ticket should be discarded. Provide this argument or the **-all** flag.

Note: This argument is not currently implemented: all tickets are discarded no matter which ticket name is specified here.

-all indicates that all tickets should be discarded. Provide this flag or the **-name** argument. In the current implementation, this flag is more sensible because it matches what actually happens no matter which option the issuer chooses.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

Both of the following discard all tickets for the issuer.

```
% kas forgetticket -all
% kas f
```

PRIVILEGE REQUIRED

None. There is no prompt for a password, and the issuer does not have to have the `ADMIN` flag set in his or her Database entry. The deletion can affect only the issuer anyway.

kas getpassword — display octal key from an Authentication Database entry.

kas getpassword -name <*name of user*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

kas getp -n <*name of user*> **[-h]**

DESCRIPTION

Prints out the contents (an octal-format encryption key) of the key field for the Database entry *name of user*.

WARNING

This command works only if the Authentication Server has been compiled with a special flag; this is normally done only for cells in the process of converting from use of AFS 2.0-style authentication to AFS 3.0 authentication. Moreover, this command does not work if issued on an AFS client. The issuer must be logged into a machine running the specially-compiled Authentication Server (a database server machine or other machine running an isolated Authentication Server).

The recommended way to examine the octal form of keys is with **kas examine** when authorization checking is disabled. That command shows a "checksum" when authorization checking is enabled, which may be suitable for some purposes.

Even when the other conditions are met, this command does not work for entries where the name includes a period (these entries are generally for the Authentication Server's internal use anyway).

ARGUMENTS

-name names the entry from which the key field should be printed out.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

The output simply prints the key after a "`Key:`" header. It does not report the key version number, but that is available from **kas examine**.

EXAMPLES

The following shows a user using this command to examine *afs*'s password.

% kas getp afs

```
Key: \020\354\315\310\313\023W\370
```

PRIVILEGE REQUIRED

None. There is no prompt for a password, and the issuer does not have to have the `ADMIN` flag set in his or her Database entry. It is assumed that any machine running the Authentication Server is secured by having only a limited number of people in its local */etc/passwd* file.

MORE INFORMATION

kas examine

kas getrandomkey — generate an encryption key at random.

kas getrandomkey
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**]** **[-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**]**
[-noauth] **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

kas getr **[-a** <*admin principal to use for authentication*>**]** **[-p** <*admin password*>**]**
[-c <*cell name*>**]** **[-s** <*explicit list of authentication servers*>$^{+}$**]** **[-n]** **[-h]**

DESCRIPTION

Returns a key generated at random (i.e., not derived from any known password). This is useful mainly for testing and debugging the Authentication Server.

WARNINGS

The output of this command is visible on the issuer's screen, making it a potential security risk to use the key in an actual Authentication Database entry (such as the "*afs*" entry).

If the **-noauth** flag is used, the connection to the Authentication Server is not authenticated, so the randomly generated key crosses the network unencrypted. The potential security risk in using the key as an actual encryption key is even greater in this case.

ARGUMENTS

-admin_username	specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.
-password_for_admin	specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.
-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

Following a "`Key:`" header, the output displays the octal and hexadecimal forms of the key.

EXAMPLES

The following shows *terry* issuing this command.

% **kas getr**

```
Password for terry:
Key: \230\206\370v1\334\373\346 (98.86.f8.76 31.dc.fb.e6)
```

PRIVILEGE REQUIRED

None, but the correct password must be provided at the prompt if the key is to be encrypted while crossing the network. If the **-noauth** flag is used, a password is not prompted for.

kas getticket — create a ticket valid for the specified server.

kas getticket -name <*name of server*> **[-lifetime** <*ticket lifetime*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth]**
[-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas gett -na <*name of server*> **[-l** <*ticket lifetime*>**]**
[-a <*admin principal to use for authentication*>**] [-p** <*admin password*>**]**
[-c <*cell name*>**] [-s** <*explicit list of authentication servers*>+**] [-no] [-h]**

DESCRIPTION

Instructs the Ticket Granting Service to create a ticket for the issuer of the command, which he or she can use for secure communication with *name of server*. The ticket is sealed with the contents of the key field in *name of server*'s Database entry.

This command is functionally similar to **klog**. It is useful primarily for debugging purposes, and is not recommended as the normal way to obtain tickets. Also, while it is possible to make *name of server* an individual user as well as a server process, such a ticket is useless because the user's workstation will not know what to do with it.

ARGUMENTS

-name specifies the Database entry from which to use the key. It may include a cell specification. For example, typing

kas getticket afs@transarc.com

gets a ticket good for all AFS servers in the Transarc Corporation cell.

-lifetime specifies an upper limit on the maximum lifetime of this ticket. The specified value cannot exceed the maximum lifetime specified in the server's Authentication Database entry, which is specified with the **kas setfields** command. To specify a number of hours, include a colon in the number (e.g., `1:00` means one hour). Otherwise, the number is assumed to be in seconds (so `3600` means one hour). If this argument is not provided, the default setting is 100:00 hours (360000 seconds).

-admin_username specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.

-password_for_admin specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.

-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

The following shows the regular user *terry* obtaining a ticket for the AFS server processes in her home cell.

```
% kas gett afs
Password for terry:
```

PRIVILEGE REQUIRED

None. However, the issuer must provide his or her correct password.

kas help — show syntax of specified **kas** command(s) or list functional description for all of them.

kas help [-topic <*help string*>+] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas h [-t <*help string*>+] [-h]

DESCRIPTION

Displays the first line (name and short description) of every **kas** command's help entry, if no *help string* is provided. For each operation code specified with **-topic**, it outputs the entire help entry. See the **Output** section below.

ARGUMENTS

-topic specifies the operation code(s) for which syntax is to be provided. If the issuer omits it, the output instead provides a short description of all **kas** commands.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

The online help entry for each **kas** command consists of two or three lines:

- The first line names the command and briefly describes what it does.
- If the command has aliases, they will appear on the next line.
- The final line, which begins with "`Usage:`", lists the command's arguments and flags in the prescribed order. Online help entries use the same symbols (brackets, etc.) as the command definitions in this manual. For an explanation of their meaning, see page x of the introductory *About This Manual* chapter.

EXAMPLES

The following displays the online help entry for the **kas setpassword** command.

% kas help setpassword

```
kas setpassword: set a user’s password
aliases: sp
Usage: kas setpassword -name <name of user>
[-new_password <new password>] [-kvno <key version number>]
[-admin_username <admin principal to use for authentication>]
[-password_for_admin <password>] [-cell <cell name>]
[-servers <explicit list of authentication servers>+] [-help]
```

PRIVILEGE REQUIRED

None. No password will be prompted for.

MORE INFORMATION

kas apropos

kas interactive — enter interactive mode for Authentication Server.

kas interactive [-admin_username <*admin principal to use for authentication*>**] [-password_for_admin** <*admin password*>**] [-cell** <*cell name*>**] [-servers** <*explicit list of authentication servers*>+**] [-noauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

kas i [-a <*admin principal to use for authentication*>**] [-p** <*admin password*>**] [-c** <*cell name*>**] [-s** <*explicit list of authentication servers*>+**] [-n] [-h]**

DESCRIPTION

Enters interactive mode. By establishing an authenticated connection in this way, the issuer will not have to type his or her password for each command as would be necessary in regular mode. The authenticated connection lasts for one hour unless the maximum ticket lifetime for the issuer or the Authentication Server is shorter.

It is also possible to establish an unauthenticated connection under the identity *anonymous* by using the **-noauth** flag. During normal operation, there is no point to doing so, because the Authentication Server still does authorization checking and will not allow *anonymous*, who is unprivileged by definition, to perform privileged commands.

A possible situation in which an issuer might wish to enter interactive mode unauthenticated is if he or she knows that attempting to authenticate will cause a problem, but wants to list some unprivileged information. Attempting to authenticate could cause a problem, for instance, if the Authentication Server no longer knows the key used to seal the ticket the user has (perhaps it is no longer in */usr/afs/etc/KeyFile*).

The other repercussions of entering interactive mode are:

- A "`ka>`" prompt replaces the issuer's regular prompt
- It is no longer necessary or legal to type **kas** at the beginning of a command. Type the operation code as the first part of the command
- It is not useful to include the common arguments described in section 4.4: **-admin_username**, **-password_for_admin**, **-cell**, **-servers**. They are ignored, because the variables corresponding to them are set as the issuer enters interactive mode, and cannot be changed without leaving interactive mode. It is legal to provide the **-help** flag.

There are two additional ways to enter interactive mode:

1. Type `kas` without any operation code. By default, the command interpreter establishes a connection with all of the Authentication Servers in the local cell. They attempt to authenticate the user logged into the machine from which the command is issued, based on the password the issuer provides at the prompt. The issuer may specify an alternate identity,

password, cell name and/or list of Authentication Servers by using the first four common arguments described in section 4.4.

2. Type `kas` followed by a user name and cell name, separated by an "@" sign (example: `kas smith@transarc.com`). The Authentication Server attempts to authenticate the specified user in the specified cell, and prompts for his or her password in the specified cell. This method is most useful when the issuer wishes to enter interactive mode with a different identity than the one under which he or she is currently logged on.

ARGUMENTS

-admin_username	specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.
-password_for_admin	specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.
-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. Using this argument on this command is useful only when authorization checking is disabled on the file server machine (during the installation of a file server machine or when **bos setauth** has been used during other unusual circumstances). Under normal authorization checking circumstances, the Authentication Servers will allow only authorized (privileged) users to issue commands that change the status of a server or configuration file, even if the **-noauth** flag was used when entering interactive mode. See section 4.4 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

The following shows a user entering interactive mode as the privileged user *admin*.

```
% kas i admin
Password for admin:
ka>
```

PRIVILEGE REQUIRED

None. A password will be prompted for, and something must be typed in response, but the issuer needs to provide the correct password only if he or she wishes to issue privileged commands while in interactive mode. If he or she provides a wrong character string, the Authentication Server assigns the unprivileged identity *anonymous*.

MORE INFORMATION

(kas) noauthentication

(kas) quit

kas list — list all entries in the Authentication Database.

kas list [-long] [-showadmin]
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**]**
[-noauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas ls [-l] [-sh] [-a <*admin principal to use for authentication*>**]**
[-p <*admin password*>**] [-c** <*cell name*>**]**
[-se <*explicit list of authentication servers*>+**] [-n] [-h]**

DESCRIPTION

Displays information on entries in the Authentication Database. The amount and type of information displayed depends on the flag used with the command:

- To display only the names associated with all entries (servers and users), issue the command with no flags.
- To display the full Authentication Database entry for all users, issue the command with the **-long** flag.
- To display the full Authentication Database entry for all users that have the ADMIN flag turned on, issue the command with the **-showadmin** flag.

ARGUMENTS

-long directs the command to display the full Authentication Database entries for all users in the database.

-showadmin directs the command to display the full Authentication Database entries for those users that have the ADMIN flag turned on.

-admin_username specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.

-password_for_admin
specifies the issuer's password. Note that provided here the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.

-cell specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.

-servers specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.

-noauth establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

The output of this command depends upon the flag used with it:

- If no flags are specified, the output only contains the name associated with each entry (both servers and users).
- If the **-long** or **-showadmin** flag is specified, the following information appears for each entry (all users or all ADMINs, respectively):
  - The ***name*** of the entry.
  - Four ***status flags*** which are set by default and can be changed using the **kas setfields** command. The non-default values of these flags are
    - ADMIN. This flag grants the user the ability to issue privileged **kas** commands (Default: NOADMIN).
    - NOCPW. This flag prevents the user from changing his or her password (Default: CPW).
    - NOTGS. This flag directs the Ticket Granting Service to refuse to issue tickets to the user (Default: TGS).
    - NOSEAL. This flag prevents the Ticket Granting Service from using the contents of this entry's key field as an encryption key (Default: SEAL).
  - The ***key*** and the key version number.
  - The ***date that the user changed his or her password***.
  - The ***entry expiration date***.
  - The ***maximum ticket lifetime***.
  - The ***last modification date*** and the user name of the person who issued the modifying command.

Note: Section 4.2 contains detailed information on these fields.

Some entries in the Authentication Database are created by the Authentication Server for its own internal use (for instance, *AuthServer.Admin*).

EXAMPLES

The following example shows the output of the **kas list** command with no flags. The list is taken from the cell *small.edu* which has five users.

% kas list

```
AuthServer.Admin
krbtgt.SMALL.EDU
afs
admin
pat
smith
jones
terry
```

The following example shows the output of the **kas list** command using the **-long** flag. Again, the listing is taken from the cell *small.edu*.

% **kas list -l -a admin**

```
User data for admin (ADMIN)
  key (0) cksum is 1563234467, last cpw:  no date
  entry never expires.  Max ticket lifetime  100.00 hours.
  last mod on Tue Jun 16 14:05:17 1992 by admin

User data for pat
  key (0) cksum is 1663454987, last cpw:  Sat Apr 25 09:21:27 1992
  entry never expires.  Max ticket lifetime  25.00 hours.
  last mod on Thu Jul 16 11:27:52 1992 by admin

User data for smith
  key (0) cksum is 1684494982, last cpw:  Tue May 26 10:23:43  1992
  entry never expires.  Max ticket lifetime  25.00 hours.
  last mod on Thu Jul 16 11:29:01 1992 by admin

User data for jones
  key (0) cksum is 1636484921, last cpw:  Mon Apr 27 19:27:21 1992
  entry never expires.  Max ticket lifetime  25.00 hours.
  last mod on Thu Jul 16 11:30:43 1992 by admin

User data for terry
  key (0) cksum is 1766494217, last cpw:  Fri Jul 10 11:53:17 1992
  entry never expires.  Max ticket lifetime  25.00 hours.
  last mod on Thu Jul 16 11:32:12 1992 by admin
>
```

PRIVILEGE REQUIRED

Issuer must have the ADMIN flag set in his or her Authentication Database entry.

kas listtickets — list all tickets (tokens) for issuer.

kas listtickets [-name <*name of server*>] [-long] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas listt [-n <*name of server*>] [-l] [-h]

DESCRIPTION

Lists all the tickets (tokens) for the issuer, if no arguments are provided. If *name of server* is provided, lists only the ticket good for that server process ("*afs*" is the most common server). If the **-long** flag is provided, the output shows octal strings representing the session key and ticket itself.

ARGUMENTS

-name specifies the server name for which to list the ticket.

-long specifies that the output should display the octal number strings constituting the session key and ticket, along with other information (see the OUTPUT section). The session key is the one also found within the ticket.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

The output reports for whom the ticket is valid and the ticket's expiration date. If no cell specification appears, then the ticket is valid in the local cell.

If the **-long** flag is used, the output displays the octal numbers making up the session key and ticket, along with the key version number and the number of bytes in the ticket (this should always be 56). If a "[>> POSTDATED <<]" marker appears where the expiration date normally does, the ticket does not become valid until the indicated time. (Only internal calls can create a postdated ticket; there is no standard interface that allows users to do this.)

EXAMPLES

The following two examples are for a user with AFS UID 1020 in the *transarc.com* cell and AFS UID 35 in the *test.transarc.com* cell. He is working on a machine in the former cell and is authenticated in both cells.

% **kas listtickets**

```
User’s (AFS ID 1020) tokens for afs [Expires Wed Jan 3 9:30:54 1990]
User’s (AFS ID 35@test.transarc.com) tokens for
afs@test.transarc.com [Expires Wed Jan 3 13:54:26 1990]
```

% **kas listtickets afs -l**

```
User’s (AFS ID 1020) tokens for afs [Expires
Wed Jan 3 9:30:54 1990]
SessionKey: \375\205\351\227\032\310\263\013
Ticket: (kvno = 0, len = 56): \033\005\221S\203d\312H\016\133\ (etc.)
```

PRIVILEGE REQUIRED

None. A password is not prompted for.

kas noauthentication — discard authenticated identity in interactive mode.

(kas) noauthentication [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

(kas) n [-h]

DESCRIPTION

Instructs the Authentication Server to close the (presumably authenticated) connection established when the issuer entered interactive mode. It opens a new unauthenticated connection, assigning the issuer the unprivileged identity *anonymous*. It does not actually flush the user's tokens from the Cache Manager's memory (as **unlog** or **kas forgetticket** do).

This command is useful only in interactive mode, in which case the issuer should omit the **kas** command suite name. If issued in non-interactive mode, it has no effect.

Like the **-noauth** flag available on a few **kas** commands (e.g., **kas interactive**) this command is only useful in unusual circumstances. During normal operation, there is no point to throwing away an authenticated identity (becoming *anonymous*). The Authentication Server still does authorization checking and will not allow *anonymous*, who is unprivileged by definition, to perform privileged commands.

ARGUMENTS

-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

The following discards the authentication information with which the user entered interactive mode.

```
kas> no
```

PRIVILEGE REQUIRED

None. Because this command may be issued only while in interactive mode, no password is prompted for.

MORE INFORMATION

kas interactive

kas quit — leave interactive mode.

(kas) quit [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

(kas) q [-h]

DESCRIPTION

Leaves interactive mode. This command terminates the authenticated connection, so the Authentication Server will again require a password when another **kas** command is issued.

This command is useful only in interactive mode, in which case the issuer should omit the **kas** command suite name. If issued in non-interactive mode, it has no effect.

ARGUMENTS

-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

The following shows the return of the normal command shell prompt when the issuer leaves interactive mode.

```
ka> quit
%
```

PRIVILEGE REQUIRED

None. Because this command may be issued only while in interactive mode, no password is prompted for.

MORE INFORMATION

kas interactive

kas setfields — set various flags, expiration date and ticket lifetime for Authentication Database entry.

kas setfields -name <*name of user*>
[-flags <*hex flag value or flag name expression*>**]**
[-expiration <*date of account expiration*>**]**
[-lifetime <*maximum ticket lifetime*>**]**
[-pwexpires <*number days password is valid ([0..254])*>**]**
[-reuse <*permit password reuse (yes/no)*>**]**
[-attempts <*maximum successive failed login tries ([0..254])*>**]**
[-locktime <*failure penalty [hh:mm or minutes]*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**]**
[-noauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas sf -na <*name of user*> **[-f** <*hex flag value or flag name expression*>**]**
[-e <*date of account expiration*>**] [-li** <*maximum ticket lifetime*>**]**
[-pw <*number days password is valid ([0..254])*>**]**
[-r <*permit password reuse (yes/no)*>**]**
[-at <*maximum successive failed login tries ([0..254])*>**]**
[-lo <*failure penalty [hh:mm or minutes]*>**]**
[-ad <*admin principal to use for authentication*>**]**
[-pa <*admin password*>**] [-c** <*cell name*>**]**
[-s <*explicit list of authentication servers*>$^{+}$**] [-no] [-h]**

DESCRIPTION

Changes the Authentication Database entry for *name of user* in the manner specified by the various optional arguments, which may occur singly or in combination. See the ARGUMENTS section for a description of the values that may be set.

The results of this command are visible in the output of the **kas examine** command.

WARNING

The mechanism used to enforce the number of failed authentication attempts set by this command may cause a user account to be locked even though the number of failed attempts is less than the permitted limit. Here is why. The Authentication Database will store the limit on failed authentication attempts (*f*); however, each Authentication Server (*S*) will only know about and keep track of *fdivS* attempts (the sync site will know about and keep track of any remainder *fmodS*). In other words, each Authentication Server will permit only a portion of the configured total failed attempts before refusing to authenticate a user. An Authentication Server will continue to refuse to authenticate a user until the time between the most recent authentication attempt and the current time exceed the configured lockout time.

Under most circumstances, this mechanism will not present a problem. If an authentication attempt is refused by one Authentication Server, another server will be contacted until either all of the available servers have been contacted or a successful authentication is made.

However, if one or more of the Authentication Server processes are unavailable, the limit on failed authentication attempts will be effectively reduced. Furthermore, if you have specified an infinite lockout time, each time a user fails to properly authenticate, the failed authentication limit itself will be effectively reduced by one (since the Authentication Server will never reset its failure counter). This will eventually result in account lockout.

For these reasons, you should follow the recommendations listed below when setting limits on failed authentication attempts and lockout times:

- Unless you have a compelling reason, do not set an infinite lockout time. This will result in continual lockouts that will have to be manually unlocked by the system administrators. It is strongly recommended that you do not set infinite lockout times on system administrator accounts; if all of your system administrator accounts become locked, the only way to remove these locks is to shut down the Authentication Servers and remove the files containing the failure information. (See section 4.3 for information on this operation.)
- Use short lockout times. Remember, although guessing passwords is a common method of attack, it is by no means a sophisticated method. In most cases, you should set you lockout times to discourage attackers; however, being overzealous in this regard will be more of a burden to your users than to potential attackers.
- Do not set the limit on failed authentication attempts too low. The loss of a server or two will reduce the limit even further.

For most cells, Transarc Corporation recommends setting the limit on authentication to 5 attempts and the lockout time to 25 minutes.

The password lifetime set with the **-pwexpires** argument is retroactive. For example, if a user changed his or her password 30 days ago, and you set the password lifetime to 30 or less, the password would expire in approximately 1 day (24 hours from the time that the password was changed). To avoid retroactive expiration, you should instruct users to change their passwords prior to setting a password lifetime.

ARGUMENTS

-name specifies the entry to be affected.

-flags sets any one of four toggling flags in *name*'s entry. The default is for none of the flags to be set. A value of *0* returns all four flags to their defaults. The following explains the four non-default values to set, their meanings and the associated defaults:

- *ADMIN* (Hex equivalent: *0x004*). The *name of user* is allowed to issue privileged **kas** commands (Default: *NOADMIN*).
- *NOTGS* (Hex equivalent: *0x008*). The Ticket Granting Service will refuse to issue tickets for *name of user* (Default: *TGS*).

- *NOSEAL* (Hex equivalent: *0x020*). The Ticket Granting Service cannot use the contents of this entry's key field as an encryption key (Default: *SEAL*).
- *NOCPW* (Hex equivalent: *0x040*). The *name of user* cannot change his/her/its own password or key (Default: *CPW*).

Both upper and lower-case letters are acceptable in specifying values for the flags.

To restore the **ADMIN** flag to its default, specify **NOADMIN**. To restore the other flags to their defaults, omit the **NO** (i.e., type **TGS**, **SEAL** or **CPW**).

To set more than one flag at once, connect them with plus signs (example: **NOTGS+ADMIN+CPW**). To remove all the current flag settings before setting new ones, precede the whole list with an equal sign (example: **=NOTGS+ADMIN+CPW**).

-expiration determines when the entry itself expires, which will render an user unable to log in to the system and a server unreachable. The default is *never*.

There are three types of legal values:

- *never*, which allows the issuer to return the expiration time to its default after having set it to a date.
- *mm/dd/yy* specifies 12:00 a.m. on the indicated date (month/day/year). Examples : *1/23/90*, *10/7/89*.
- *"mm/dd/yy hh:mm"* specifies a time "hh:mm" (hour:minutes) on the indicated date (month/day/year). The time should be in 24-hour format (for example, 20:30 is 8:30 p.m.) Date format is the same as for a date alone. Surround the entire instance with quotes because it contains a space. Examples : *"1/23/90 22:30"*, *"10/7/89 3:45"*.

Legal values for *yy* run from *00* to *37*, which are interpreted as the years 2000-2037, and from *70* to *99* which are interpreted as 1970-1999. (This restriction is because the Authentication Server converts the date into the number of seconds elapsed since 1 February 1970, to comply with the standard UNIX date representation; dates later than sometime in February 2038 exceed the representation's capacity.)

-lifetime specifies the upper limit on the validity lifetime that the TGS may stamp on a ticket issued to an individual or for a server. That is, if *name of user* is an individual, this value is the maximum lifetime of a ticket issued to the user. If *name of user* is a server such as "*afs*," this value is the maximum lifetime of a ticket that the TGS issues to clients in order to contact the server.

To specify a number of hours, include a colon in the number (example: `1:00` means one hour). Otherwise, the number is assumed to be in seconds (so `3600` means one hour). If this argument is not provided, the default setting is 100:00 hours (360000 seconds).

-pwexpires defines the number of days for which *name of user*'s password will be valid from the date that the password is changed. Beyond the specified number of days, the password will be invalid (that is, the user cannot use it to authenticate with his or her account). For up to 30 days after password expiration, the user can use his or her old password to change passwords. See the WARNING section for more information

The range of valid values is 0 through 254. Use 0 to indicate that the password never expires; use 1 through 254 to specify the number of days until expiration. By default, a user's password will never expire.

-reuse specifies whether or not the user can reuse any of his or her last twenty passwords. The valid values for this field are *yes* (the user can immediately reuse old passwords) and *no* (the user can only reuse old passwords after twenty password changes). By default, the user can reuse passwords (*yes*).

-attempts specifies the maximum number of times that a user can consecutively fail to correctly type in his or her password during authentication (AFS **login**, **klog**, etc.,) before being locked out of his or her user account for the amount of time specified by the **-locktime** argument. The range of valid values is 0 through 254. Use 0 to indicate that there is no failed authentication limit; this is the default value.

-locktime specifies the amount of time that a user will be locked out of his or her user account if he or she exceeds the maximum number of failed authentication attempts specified by the **-attempts** argument.

You can specify a time in either hours and minutes (hh:mm) or minutes only (mm). If a 0 is specified, the account will be locked indefinitely (until a system administrator unlocks it using **kas unlock**). If a time is entered (other than 0), it will be rounded up to the next highest increment of 8.5 minutes; so all locked accounts will be locked for some multiple of 8.5 minutes. The maximum finite value is 36 hours; for example, if *99:99* is specified, the locktime will be 36 hours.

-admin_username
specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.

-password_for_admin
specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.

-cell specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.

-servers specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.

-noauth establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

In the following, *admin* grants administrative privilege to *smith*, and makes *smith*'s entry expire at midnight on 31 December 1995.

```
% kas sf smith ADMIN 12/31/95
Password for admin:
```

In the following, *admin* sets the number of days for which *pat*'s password will be valid to 60 days and disallows *pat* from reusing passwords.

```
% kas sf pat -pw 60 -r no
Password for admin:
```

PRIVILEGE REQUIRED

Issuer must have the ADMIN flag set in his or her Authentication Database entry.

MORE INFORMATION

kas examine

kas setkey — insert octal key directly into key field of Database entry.

kas setkey -name <*name of user*> **-new_key** <*eight byte new key*>
[-kvno <*key version number*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**]**
[-noauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas setk -na <*name of user*> **-ne** <*eight byte new key*> **[-k** <*key version number*>**]**
[-a <*admin principal to use for authentication*>**] [-p** <*admin password*>**]**
[-c <*cell name*>**] [-s** <*explicit list of authentication servers*>+**] [-no] [-h]**

DESCRIPTION

Accepts the octal number string *eight byte new key*, places it in the key field of the Database entry for *name of user* and marks it with key version number *key version number*. This command makes most sense when *name of user* is a server process (such as "afs") rather than an individual user, as discussed in the **Warning** section. For individual users, use the **kas setpassword** command instead.

When changing the key in a server entry, remember to alter the */usr/afs/etc/KeyFile* with the **bos addkey** command and use the same key version number in both places.

WARNING

Using this command to change user passwords is needlessly complicating, for the following reason. The Authentication Database stores both user passwords and server keys as octal strings, never as character strings. When a character string is provided in the **kas setpassword** command (see above), the Authentication Server calls an algorithm to convert the string into an octal-form encryption key, then stores the key in the key field of the appropriate entry. The **kas setkey** command (this one) bypasses the conversion algorithm, entering an *eight byte new key* directly into the key field. If an octal string is simply invented, there is no way to discover which character string it corresponds to. To use this command to change a user password, the issuer would need to translate the password character string into an octal key using **kas stringtokey** and specify the resulting string as *eight byte new key* here. Otherwise, the user would not know what password to type at login.

The main use for this command is to set server encryption keys, if the system administrator first executes the **bos addkey** command to add a new key to */usr/afs/etc/KeyFile*. That command accepts only strings and converts them into octal keys, just like **kas setpassword**, but the output from the **bos listkeys** command will then reveal the converted octal form of the string, which is the *eight byte new key* that should be specified here.

ARGUMENTS

-name	specifies the entry for which to replace the key field.
-new_key	specifies a string of eight bytes, each consisting of EITHER a three-digit octal number preceded by a backslash or a single ASCII character. The issuer must provide it on the command line where it is visible, as there is no prompt for it. Outside interactive mode, surround *octalkey* with single quotes, or the shell will throw away the backslashes. When in interactive mode, do not include the single quotes.
-kvno	specifies the key version number associated with the new key. It must be an integer in the range *0* through *255*. In addition, *999* is reserved for the "bcrypt" key used in AFS 2.0 authentication. If the issuer does not provide this argument, it defaults to *0*, which is probably not desirable for server keys.
-admin_username	specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.
-password_for_admin	specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.
-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

The following shows *admin* setting the "*afs*" key with key version number 15.

```
% kas setk afs \222\260\334\241c\212\274W 15
Password for admin:
```

PRIVILEGE REQUIRED

Individual users may use this command to change their own keys, though this is not advised for the reasons stated in the WARNING section. To change server encryption keys or the key field in another user's entry, issuer must have the ADMIN flag set in his or her Authentication Database entry.

MORE INFORMATION

bos addkey

bos listkeys

kas setpassword

kas setpassword — change key field in Database entry.

kas setpassword -name <*name of user*> **[-new_password** <*new password*>**]**
[-kvno <*key version number*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**]**
[-noauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas sp -na <*name of user*> **[-ne** <*new password*>**] [-k** <*key version number*>**]**
[-a <*admin principal to use for authentication*>**] [-p** <*admin password*>**]**
[-c <*cell name*>**] [-s** <*explicit list of authentication servers*>+**] [-no] [-h]**

DESCRIPTION

Accepts a character string *new password* of unlimited length, scrambles it into a form suitable for use as an encryption key, and places it in the key field of *name*'s Authentication Database entry.

There is little reason to type *new password* on the command line, where it is visible. If the issuer does not provide the password on the command line, a

```
new_password:
```

prompt appears, followed by a second prompt requesting a repetition of the new password. In both cases the password does not echo visibly. The second prompt guarantees that the issuer did not make a typing mistake when responding to the first prompt.

When adding server keys (such as "afs"), remember to add the new key to the */usr/afs/etc/KeyFile* using the **bos addkey** command and to use the same key version number there as here. See the *AFS System Administrator's Guide* for instructions.

During execution, this command will make the following checks:

- If the program *kpwvalid* exists in the same directory as **kas setpassword**, it will be executed using both the old and the new password as input. The *kpwvalid* program can either be the example program provided by Transarc Corporation, or a program customized for your cell. Custom *kpwvalid* programs can be written in any compatible programming language or *kpwvalid* can be a shell script.

  To protect against the risk of Trojan horses, *kpwvalid* must reside in the same directory as **kas setpassword**. The directory must be in AFS and must permit write access to *system:administrators* only.

  A custom *kpwvalid* program or shell script must follow the general behavior of the example program provided by Transarc (shown below); however, the exact check(s) made by a custom version of *kpwvalid* are dependent only on the needs and conventions of your cell. A customized program or shell script must write a zero to standard output (stdout) followed by a newline if the new password is valid; otherwise it will write a non-zero decimal number to stdout. Any error messages generated by *kpwvalid* must be directed to standard error (stderr) only; do not direct error messages to stdout.

  Below is the content of the example *kpwvalid* program provided by Transarc. This program checks to make sure that a new password contains eight or more characters:

```
#include <stdio.h>

/* returns 0 if the password is long enough, otherwise non-zero */
main()
{
char oldpassword[512];
char password[512];
int rc;

if (fgets(oldpassword, 512, stdin))
   while (fgets(password, 512, stdin)) {
      if (strlen(password) > 8) { /* password includes a newline */
         rc = 0;
         fputs("0\n",stdout);
         fflush(stdout);
      }
      else {
         rc = 1;
         fputs("Passwords must contain at least 8 characters.\n",
               stderr);
         fputs("1\n",stdout);
         fflush(stdout);
      }

exit(rc);
}
```

- If the user is prohibited from reusing old passwords, **kas setpassword** will check to make sure that the new password does not match any of the last twenty passwords. The **kas setfields** command has an argument that allows you to prohibit the reuse of the last twenty old passwords.

  If a user attempts to create a password that is the same as or similar to a recently used password, the following error message will be displayed:

```
kpasswd:  Password was not changed because it seems like a reused
password
```

Note: To prevent users from getting around this restriction by simply changing their passwords twenty times (or running a script which automatically does this), you can set a minimum time between password changes when **kaserver** is started. If a user attempts to change a password before the minimum time between password changes has passed, the following error message will be displayed:

```
kpasswd:  Password was not changed because you changed it too
recently; see your systems administrator
```

ARGUMENTS

-name	specifies the entry for which to replace the key field.
-new_password	is the character string the user will type when logging in. It should be 8 characters or less, as some non-AFS programs cannot handle longer passwords. The Authentication Server scrambles it into a string of octal numbers before storing it in the key field.
-kvno	specifies the key version number associated with the new key. It must be an integer between *0* and *255*. In addition, *999* is reserved for the "bcrypt" key used in AFS 2.0 authentication. If the issuer does not provide this argument, it defaults to 0, which is probably not desirable for server keys.
-admin_username	specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.
-password_for_admin	specifies the issuer's password. If provided here, the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.
-cell	specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.
-servers	specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.
-noauth	establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

The following shows privileged user *admin* changing *pat's* password (presumably because *pat* forgot the former password or got locked out of his account in some other way.

% **kas sp pat**

```
Password for admin:
new_password:
Verifying, please re-enter new_password:
```

PRIVILEGE REQUIRED

An individual user may change his or her own password, although use of the **kpasswd** command is recommended instead. To change another user's password or the password (server encryption key) for server processes such as "*afs*", issuer must have the `ADMIN` flag set in his or her Authentication Database entry.

MORE INFORMATION

bos addkey

kas setkey

kas statistics — display statistics from one Authentication Server process.

kas statistics [-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**]**
[-noauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas sta [-a <*admin principal to use for authentication*>**] [-p** <*admin password*>**]**
[-c <*cell name*>**] [-s** <*explicit list of authentication servers*>$^{+}$**] [-n] [-h]**

DESCRIPTION

Displays statistics from one of the cell's database server machines. It is recommended that the issuer use **-servers** to specify one machine name. Otherwise, the command interpreter displays statistics for a machine chosen at random from all the database server machines with which it has established connections.

WARNING

If this command is issued in interactive mode, it is not possible to specify a file server machine because the **-servers** argument is inoperative. If the issuer is interested in a particular file server machine, it is best to leave interactive mode.

ARGUMENTS

-admin_username specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.

-password_for_admin
specifies the issuer's password. Note that provided here the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.

-cell specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.

-servers specifies the database server machine(s) with which to establish a connection. It is recommended that the issuer use this argument and specify a single machine.

-noauth establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

The information includes:

- Server's network address number and last startup date.
- Number of requests and aborted requests for various services: authentication, ticket granting, password setting, entry listing, etc.
- Number of entries with `ADMIN` flag.

EXAMPLES

The following show the output when someone authenticated as the privileged user *admin* issues this command about *fs1.transarc.com.*

% kas stat -s fs1.transarc.com

```
56 allocs, 46 frees, 0 password changes
Hash table utilization = 0.100000%
From host c037cf05 started at Tue Mar 20 12:42:02 1990:
  of 88 requests for Authenticate, 18 were aborted.
  of 14 requests for GetTicket, 0 were aborted.
  of 4 requests for CreateUser, 1 were aborted.
  of 12 requests for SetFields, 4 were aborted.
  of 3 requests for DeleteUser, 0 were aborted.
  of 23 requests for GetEntry, 4 were aborted.
  of 18 requests for ListEntry, 0 were aborted.
  of 2 requests for GetStats, 1 were aborted.
  of 2 requests for GetRandomKey, 0 were aborted.
Used 6.015 seconds of CPU time.
1 admin accounts
```

PRIVILEGE REQUIRED

Issuer must have the `ADMIN` flag set in his or her Authentication Database entry.

MORE INFORMATION

kas debuginfo

kas stringtokey — convert a character string into an octal string.

kas stringtokey -string <*password string*> **[-cell** <*cell name*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

kas str -s <*password string*> **[-c** <*cell name*>**] [-h]**

DESCRIPTION

Converts the character string *password string* into an octal string suitable for use as an encryption key. This is useful for showing what form a given password would take in the cell.

The algorithm that the Authentication Server uses to generate keys combines *password string* with the cell's name during the generation, so use of *cell name* allows the issuer to see what a string would look like when scrambled into a key in a certain cell.

ARGUMENTS

-string	specifies the password-like character string to be converted into a key.
-cell	specifies the cell name to be used in converting *password string* into a key. If the issuer does not provide it, the default is the cell in which the command is issued. Provide a complete Internet-style name.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

OUTPUT

The output is of the form:

`Converting` *password string* `in realm '`*cell name* `yields key='`*key*`'.`

EXAMPLES

The following shows a user in the Transarc Corporation cell deriving the octal key equivalent of the string "*new_pswd*".

% kas str new_pswd

```
Converting new_pswd in realm 'TRANSARC.COM' yields
    key='\255\364b\354\221s\235\313'.
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

kas getrandomkey

kas setkey

kas unlock — unlock locked user account after limits on consecutive failed authentication attempts exceeded.

kas unlock -name <*authentication ID*>
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**]**
[-noauth] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

kas u -na <*authentication ID*> **[-a** <*admin principal to use for authentication*>**]**
[-p <*admin password*>**] [-c** <*cell name*>**]**
[-s <*explicit list of authentication servers*>+**] [-no] [-h]**

DESCRIPTION

Unlocks the account specified with **-name**; the account is locked when the maximum number of consecutive authentication failures is exceeded using either the AFS **login** or **klog** commands. The maximum number of successive authentication failures, as well as the amount of time that the account will be locked if that number is exceed are set using the **kas setfields** command and displayed using the **kas examine** command.

WARNING

If this command is issued in interactive mode, it is not possible to specify a file server machine because the **-servers** argument is inoperative. If the issuer is interested in a particular file server machine, it is best to leave interactive mode.

ARGUMENTS

-name specifies the user name associated with the account that you want to unlock.

-admin_username specifies the user name under which the issuer wishes to perform the command. If the issuer does not provide it, the current identity is used. See section 4.4 for more details.

-password_for_admin
specifies the issuer's password. Note that provided here the password is visible on the screen. If the issuer does not provide it, it will be prompted for and not be visible on the screen. See section 4.4 for more details.

-cell specifies the cell in which to run the command, if not the local cell. See section 4.4 for more details.

-servers specifies the database server machine(s) with which to establish a connection. See section 4.4 for more details.

-noauth establishes an unauthenticated connection between the Authentication Servers and issuer, whom they assign the unprivileged identity *anonymous* rather than attempting mutual authentication. See section 4.4 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 4.4 for more details.

EXAMPLES

In the following example, the system administrator *admin* is unlocking *erz*'s user account:

```
% kas u erz admin
Administrator's (admin) Password:
```

PRIVILEGE REQUIRED

Issuer must have the `ADMIN` flag set in his or her Authentication Database entry.

MORE INFORMATION

kas examine

kas setfields

login

klog

5. The package Command and Configuration File Lines

This chapter defines the **package** command that system administrators can use to configure the local disk on client machines automatically. It assumes the reader is familiar with the concepts described in the *AFS System Administrator's Guide*.

Running **package** involves several preparatory stages:

1. Preparing a ***prototype file*** for each desired client machine configuration, defining the files, directories, symbolic links and devices that should appear on the machine's local disk. Since many configurations are appropriate for a range of machine types, prototype files often include variables which make them machine-independent.
2. Altering a Transarc-supplied Makefile to make it appropriate for the local cell's needs.
3. "Compiling" the prototype file(s) by replacing the variables with values appropriate for a given machine type, and installing the resulting machine-specific ***configuration files*** into the appropriate directory.
4. Placing a *.package* file in the root (*/*) directory of each client's local disk, to define which configuration file to use in arranging the local disk.
5. Copying the **package** binary into the */etc/package* file on each client machine, or making */etc/package* a symbolic link to the correct binary in AFS.
6. Altering each client machine's */etc/rc* to include a call to **package**. This enables the administrator to reconfigure a client machine's local disk simply by altering the *.package* and/or configuration files appropriately and rebooting the machine.

The *AFS System Administrator's Guide* details all the necessary procedures. This chapter defines the syntax of the **package** command and of the lines that may appear in a prototype file.

In addition to running **package** by rebooting, it is possible to invoke it directly from the command shell on the client machine whose disk is to be configured. It is not possible to invoke **package** remotely.

package — configure local disk.

package [initcmd] [-config <*base name of configuration file*>**]**
[-fullconfig <*full name of configuration file, or stdin for standard input*>**]**
[-overwrite] [-noaction] [-verbose] [-silent] [-rebootfiles]
[-debug] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

package [initcmd] [-c <*base name of configuration file*>**]**
[-f <*full name of configuration file, or stdin for standard input*>**]**
[-o] [-n] [-v] [-s] [-r] [-d] [-h]

DESCRIPTION

Configures the machine's local disk to comply with the indicated configuration file. By default, **package** does nothing to elements on disk that have no counterpart in the configuration file, but the issuer can have **package** remove any element found in a directory on disk that is not also found in that directory in the configuration file. See the "R" update code in the definition of the "D" line.

If an element named in the configuration file already exists on disk but is not identical to the configuration file element, then **package** updates the existing disk version. For example, if an element exists on disk as a symbolic link but is defined as a directory or file in the configuration file, then **package** will replace the symbolic link with a directory or file. The same holds for directory or file elements on disk that appear in the configuration file as a symbolic link. Similarly, if the contents of a file do not match those in the "source" file referred to in the configuration file, then **package** overwrites the existing file with the contents of the source file. The issuer can specify different updating behavior through the use of "update codes" on the lines in the configuration file. See the definition of the "F" line.

ARGUMENTS

initcmd accommodates the command's use of the AFS command parser. It can be omitted and ignored.

-config specifies the pathname of the configuration file to use, ending in the file's "base name", which omits the suffix that indicates the machine type. **Package** can determine a machine's type and automatically select the appropriate version of the base file name. An example of the proper value for this argument is *staff* rather than *staff.rt_aix221*.

If only a file name is provided, **package** looks for it in the current working directory. It also interprets partial pathnames relative to that directory.

Provide this argument or **-fullconfig**.

-fullconfig specifies the configuration file to use. It accepts two types of values:

- The full pathname of the configuration file to use, complete with an extension indicating the machine-type (examples: *staff.rt_aos4*, *staff.dkload.pmax_ul43*).

- The string **stdin**, which indicates that the issuer will provide configuration information via standard input (stdin), either as a piped file or by typing the configuration file at the keyboard. Type **^D** (Control-D) to conclude the input.

Provide this argument or **-config**.

-overwrite tells **package** to overwrite elements on the local disk with the source version indicated in the configuration file, even if the disk version's "user" `w` mode bit is turned off. Files protected by the **I** update code are not overwritten; see the "F" line definition.

-noaction indicates that **package** should not actually perform the command, but instead should print a list (at standard out) of any problems it would encounter in executing the command. If the **-verbose** flag is added, **package** produces a trace of all the actions it would perform in executing the command, but does not actually perform them.

-silent suppresses some of the trace messages that **package** produces at standard out by default. It still reports on major problems encountered.

-verbose indicates that **package** should produce a more detailed trace of its actions (at standard out). The trace records on the transfer and ownership/mode bit setting of each element in the configuration file.

-rebootfiles indicates that **package** should not overwrite any element marked with the `Q` update-mode code in the configuration file. This effectively prevents the machine from rebooting automatically again when **package** is invoked from */etc/rc*.

-debug enables debugging output, which is directed to standard out (stdout) unless the issuer directs it into a file. By default, no debugging output is produced.

-help prints the online help for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

This command is usually invoked in an initialization file such as */etc/rc*, rather than typed at the command shell prompt.

The following invokes the version of the *staff* configuration file appropriate for this machine's system type, and produces verbose output.

```
/etc/package -c staff -v
```

The following tells **package** to use the configuration file whose basename is stored in */.package* on the local machine. This can be convenient because the administrator can put it (the same command) in every machine's */etc/rc* file and still configure machines differently, by putting the appropriate basename in */.package*.

```
/etc/package -c ‘cat /.package‘ -v
```

PRIVILEGE REQUIRED

Issuer must be logged into the machine's UNIX file system as the super user "root."

package apropos — show each help entry containing keyword.

package apropos [-topic <*help string*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

package ap [-t <*help string*>**] [-h]**

DESCRIPTION

Displays the first line of the online help entry for any **package** command that has *help string* in its name or short description.

ARGUMENTS

-topic specifies the keyword string to search for. If it is more than a single word, surround it with double quotes or other delimiters. Type all *help string*s for **package** commands in lowercase letters.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one.

OUTPUT

The first line of a command's online help entry names the command and briefly describes what it does. The **package apropos** command displays the first line for any **package** command where *help string* is part of the command name or first line.

To see the remaining lines in a help entry, which provide the command's alias (if any) and syntax, use the **package help** command.

EXAMPLES

The following lists all **package** commands that have the word "*help*" in their operation code or short online description.

```
% package a help
apropos: search by help text
help: get help on commands
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

package help **package_test**

package help — show syntax of specified **package** command(s) or list functional description for all of them.

package help [-topic <*help string*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

package h [-t <*help string*>+**] [-h]**

DESCRIPTION

Displays the first line (name and short description) of every **package** command's online help entry, if no *help string* is provided. For each operation code specified with **-topic**, it outputs the entire help entry. See the OUTPUT section below.

ARGUMENTS

-topic	specifies the operation code(s) for which the syntax is to be displayed. If it is omitted, the output instead provides a short description of all **package** commands.
-help	prints the online help entry for this command. Do not provide any other arguments or flags with this one.

OUTPUT

The online help entry for each **package** command consists of two or three lines.

- The first line names the command and briefly describes what it does.
- If the command has aliases, they will appear on the next line.
- The final line, which begins with "`Usage:`", lists the command's arguments and flags in the prescribed order. Online help entries use the same symbols (brackets, etc.) as the command definitions in this manual. For an explanation of their meaning, see page x of the introductory *About This Manual* chapter.

EXAMPLES

The following displays the online help entry for the **package initcmd** command.

```
% package help initcmd
package initcmd: initialize the program
Usage: package [initcmd] [-config <base name of configuration file>]
[-fullconfig <full name of configuration file, or stdin for standard
input>] [-overwrite] [-noaction] [-verbose] [-silent] [-rebootfiles]
[-debug] [-help]
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

package apropos

Configuration File "B" Line — define block special device.

B *<device name> <major device number> <minor device number> <owner name> <group name> <mode bits>*

DESCRIPTION

Defines a block special device, such as a disk, that deals with input in units of multi-byte blocks.

ARGUMENTS

B	should be a capital letter and tells the command interpreter that this line defines a block special device.
device name	names the block special device being defined.
major device number	specifies the device's major device number. To learn the correct value, consult documentation provided by the machine and/or operating system's manufacturer.
minor device number	specifies the device's minor device number. To learn the correct value, consult documentation provided by the machine/operating system's manufacturer.
owner name	names the owner for the device in the UNIX file system (the user who will appear in the device's "owner" field in the output from **ls -l**).
group name	names the group for the device in the UNIX file system (the group that will appear in the device's group field in the output from **ls -lg**).
mode bits	defines the mode bits that protect the device in the UNIX file system. Legal values are the standard three- or four-digit decimal numbers corresponding to combinations of rights. Example: *755* corresponds to `rwxr-xr-x`.

EXAMPLES

The following defines disk */dev/hd0a* with major and minor device numbers *1* and *0*.

```
B /dev/hd0a 1 0 root wheel 644
```

Configuration File "C" Line — define character special device.

C *<device name> <major device number> <minor device number> <owner name> <group name> <mode bits>*

DESCRIPTION

Defines a character special device, such as a terminal or tty, that deals with input in single character units.

ARGUMENTS

C	should be a capital letter and tells the command interpreter that this line defines a character special device.
device name	names the character special device being defined.
major device number	specifies the device's major device number. To learn the correct value, consult documentation provided by the machine/operating system's manufacturer.
minor device number	specifies the device's minor device number. To learn the correct value, consult documentation provided by the machine/operating system's manufacturer.
owner name	names the owner for the device in the UNIX file (the user who will appear in the device's "owner" field in the output from **ls -l**).
group name	names the group for the device in the UNIX file system (the group that will appear in the device's group field in the output from **ls -lg**).
mode bits	defines the mode bits that protect the device in the UNIX file system. Legal values are the standard three- or four-digit decimal numbers corresponding to combinations of rights. Example: *644* corresponds to `rw-r--r--`.

EXAMPLES

The following defines tty */dev/ttyp5* with major and minor device numbers *6* and *5*.

```
C /dev/ttyp5 6 5 root wheel 666
```

Configuration File "D" Line — define directory.

D[<*update code*>**]** <*directory*> <*owner name*> <*group name*> <*mode bits*>

DESCRIPTION

Names a directory to be created on the local disk. If a directory of this name already exists, **package** does nothing. However, if an element of the same name exists on disk as a symbolic link, **package** replaces the symbolic link with an actual directory.

ARGUMENTS

D	should be a capital letter and tells the command interpreter that this line creates a directory.
update code	is optional and if provided should follow the letter **D** directly, without an intervening space. Choose one of the two legal values: • **X**, which indicates that the directory is a *lost+found* directory (used by the **fsck** program). • **R**, which indicates that **package** should remove any files from the local disk directory that do not appear in the configuration file. It will also remove any subdirectories of the directory, and files in them, that do not appear in the configuration file.
directory	specifies the full pathname of the directory to be created.
owner name	names the owner for the directory in the UNIX file system (the user who will appear in the directory's "owner" field in the output from **ls -l**).
group name	names the group for the directory in the UNIX file system (the group that will appear in the directory's group field in the output from **ls -lg**).
mode bits	defines the mode bits that protect the directory in the UNIX file system. Legal values are the standard three- or four-digit decimal numbers corresponding to combinations of rights. Example: *755* corresponds to `rwxr-xr-x`.

EXAMPLES

The following defines the */usr* directory.

```
DR /usr root wheel 755
```

Configuration File "F" Line — create/update file.

F[<*update code*>**]** <*file*> <*source file*> **[**<*owner name*> <*group name*> <*mode bits*>**]**

DESCRIPTION

Names a file to be created on the local disk. Its contents come from the indicated source file, which may reside either in AFS or on the local disk itself. If **package** cannot locate the source file, it aborts.

If a file of this name already exists on disk, then its contents are updated with (overwritten by) the contents of the source file, unless the **I** update code is used to prevent that. Other update codes also modify the way in which the update takes place; see the ARGUMENTS section.

If a symbolic link or directory exists on disk under this name, **package** replaces it with the indicated file.

ARGUMENTS

F should be a capital letter and tells the command interpreter that this line creates/updates a file.

update code determines the manner in which the source file is used to update the contents of the file, if the latter already exists on the local disk. This argument is optional and if provided should follow the letter **F** directly, without an intervening space. Any number of the update-mode codes may combined with one another, in any order. The legal values are:

- **A**, which indicates that the pathname in the *source file* slot is the complete pathname of the source file, including the filename. By default, the source file pathname does not include a file name at the end—instead, **package** appends the name in the *file* slot to the *source file* pathname in order to determine the source file's full name. Thus, this code allows the name of the file being created to differ from the name of the source file. See also the EXAMPLE(S) section.
- **I**, which indicates that the file should not be overwritten if it already exists on the local disk. If it does not exist, then **package** creates it, and copies the contents of the source file into it.
- **O**, which indicates that **package** should save the existing local disk version of the file with a *.old* extension, before overwriting its contents with the source file's.

- **Q**, which indicates that if **package** overwrites the file, then it should exit with status code 4. If the standard *package*-related changes have been made to */etc/rc*, then status code 4 will cause an automatic reboot of the machine. This is useful in cases where the machine must reboot in order for updates to the file to have any effect (*/vmunix* is a common example).

file specifies the complete pathname of the file to be created or updated on the local disk.

source file specifies the file (in AFS or on the local disk) whose contents should be used to create or update the disk file.

If the **A** update code appears on the line, this slot should specify the *source file*'s complete pathname. Otherwise, this slot should specify only the part of the pathname that precedes the name of the local disk *file*. As an example, suppose that the configuration file is for a Transarc cell machine running AIX 2.2.1, so that the source for the local disk file called */bin/grep* is */afs/transarc.com/rt_aix221/bin/grep*. If the **A** does not appear, then the proper filler for the *source file* slot is **/afs/transarc.com/rt_aix221**. See also the EXAMPLES section.

owner name names the owner for the file in the UNIX file system (the user who will appear in the file's "owner" field in the output from **ls -l**).

This slot may be left empty, in which case the file's owner is set to match the source file's owner. If this slot is empty, the *group name* and *mode bits* slots must also be empty.

group name names the group for the file in the UNIX file system (the group that will appear in the file's group field in the output from **ls -lg**).

This slot may be left empty, in which case the file's group is set to match the source file's group. If this slot is empty, the *owner name* and *mode bits* slots must also be empty.

mode bits defines the mode bits that protect the file in the UNIX file system. Legal values are the standard three- or four-digit decimal numbers corresponding to combinations of rights. Example: *755* corresponds to `rwxr-xr-x`.

This slot may be left empty, in which case the file's mode bits are set to match the source file's mode bits. If this slot is empty, the *owner name* and *group name* slots must also be empty.

EXAMPLES

The following, appropriate for a machine running AIX 2.2.1 in the Transarc cell, creates/updates the file */bin/grep* on the local disk, using */afs/transarc.com/rt_aix221/bin/grep* as the source.

```
F /bin/grep /afs/transarc.com/rt_aix221 root wheel 755
```

The next example specifies an absolute pathname for the source file, as indicated by the **A** update code. The **Q** code makes **package** return status code 4 as it exits, prompting a reboot of the machine if the standard **package**-related changes have been made to */etc/rc*. This example also leaves the *owner name*, *group name* and *mode bits* slots empty, so that the disk file adopts the source file's values for those slots.

```
FAQ /usr/vice/etc/ThisCell /afs/transarc.com/common/etc/ThisCell
```

Configuration File "L" Line — create symbolic link.

L[<*update code*>**]** <*link*> <*actual file*> **[**<*owner name*> <*group name*> <*mode bits*>**]**

DESCRIPTION

Creates a symbolic link to a directory or file that exists either in AFS or elsewhere on the local disk. If package cannot access the actual directory/file, it aborts.

If a file or directory currently exists on the local disk under this name, **package** replaces it with a symbolic link.

ARGUMENTS

L should be a capital letter and tells the command interpreter that this is a symbolic link definition.

update code modulates the way that **package** creates the symbolic link, as discussed for each code. It is optional and if provided should follow the letter **L** directly, without an intervening space. The two legal update-mode codes may combined with one another, in either order. The legal values are:

- **A**, which indicates that the pathname in the *actual file* slot is the complete pathname of the actual file, including the filename. By default, the actual file pathname does not include a file name at the end—instead, **package** appends the name in the *link* slot to the *actual file* pathname in order to determine the actual file's full name. Thus, this code allows the name of the symbolic link being created to differ from the name of the actual file. See also the EXAMPLE(S) section.
- **I**, which indicates that the symbolic link should not be overwritten if it already exists on the local disk. If it does not exist, then **package** creates it.

link specifies the complete pathname of the symbolic link to be created on the local disk.

actual file specifies the file (in AFS or on the local disk) to which the symbolic link refers.

If the **A** update code appears on the line, this slot should specify the *actual file*'s complete pathname. Otherwise, this slot should specify only the part of the pathname that precedes the name of the local disk *link*. As an example, suppose that the configuration file is for a Transarc cell machine running Ultrix 3.0, so that */etc/ftpd* is a symbolic link to */afs/transarc.com/vax_ul3/etc/ftpd*. If the **A** does not appear, then the proper filler for this slot is **/afs/transarc.com/vax_ul3**, not **/afs/transarc.com/vax_ul3/etc/ftpd**. See also the EXAMPLE(S) section.

owner name names the owner for the symbolic link in the UNIX file system (the user who will appear in the symbolic link's "owner" field in the output from **ls -l**).

This slot may be left empty, in which case the symbolic link's owner is set to match the UNIX user name of the issuer of the **package** command (that is, "root"). If this slot is empty, the *group name* and *mode bits* slots must also be empty.

group name names the group for the symbolic link in the UNIX file system (the group that will appear in the symbolic link's group field in the output from **ls -lg**).

This slot may be left empty, in which case the symbolic link's group is set to match the group for the issuer of the **package** command (defined in the issuer's */etc/rc* entry. If this slot is empty, the *owner name* and *mode bits* slots must also be empty.

mode bits defines the mode bits that protect the symbolic link in the UNIX file system. Legal values are the standard three- or four-digit decimal numbers corresponding to combinations of rights. Example: *755* corresponds to `rwxr-xr-x`.

This slot may be left empty, in which case the symbolic link's mode bits are set to *777* (`rwxrwxrwx`). If this slot is empty, the *owner name* and *group name* slots must also be empty.

EXAMPLES

The following, appropriate for a machine running Ultrix 3.0 in the Transarc cell, creates a symbolic link from */etc/ftpd* on the local disk to the file */afs/transarc.com/vax_ul3/etc/ftpd* in AFS.

```
L /etc/ftpd /afs/transarc.com/vax_ul3 root wheel 644
```

The next example uses an absolute pathname (as indicated by the **A** update code), because the symbolic link and actual file names differ. This example also leaves the *owner name*, *group name* and *mode bits* slots empty, so that the symbolic link adopts values for those slots from the actual file.

```
LA /etc/printcap /afs/transarc.com/common/etc/printcap.remote
```

Configuration File "S" Line — define socket.

S <*socket name*> **[**<*owner name*> <*group name*> <*mode bits*>**]**

DESCRIPTION

Defines a socket (a communications device for UDP and TCP/IP connections).

ARGUMENTS

S	should be a capital letter and tells the command interpreter that this is a socket definition.
socket name	names the socket being defined.
owner name	names the owner for the socket in the UNIX file system (the user who will appear in the socket's "owner" field in the output from **ls -l**). This slot may be left empty, in which case the socket's owner is set to match the UNIX user name of the issuer of the **package** command (that is, "root"). If this slot is empty, the *group name* and *mode bits* slots must also be empty.
group name	names the group for the socket in the UNIX file system (the group that will appear in the socket's group field in the output from **ls -lg**). This slot may be left empty, in which case the socket's group is set to match the group for the issuer of the **package** command (defined in the issuer's */etc/rc* entry. If this slot is empty, the *owner name* and *mode bits* slots must also be empty.
mode bits	defines the mode bits that protect the socket in the UNIX file system. Legal values are the standard three- or four-digit decimal numbers corresponding to combinations of rights. Example: *755* corresponds to `rwxr-xr-x`. This slot may be left empty, in which case the socket's mode bits are set to *777* (`rwxrwxrwx`), as modulated by the cell's umask. If this slot is empty, the *owner name* and *group name* slots must also be empty.

EXAMPLES

The following defines the socket */dev/printer*:

```
S /dev/printer root wheel 777
```

Configuration File %define — define variable or declare string.

%define <*variable*> <*value*>
%define <*declaration*>

DESCRIPTION

If followed by two arguments, defines the second argument as the value of the first, which may appear as a variable (in the form `${variable}`) in the same or other prototype/library files. The second argument is substituted for occurrences of the variable wherever it occurs.

If followed by a single argument, declares the argument to be defined, making it available for use as a controller in **%ifdef** and **%ifndef** statements. That is, **%ifdef** statements mentioning this variable will resolve as "true," **%ifndef** statements as "false."

ARGUMENTS

variable	as the first of two arguments, specifies the name of the variable for which a value is defined. This argument may appear in variable form (that is, as `${variable}`) in this or other prototype/library files.
value	as the second of two arguments, specifies the value to be assigned to variable-form occurrences of the first argument.
declaration	as the single argument, declares the indicated string to be defined.

EXAMPLES

The following defines the value for the variable `${diskmode}`. This variable is used elsewhere in the template to fill the <*owner name*>, <*group name*> and <*mode bit*> slots on a "D", "F" or "L" line.

```
%define diskmode root wheel 644
```

The following declares the string "`afsd`" to be defined.

```
%define afsd
```

Configuration File %ifdef — specify action to perform if value is declared or defined.

%ifdef <*declaration*>
<*action*>+
[%else [<*declaration*>**]**
<*alternate action*>+**]**
%endif <*declaration*>

DESCRIPTION

Specifies the action(s) to perform if the indicated declaration has been made with a single-argument **%define** statement, or if the indicated variable has a value.

The optional **%else** statement specifies alternate action(s) to perform if the indicated declaration has never been made, or has been undone with **%undef**.

It is possible to nest any number of **%ifdef** and **%ifndef** statements.

ARGUMENTS

declaration	specifies the string that must be declared, or variable that must be defined, if the *action* is to be performed. Its second occurrence (following **%else**) is optional, serving only to clarify which **%ifdef** the **%else** belongs to. The first and third occurrences (the latter following **%endif**) are required.
action	specifies each action to perform if the **%ifdef** statement resolves to "true." Each action should appear on a separate line. Both "**%**-statements" and definition lines are acceptable actions.
alternate action	specifies each action to perform if the **%ifdef** statement resolves to "false." Each action should appear on a separate line. Both "**%**-statements" and definition lines are acceptable actions.

EXAMPLES

The following specifies that if the string `rt_aos4` is currently declared, then the three indicated library files will be included in the configuration file resulting from compilation of this prototype file.

```
%ifdef rt_aos4
%include ${wsadmin}/lib/rt_aos4.readonly
%include ${wsadmin}/lib/rt_aos4.generic
%include ${wsadmin}/lib/rt_aos4.generic.dev
%endif rt_aos4
```

Configuration File %ifndef — specify action to perform if value is not declared or defined.

%ifndef <*declaration*>
<*action*>+
[%else [<*declaration*>**]**
<*alternate action*>+**]**
%endif <*declaration*>

DESCRIPTION

Specifies the action(s) to perform if the indicated declaration has never been made or has been undone with an **%undef** statement, or if the indicated variable does not have a value.

The optional **%else** statement specifies alternate action(s) to perform if the indicated declaration/definition has been made, or if the variable does have a value.

It is possible to nest any number of **%ifdef** and **%ifndef** statements.

ARGUMENTS

declaration	specifies which string must not be declared, or which variable must not be defined, in order for *action* to be performed. Its second occurrence (following **%else**) is optional, serving only to clarify which **%ifdef** the **%else** belongs to. The first and third occurrences (the latter following **%endif**) are required.
action	specifies each action to perform if the **%ifndef** statement resolves to "true" (that is, the string is not declared or variable is not defined). Each action should appear on a separate line. Both "**%**-statements" and definition lines are acceptable actions.
alternate action	specifies each action to perform if the **%ifndef** statement resolves to "false" (that is, the string is declared or variable does have a defined value). Each action should appear on a separate line. Both "**%**-statements" and definition lines are acceptable actions.

EXAMPLES

The following, appropriate for the Transarc Corporation cell, defines "`transarc.com`" as the value of the `${cell}` variable if it does not already have a value.

```
%ifndef cell
%define cell transarc.com
%endif cell
```

Configuration File %include — include library file.

%include <*file pathname*>

DESCRIPTION

Causes contents of indicated library file to be included in configuration file resulting from compilation of this prototype file.

ARGUMENTS

file pathname	specifies the complete pathname of the library file to be included. The pathname may include variables.

EXAMPLES

The following includes the file *base.generic* from the *lib* subdirectory of the directory in which the cell keeps **package** related files. The `${wsadmin}` variable will be resolved to an actual pathname (such as */afs/transarc.com/wsadmin*) during compilation.

```
%include ${wsadmin}/lib/base.generic
```

Configuration File %undef — declare value to be defined no longer.

%undef <*declaration*>

DESCRIPTION

Specifies that the indicated string is no longer defined. Any **%ifndef** statements involving this string will then resolve as "true," **%ifdef** statements as "false."

ARGUMENTS

declaration specifies the string that should no longer is no longer defined.

EXAMPLES

The following declares the string "`afsd`" not to be defined.

```
%undef afsd
```

6. The pts Commands

This chapter defines the **pts** commands that system administrators and regular users use to contact the Protection Server. It assumes the reader is familiar with the concepts described in the *AFS System Administrator's Guide*.

AFS augments and refines the standard UNIX scheme for controlling access to files and directories. Instead of relying only on the mode bits that define access rights for individual files, AFS associates an ***access control list (ACL)*** with each directory. The ACL lists the user and group access rights to the files in the associated directory. Each ACL applies to all of the files in its associated directory. Using ACLs allows AFS to define seven rights rather than just the standard three. (It is still possible to set the mode bits, but AFS interprets them in a different way; see the chapter on protection in the *AFS System Administrator's Guide* for details.)

Another refinement to the standard UNIX protection scheme is that users can define their own protection ***groups*** and then place the groups on ACLs as though they were individual users. This makes it easy to assign the same rights to many people simultaneously. Groups make it possible to add someone to many ACLS by adding them to a group that already exists on those ACLs (and it is just as easy to remove someone the same way). Machines can also be members of a group. When logged into a machine that is a member of a group, a user is given all of the access rights of that group.

The Protection Server maintains a Protection Database that stores entries for groups, individual users, and machines, and charts the membership relationships between them. Section 6.1 details the structure of database entries.

The **pts** command interface allows system administrators and regular system users to create, modify, and delete entries in the Protection Database by interacting with the Protection Server. Members of *system:administrators* can manipulate any user, machine, or group entry. Regular users may manipulate the group entries they own or for which they have the necessary privilege. Consult each command description to learn about necessary privileges.

Refer to the *Command Summary* at the end of this document for a complete list of **pts** commands and their syntax.

6.1. The Information in Protection Database Entries

Each user, machine, or group record in the Protection Database contains relevant protection data about that entry. The majority of the **pts** commands are used to access or edit the information in these records. Below is a list of the fields that make up these records:

- ***name***. For users, this is the character string typed when logging in. For machines, the name is the IP address (or range of IP addresses specified with wild cards) corresponding to the machine (or group of machines).

  For groups, there are two different styles of names: ***regular*** and ***prefix-less***. Regular group names have two parts, separated by a colon. The first part, the *owner-name* field, specifies the owner of the group. The second part, the *group-name* field, is the name of the group, as chosen by the creator. In a more graphic form:

  owner-name:group-name

  Prefix-less group do not have the *owner-name* field or the colon. Only members of the *system:administrators* group are allowed to create prefix-less groups.

- ***AFS UID***. This is a unique identification number that the AFS server processes use internally. It is similar in function to a UNIX UID, but operates in the AFS file system rather than the UNIX file system. Users and machines have positive integer AFS UIDs, and groups have negative integer AFS UIDs.

  Normally, the Protection Server assigns AFS UIDs automatically when creating entries. As a member of the *system:administrators* group, you can specify the AFS UID for a user, machine, or group entry as you create it, and can manipulate the counters used in automatic allocation. See the **pts listmax** and **pts setmax** commands for more information about the AFS UID counters.

- ***owner***. This is the user or group that owns the entry (that is, is allowed to make changes to various aspects of an entry like its ownership, name, list of groups/members, etc.,) or delete it entirely. The owner cannot necessarily change the group-creation quota, unless he or she is a member of *system:administrators*.

  As a member of the *system:administrators* group, you have total administrative control over every entry in the Protection Database, even if you are not the owner. See the discussion of the **pts setfields** command for more information.

- ***creator***. The name of the user who originally created the entry using the **pts createuser** and **pts creategroup** command. This field is mainly useful as an audit trail, and cannot be changed.

- ***membership count***. For users and machines, this indicates how many groups the user/machine belongs to. For groups, it indicates how many members belong to the group. This number cannot be set explicitly.

- ***privacy flags***. This field indicates who is allowed to list certain information about the entry or change it in certain ways. See the discussion of the **pts examine** and **pts setfields** commands for more information.

- ***group-creation quota***. This field indicates how many groups a user is allowed to create. It is set to `20` at the time each user or machine entry is created. The creation quota for machines is meaningless because it is not possible to authenticate as a machine. The quota is also meaningless for groups and should have the value `0`.

Only members of *system:administrators* may reset a group-creation quota using the **pts setfields** command. The most common reason is to grant additional quota to a user who has exhausted the original quota of 20, but you may also use it to set a lower quota.

- ***membership list***. For users and machines, this lists all the groups they belong to. The system-defined groups *system:anyuser* and *system:authuser* do not appear.

  For groups, this lists all the members of the group. Groups may not be members of other groups. This information is not available for the system-defined groups *system:anyuser* and *system:authuser*.

- ***groups owned***. This records all the group entries that this user or group owns. A group may own another group. A machine may theoretically own a group, but this provides no additional functionality.

6.2. The Standard System Groups

In addition to the groups that users and administrators can create, AFS defines three standard groups:

- ***system:anyuser***. This group is unusual in that it has no stable membership. Instead, its membership changes as people attempt to access files in the cell's file tree. It includes everyone able to access the local cell, authenticated or not, including people who have logged into a local machine as "root," or AFS users from foreign cells who have telnetted to a local machine.

- ***system:authuser***. This group also lacks a stable membership—its membership changes as people authenticate and unauthenticate in the local cell. It includes everyone who is currently authenticated (has a valid token) in the cell. This means that they have proved their identities to the Authentication Server by providing the correct passwords, either during login or by issuing the **klog** command at another time.

- ***system:administrators***. This group includes the few users authorized to administer the cell. Members of this group have implicit ADMINISTER rights on the ACL of every directory in the system. Only members of *system:administrators* can issue privileged **pts** commands. Because of the extensive power granted by membership in this group, it is recommended that only a few high-level administrators be included in this group.

When the Protection Server initializes the Protection Database for the first time in a cell, it automatically creates entries for these groups in the Protection Database and allocates them AFS UIDs. (The IDs assigned are standard and should not be altered, although they may be examined with **pts examine**.) The counter that the Protection Server uses when allocating AFS UIDs is set beyond these standard IDs so that there will be no conflict when other groups are created later.

In addition to having no stable membership, the *system:anyuser* and *system:authuser* groups are unique in other ways. It is not possible to use **pts adduser** to add members to these groups, nor can their membership be listed with **pts membership**. Like other groups, these two can appear on ACLs, which provides a convenient way to allow all users (or all authenticated users) certain rights on a directory. As each AFS volume is created, the ACL on its root directory is set to grant READ and LOOKUP rights to *system:anyuser*.

The *system:administrators* group does have a stable membership, which only current members of it may alter. The person who installs a cell's first file server machine should add himself or herself to this group or create a "dummy" administrative account while the Protection Server is still running in "no authorization checking" mode. See the *AFS Installation Guide* for details. It is possible to place *system:administrators* on ACLs, but that is generally not necessary because members of the group have implicit ADMINISTER rights on every directory.

6.3. Common Arguments and Flags

All **pts** commands accept optional arguments and flags. They are listed in the command descriptions and are described here in detail:

[-cell <*cell name*>]

This argument specifies that the command should be run in the cell specified by *cell name*. The issuer may abbreviate *cell name* to the shortest form that distinguishes it from the other cells listed in */usr/vice/etc/CellServDB* on the client machine on which the command is issued. By default, commands are executed in the local cell, as defined:

- First, by the value of the environment variable AFSCELL.
- Second, in */usr/vice/etc/ThisCell* on the client machine on which the command is issued.

[-noauth]

This flag instructs the Protection Server not to authenticate the user of the command, and thus establishes an unauthenticated connection between the user and the Protection Server (the user is recognized as the unprivileged user *anonymous*). It is useful only when authorization checking is disabled on the file server machine (during the installation of a file server machine or when **bos setauth** has been used during other unusual circumstances). In normal circumstances, the Protection Server allows only authorized (privileged) users to issue commands that change the status of a server or configuration file, and will refuse to perform such an action even if the **-noauth** flag is used.

[-test]

This flag directs the **pts** command interpreter to refer to the *CellServDB* file in */usr/afs/etc*, not */usr/vice/etc*, where it normally would. This is useful when a developer is testing the Protection Server by running an isolated instance on his or her own workstation. By including only that workstation (which is not a database server machine for the cell) in */usr/afs/etc/CellServDB* and then using this flag, the developer makes sure that the test does not affect the real Protection Server and Database in use in the remainder of the cell.

[-force]

This flag directs the **pts** command interpreter to continue executing the command, if possible, even if it encounters problems during the command's execution. The command interpreter performs as much of the requested operation as possible, rather than aborting if it encounters a problem. The command interpreter reports any errors it encounters during the command's execution. This flag is especially useful if you list many arguments with a command's switch; if one of the arguments is invalid, the command reports the error and proceeds with the remaining arguments.

[-help]

This flag has the same function as the **pts help** command: it prints the command's online help message on the screen. No other arguments or flags should be provided at the same time. Even if they are, this flag overrides them, and the only effect of issuing the command is that the help message appears.

6.4. Privileges Required for pts Commands

To issue most of the **pts** commands on a user or group Protection Database entry, the user issuing the command must have the required privileges. The exact privileges required, as well as any conditions that must be met, are listed in the PRIVILEGE REQUIRED section of each **pts** command listed in this chapter. Below is a summary of privilege requirements.

Members of the *system:administrators* group are permitted to issue privileged **pts** commands and use privileged command arguments on all user and group Protection Database entries. (Only someone who already belongs to *system:administrators* may add a user to the *system:administrators* group. See the **pts adduser** command.)

Regular users (that is, user that are not members of *system:administrators*) can always list the information associated with the group entries they own, as well as, their own user entries. Regular users are not permitted to use certain commands and command arguments reserved for members of the *system:administrators* group. Depending on how a group owner has defined the privacy flags field for his or her owned group, regular users may also be permitted to access and manipulate the Protection Database entry for that group. (The **pts setfields** command is used to define the privacy flags field.)

pts adduser — add users and machines to groups.

pts adduser -user <*user name*>+ **-group** <*group name*>+
[-cell <*cell name*>**] [-noauth] [-test] [-force] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts ad -u <*user name*>+ **-g** <*group name*>+ **[-c** <*cell name*>**] [-n] [-t] [-f] [-h]**

DESCRIPTION

Makes each specified user and machine entry a member of each specified *group name* entry. While there is theoretically no limit to the number of members in a group, the **pts membership** command can list no more than 5000 members, even if more exist.

WARNING

The effects of adding a user to a group are not immediate. Before a new user member of a group gets the access rights of that group, he or she must re-authenticate (issue **klog**). As long as a user retains old tokens, he or she retains the access rights acquired with those tokens.

ARGUMENTS

-user	specifies the user name of each user entry and/or the IP address of each machine entry to be added to each group.
-group	specifies the complete group name (including the owner field if applicable) of each group to which each indicated user and/or machine entry should be added.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following adds user *smith* to *system:administrators*.

% pts ad smith system:administrators

The following adds user *pat* to *terry:friends* and to *staff*:

% **pts ad pat terry:friends staff**

The following adds users *jones*, *terry* and *pat* to *smith:colleagues*. The switches are necessary because there is more than one instance of the first list-taking argument.

% **pts ad -u jones terry pat -g smith:colleagues**

The following adds the machine entries in the Transarc Corporation subnet to the group *bin-prot*. Because of the IP address range of the Transarc Corporation subnet, the system administrator was able to group the machines into three machine entries (using the wildcard scheme discussed under the **pts createuser** command).

% **pts ad 158.98.0.0 192.55.207.0 192.54.226.0 -gr bin-prot**

PRIVILEGE REQUIRED

Depends on how the group's fourth privacy flag is set:

- If it is a hyphen, only the group's owner and members of *system:administrators* may add members.
- If it is a lowercase a, members of the group may add other members.
- If it is an uppercase A, everyone on the system may add members.

The **pts setfields** command sets the privacy flags and the **pts examine** command displays them.

MORE INFORMATION

pts examine

pts membership

pts removeuser

pts setfields

pts apropos — show each help entry containing keyword.

pts apropos -topic <*help string*> [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

pts ap -t <*help string*> **[-h]**

Displays the first line of the help entry for any **pts** command that has *help string* in its name or short description.

ARGUMENTS

-topic	specifies the keyword string for which to search. If it is more than a single word, surround it with double quotes or other delimiters. Type *help string*s for **pts** commands in lowercase letters.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

OUTPUT

The first line of a command's online help entry names the command and briefly describes what it does. The **pts apropos** command displays that first line for any **pts** command where *help string* is part of the command name or first line.

To see the remaining lines in a help entry, which provide the command's alias (if any) and syntax, use the **pts help** command.

EXAMPLES

The following lists all **pts** commands that have the word "*create*" in their operation code or short online description.

% pts ap create

```
createGroup: create a new group
createuser: create a new user
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

pts help

pts chown — change owner of entry in Protection Database.

pts chown -name <*group name*> **-owner** <*new owner*>
[-cell <*cell name*>**] [-noauth] [-test] [-force] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts ch -na <*group name*> **-o** <*new owner*> **[-c** <*cell name*>**] [-no] [-t] [-f] [-h]**

DESCRIPTION

Changes the owner of *group name* to *new owner*, by putting the latter in the owner field of *group name*'s Database entry. The new owner may be either an individual or another group.

Note: Although a machine can theoretically own a group entry, this ownership confers no functionality and is not recommended. All discussion of this command will refer only to user and group ownership.

In the case of regular groups, this command automatically changes the owner-name field (before the colon) of the group name to match *new owner*. If *new owner* is itself a group, then only its owner-name field, not its whole name, is put before the colon in the new name. The change in the owner-name field of a regular group's name does not propagate down to any groups owned by the group.

ARGUMENTS

-name	specifies the current name of the group whose owner is to change.
-owner	specifies the name of the user or group which is to become the new owner of the entry.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following shows user *terry* transferring the ownership of her group *terry:friends* to *pat*. A side effect is that the group name changes to *pat:friends*.

% pts chown terry:friends pat

The following shows user *terry* transferring the ownership of her group *terry:friends* to the group *pat:buddies*. A side effect is that the group name changes to *pat:friends*.

pts chown terry:friends pat:buddies

PRIVILEGE REQUIRED

Issuer must be *group name*'s current owner or belong to the *system:administrators* group.

MORE INFORMATION

pts rename

pts creategroup — create (empty) group entries in Protection Database.

pts creategroup -name <*group name*>+ **[-owner** <*owner of the group*>**]**
[-id <*id (negated) for the group*>+**] [-cell** <*cell name*>**]**
[-noauth] [-test] [-force] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

pts cg -na <*group name*>+ **[-o** <*owner of the group*>**] [-i** <*id (negated) for the group*>+**]**
[-c <*cell name*>**] [-no] [-t] [-f] [-h]**

DESCRIPTION

Creates an entry in the Protection Database for each specified *group name*. The issuer of the command is recorded as the group's creator. The issuer is also recorded as the group's owner unless the **-owner** argument is provided to name a different individual or a group as the owner. By default, the Protection Server allocates the next available AFS UID (as determined by the `GroupMax` counter), but members of *system:administrators* can assign a different ID.

Two types of groups can be created: ***regular*** and ***prefix-less***; though only a member of *system:administrators* may create the latter type. The rules for naming the two types of groups are discussed in the ARGUMENTS section.

Creating a group lowers the issuer's group-creation quota by one. See the **pts setfields** or **pts examine** command for more information.

ARGUMENTS

-name specifies the name of each group. A group name may be up to 63 characters in length and contain lowercase letters, numbers, and punctuation (except the colon); uppercase letters are not allowed.

When creating regular groups, *group name* must have the following format:

owner-name:group-name

The *owner-name* field must match the true owner of the group, as follows:

- If the issuer does not include the optional **-owner** argument, *owner-name* must match his or her user name.
- If the issuer does use **-owner** to name an individual, then *owner-name* must match the owner of the group's user name.
- If the issuer uses **-owner** to name a regular group as the owner, then *owner-name* must match the *owner-name* field of the owner group. If the issuer uses **-owner** to name a prefix-less group as the owner, then *owner-name* must match the entire owner group name.

When creating prefix-less groups, the *owner-name* field is omitted, leaving only the *group-name* field; the colon is unnecessary. Only members of *system:administrators* can create prefix-less groups.

-owner specifies a user different from the issuer of the command or a group (regular or prefix-less) that is to own each specified group. If the specified owner is to be a group, that group must already have at least one member; this requirement prevents the creator from assigning self-ownership to a group during its creation.

-cell specifies the cell in which to run the command. See section 6.3 for more details.

-noauth tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.

-test tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.

-force directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following shows *pat* creating the groups *pat:friends* and *pat:colleagues*. Using the **-name** switch is required because more than one instance is provided but this argument is not the last list-taking one.

```
% pts cg -name pat:friends pat:colleagues
```

The following shows *smith*, who belongs to *system:administrators*, creating the prefix-less group *staff* and assigning its ownership to *system:administrators* rather than to himself.

```
% pts cg staff system:administrators
```

The following shows *pat* creating the group *smith:team-members*, which is allowed because the owner is set to *smith* at the same time. Note that creation of the group will be charged against *pat*'s group-creation quota, not *smith*'s.

```
% pts cg smith:team-members smith
```

PRIVILEGE REQUIRED

To assign AFS UIDs with the **-id** argument or create prefix-less groups, the issuer must belong to the *system:administrators* group. Any user may create regular groups, but he or she must:

- Be authenticated. The command does not work if the **-noauth** flag is provided.
- Have a group creation quota greater than `0`. The **pts examine** command displays this quota.

MORE INFORMATION

pts examine

pts listmax

pts setfields

pts setmax

pts createuser — create user and machine entries in the Protection Database.

pts createuser -name <*user name*>+ **[-id** <*user id*>+**] [-cell** <*cell name*>**]**
[-noauth] [-test] [-force] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

pts cu -na <*user name*>+ **[-i** <*user id*>+**] [-c** <*cell name*>**] [-no] [-t] [-f] [-h]**

DESCRIPTION

Creates an entry in the Protection Database for each specified user and/or machine. The name specified for a user becomes his or her user name that must be typed at login; for a machine entry, you must specify the machine's corresponding IP address (or range of IP addresses for a grouping of machines, using wildcards).

The issuer of the command is recorded as the group's creator and *system:administrators* as its owner. The Protection Server normally allocates the next available AFS UID (as determined by the `UserMax` counter), but the issuer may assign a different ID.

ARGUMENTS

-name specifies

- The user name for each new user entry. It may contain up to 63 numbers and lowercase letters. However, it is recommended that user names be shorter than 8 characters because many UNIX programs cannot handle longer names. Also, do not include any characters considered special in UNIX, particularly the colon and period.
- The IP address (or IP address range, using wildcards) of the corresponding machine(s). Below is an explanation of how to use the wildcards in an entry for an IP address. (An **X** is used to indicate any number from an IP address.)

  X.X.X.0 matches all machines whose IP addresses match the first three numbers in the specified address. For example, the address **142.2.50.0** matches both **142.2.50.119** and **142.2.50.120**, but does not match **142.2.55.86**.

  X.X.0.0 matches all machines whose IP addresses match the first two numbers in the specified address. For example, the address **142.2.0.0** matches both **142.2.50.119** and **142.2.55.86**, but does not match **142.4.30.95**.

  X.0.0.0 matches all machines whose IP addresses match the first number in the specified address. For example, the address **142.0.0.0** matches both **142.2.50.119** and **142.4.30.95**, but does not match **158.98.4.10**.

You cannot define an entry in the Protection Database that consists of four wildcards **0.0.0.0**.

-id specifies the AFS UID of each new user or machine.

Providing this argument is not generally recommended, as it can cause some complicating side effects. If the issuer does not provide it, the Protection Server will automatically allocate the next available AFS UID, which is one greater than the current value of the `UserMax` counter.

If AFS UIDs are specified, it is best to provide an AFS UID for each specified user name/machine IP address so that the two can be paired (that is, the first user name/IP address gets the first AFS UID, the second gets the second, etc.) If there are more user names/IP addresses than AFS UIDs, then the Protection Server allocates its own AFS UIDs to the unmatched user names/IP addresses. In the reverse situation, the excess AFS UIDs are ignored.

If a specified AFS UID is greater than the current value of the `UserMax` counter, then `UserMax` is reset to equal that AFS UID. Specifying an AFS UID less than `UserMax` has no effect on the value of `UserMax`; however, it is not recommended because of possible conflicts with previously created user/machine entries. Before specifying AFS UIDs, the administrator may wish to use the **pts listmax** command to learn the current value of `UserMax`.

-cell specifies the cell in which to run the command. See section 6.3 for more details.

-noauth tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.

-test tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.

-force directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following shows the creation of user *johnson*.

% pts cu johnson

The following shows the creation of users *smith* and *pat*. The issuer explicitly assigns them AFS UIDs 1143 and 1144 respectively.

% pts cu -na smith pat -i 1143 1144

The following shows the creation of three machine entries in the Transarc Corporation cell. Three entries are required to encompass all of the machines on the Transarc Corporation subnet without including machines from other subnets:

% pts cu 158.98.0.0 192.55.207.0 192.54.226.0

PRIVILEGE REQUIRED

Issuer must belong to the *system:administrators* group.

MORE INFORMATION

pts listmax

pts setmax

pts delete — delete entries from Protection Database.

pts delete -nameorid <*user or group name or id*>+ **[-cell** <*cell name*>**]**
[-noauth] [-test] [-force] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

pts del -na <*user or group name or id*>+ **[-c** <*cell name*> **[-no] [-t] [-f] [-h]**

DESCRIPTION

Removes each specified user, machine, and/or group entry from the Protection Database. When a user or group entry is deleted, its AFS UID appears on ACLs instead of the user name or group name. (Use the **fs cleanacl** command to remove obsolete AFS UIDs from ACLs.)

Deleting a user or machine entry removes the user name or IP address from the group membership lists. Deleting a group entry causes:

- The group name to be removed from the membership list of every user and machine Protection Database entry on which it appeared.
- The group-creation quota of the group's creator to be incremented by one, even if the creator no longer owns the group. See the description of the **pts setfields** or **pts examine** command for more information about group creation quotas.

ARGUMENTS

-name	specifies the name or AFS UID of each user or group, or the IP address or AFS UID of each machine entry to be deleted. Users, machines, and groups can be specified on the same command line, as can names (IP addresses for machines) and AFS UIDs. Precede the AFS UID of each group with a hyphen to indicate that it is negative.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following shows users *pat* and *terry* being removed from the Protection Database:

% pts del pat terry

The following shows the issuer removing the group with AFS UID -215 from the Protection Database. Using the **-i** switch is required because no instance is provided for the **-name** argument.

% pts del -i -215

PRIVILEGE REQUIRED

To delete user and machine entries, the issuer must belong to the *system:administrators* group. To delete group entries, the issuer must either be the group's owner or belong to the *system:administrators* group.

MORE INFORMATION

fs cleanacl

pts adduser

pts creategroup

pts createuser

pts examine

pts removeuser

pts setfields

pts examine — show entries from the Protection Database.

pts examine -nameorid <*user or group name or id*>+ **[-cell** <*cell name*>**] [-noauth] [-test] [-force] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts e -na <*user or group name or id*>+ **[-c** <*cell name*>**] [-no] [-t] [-f] [-h]**

DESCRIPTION

Displays information about the specified user, machine, and/or group entries, as described in the OUTPUT section below.

ARGUMENTS

-name	specifies the name or AFS UID of each user or group, or the IP address or AFS UID of each machine whose Protection Database entry is to be displayed. Users, machines, and groups can be specified on the same command line, as can names (IP addresses for machines) and AFS UIDs. Precede the AFS UID of each group with a hyphen to indicate that it is negative.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

OUTPUT

The output of the command contains the following information about each specified entry:

- ***name***. For users, this is the user name. For machines, it is the IP address. For groups, it is the complete group name with owner field included if applicable. This field may be changed with the **pts rename** command.
- ***AFS UID***. This number is the AFS UID of the entry. It is positive for users and machines, negative for groups.
- ***owner***. This is the user/group entitled to administer the entry. For user and machine entries, this field is automatically set to *system:administrators* at creation time and should not be changed. For group entries, this field may be changed with the **pts chown** command.

- ***creator***. This is the user who initially created the entry; its main use is as an audit trail.
- ***memberships field***. For user and machine entries, this number indicates the number of groups to which the user or machine belongs; for group entries, it indicates the number of users and/or machines belonging to the group. The **pts membership** command reveals the identities of the groups, users, and machines.
- ***protection field***. This field defines the access and manipulation privileges for the entry. The value may be changed with the **pts setfields** command.

  There are five slots, one for each of the five privacy flags. The flags always appear in the order listed here, and something always appears in all five slots. The meaning of the privacy flags and the single letter corresponding to each follows:

STATUS **(s)**:	list status information about an entry (issue the **pts examine** command).
OWNED **(o)**:	list the groups owned by a user or group (issue the **pts listowned** command).
MEMBERSHIP **(m)**:	list the groups to which a user or machine belongs or the users and machines who belong to a group (issue the **pts membership** command).
ADD **(a)**:	add a user or machine to a group (issue the **pts adduser** command). This makes sense only with group entries.
REMOVE **(r)**:	remove a user or machine from a group (issue the **pts removeuser** command). This makes sense only with group entries.

  One of three entries may appear in each flag slot; these entries define the set of users that possesses the right associated with that slot:

  – A ***hyphen*** in the slot indicates that only the owner and members of *system:administrators* have the right. For user entries, the user in question also has the right.

  – A ***lowercase letter*** in the slot indicates that members of the group have the right. This applies to group entries only.

  – An ***uppercase letter*** in the slot indicates that everyone has the right. This is equivalent to *system:anyuser* on standard directory ACLs.

  The default privacy flags are "`S----`" for user and machine entries and "`S-M--`" for group entries.

  User and machine entries do not have members, so the last two slots, ADD and REMOVE, are meaningless for them. For the same reason, there is no sensible interpretation of any of the lowercase letters for user entries; they are equivalent to the hyphen. The lowercase letters are equivalent to the hyphen for self-owned groups, too, since all the members of a self-owned group are owners of the group.

- ***group quota***. This number indicates how many additional groups the user can create. When a user or machine entry is created, the default value is `20`; however, a machine cannot create groups because it is not possible to authenticate as a machine. The **pts setfields** command allows system administrators to reset this number. This field is meaningless for groups and should always be `0` for them.

EXAMPLES

The following shows the output for *terry*'s entry:

% **pts ex terry**

```
Name: terry, id: 1045, owner: system:administrators, creator: admin,
membership: 9, flags: S----, group quota: 15.
```

The following shows the output for the machine entry *158.98.4.10*:

% **pts ex 158.98.4.10**

```
Name: 158.98.4.10, id: 5151, owner: system:administrators, creator: byu,
membership: 1, flags: S----, group quota: 20.
```

In the following, the issuer is interested in the groups with AFS UIDs -673 and -674:

% **pts ex -673 -675**

```
Name: terry:friends, id: -673, owner: terry, creator: terry, membership: 5,
flags: S-M--, group quota: 0.  Name: smith:colleagues, id: -675, owner: smith,
creator: smith, membership: 14, flags: SOM--, group quota: 0.
```

PRIVILEGE REQUIRED

Depends on how the entry's first privacy flag is set:

- If it is lowercase `s` and this is a group entry, only members of the group and of *system:administrators* may examine the entry. If this is a user entry, only members of *system:administrators* and the user may examine the entry. If this is a machine entry, only members of *system:administrators* may examine the entry.
- If it is uppercase S, everyone on the system may examine the entry.

It is not possible to set this flag to the hyphen (the command interpreter would not be able to distinguish the hyphen from one that indicates a flag or switch.

The **pts setfields** command sets the privacy flags.

MORE INFORMATION

pts adduser

pts membership

pts rename

pts removeuser

pts chown

pts setfields

pts listowned

pts help — show syntax of specified **pts** command(s) or list functional description for all of them.

pts help [-topic <*help string*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts h [-t <*help string*>+**] [-h]**

DESCRIPTION

Displays the first line (name and short description) of every **pts** command's help entry, if no *help string* is provided. For each operation code specified with **-topic**, it outputs the entire help entry. See the **Output** section below.

ARGUMENTS

-topic	specifies the operation codes for which syntax is to be provided. If it is omitted, the first line (name and short description) of every **pts** command is displayed.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

OUTPUT

The online help entry for each **pts** command consists of two or three lines:

- The first line names the command and briefly describes what it does.
- The second line shows any aliases the command has (it does not appear for every command).
- The final line, which begins with "`Usage:`", lists the command's arguments and flags in the prescribed order. Online help entries use the same symbols (for example, brackets) as the command definitions in this manual. For an explanation of their meaning, see page x of the introductory *About This Manual* chapter.

EXAMPLES

The following displays the online help entry for the **pts membership** command:

% **pts help membership**

```
pts membership:  list membership of a user or group aliases: groups
Usage: pts membership -nameorid <user or group name or id>+ [-cell
<cell name>] [-noauth] [-test] [-force] [-help]
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

pts apropos

pts listmax — show `GroupMax` and `UserMax` counters.

pts listmax [-cell <*cell name*>**] [-noauth] [-test] [-force] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts listm [-c <*cell name*>**] [-n] [-t] [-f] [-h]**

DESCRIPTION

Displays the values of the `UserMax` and `GroupMax` counters, which the Protection Server uses to keep track of the AFS UIDs it allocates to new users/machines and groups, respectively. The numbers shown were assigned to the last user/machine and group added to the Protection Database. The next group created will receive an AFS UID one number less than the `GroupMax` shown, and the next user/machine an AFS UID one number greater than the `UserMax` shown.

Members of *system:administrators* may issue **pts setmax** to reset one or both counters.

ARGUMENTS

-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

OUTPUT

The output lists the counters in the following format:

`Max user id is` *UserMax* `and max group id is` *GroupMax*.

EXAMPLES

The following displays the output of this command:

% **pts listm**

```
Max user name is 1271 and max group id is -382.
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

pts setmax

pts listowned — show groups owned by specified users and/or groups.

pts listowned -nameoreid <*user or group name or id*>+ **[-cell** <*cell name*>**]**
[-noauth] [-test] [-force] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

pts listo -na <*user or group name or id*>+ **[-c** <*cell name*>**] [-no] [-t] [-f] [-h]**

DESCRIPTION

Lists the names of the groups owned by the specified users and/or groups. If a value of *0* is provided for **-nameorid**, the command also lists all "orphaned groups" (groups that still exist in the database even though their owners have been deleted from it).

Note: Although a machine can theoretically own a Protection Database entry, this ownership confers no functionality and is not recommended. All discussion of this command will refer only to user and group ownership.

ARGUMENTS

-nameorid	specifies the name or AFS UID of each user or group for which group ownership information is to be displayed. Users and groups can be specified on the same command line, as can names and AFS UIDs. Precede the AFS UID of each group with a hyphen to indicate that it is negative. Providing a value of *0* (zero) lists entries for groups that still exist in the database even though the entries for their owners have been deleted.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

OUTPUT

The command displays group ownership information for each specified user or group. The first line of the output displays the *name* and *AFS UID* of the user or group for which ownership information is to be displayed, as follows:

`Groups owned by` *name* `(id:` *AFS UID*`) are:`

A list of groups owned by the user or group follows. The list does not include groups owned by groups that the user or group owns or to which the user or group belongs. If the user or group does not own any groups, the preceding line appears with nothing after it.

If the second privacy flag is the hyphen (as it is by default), most users will not be allowed to list the owned groups. A `Permission denied` message is displayed in this case.

EXAMPLES

The following shows the groups owned by *terry* and *terry:friends*, which does not own any groups:

% **pts listo terry terry:friends**

```
Groups owned by terry (id: 1045) are:  terry:friends terry:project1
terry:project2 Groups owned by terry:friends (id: -673) are:
```

The following shows the message that *smith* receives when he tries to list the groups owned by *pat*. The message is displayed because the second privacy flag on *pat*'s entry is the hyphen.

% **pts listo pat**

```
pts: Permission denied so failed to get owner list for pat (id: 1144)
```

PRIVILEGE REQUIRED

Depends on how the entry's second privacy flag is set:

- If it is the hyphen and this is a group entry, only the owner of the group and members of *system:administrators* may list the groups owned by this group. If this is a user entry, only the user himself/herself and members of *system:administrators* may list the groups owned by the user. The hyphen is the default.
- If it is uppercase `O`, everyone on the system may list the groups owned by this user/group.

It is not possible to set the second flag to lowercase `o` (which would be equivalent in meaning to the hyphen). The **pts setfields** command sets the privacy flags; the **pts examine** command displays them.

MORE INFORMATION

pts examine

pts setfields

pts membership — show membership list for user or group.

pts membership -nameorid <*user or group name or id*>+ **[-cell** <*cell name*>**]** **[-noauth] [-test] [-force] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts m -na <*user or group name or id*>+ **[-c** <*cell name*>**] [-no] [-t] [-f] [-h]**

pts groups -na <*user or group name or id*>+ **[-c** <*cell name*>**] [-no] [-t] [-f] [-h]**

DESCRIPTION

For each specified user name and/or machine IP address, this command lists the groups to which the user or machine belongs. For each specified group name, it lists the users and/or machines who belong to the group. The output is limited to 5000 names for any user, machine, or group, even if there are actually more in the list.

It is not possible to list the members of *system:anyuser* or *system:authuser*, nor do they appear in the list of groups to which a user belongs. The membership of these groups is not stable, as it depends on, respectively, each user's current login or authentication status.

The **pts adduser** command adds users and machines to groups, and the **pts removeuser** command removes them from groups

ARGUMENTS

-nameorid	specifies the name or AFS UID of each user or group, or the IP address or AFS UID of each machine about which group membership information is to be displayed. Users, machines, and groups can be specified on the same command line, as can names (IP addresses for machines) and AFS UIDs. Precede the AFS UID of each group with a hyphen to indicate that it is negative.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

OUTPUT

For each specified user and machine, the output contains the following header line followed by a list of the groups to which the user belongs:

`Groups` *user name* `(id:` *AFS UID*`) is a member of:`

For each specified group, the output contains the following header line followed by a list of the users and/or machines who belong to the group:

`Members of` *group name* `(id:` *AFS UID*`) are:`

The command will list up to 5000 members of a group, or 5000 groups to which an individual belongs.

EXAMPLES

The following shows the groups to which *pat* belongs and the members of *smith:friends*. Note that *pat*'s third privacy flag must have changed from the default hyphen for a non-administrative user to elicit this type of listing.

% **pts mem -name pat smith:friends**

```
Groups pat (id: 1144) is a member of:  smith:friends staff
johnson:project-team Members of smith:friends (id: -562) are:  pat
terry jones richard thompson
```

The following shows the groups to which machine *158.98.4.10* belongs (*bin-prot* is a binary protection group composed solely of machines for licensing purposes):

% **pts mem 158.98.4.10**

```
Groups 158.98.4.10 (id: 5151) is a member of:  bin-prot
```

PRIVILEGE REQUIRED

Depends on how the entry's third privacy flag is set:

- If it is the hyphen and this is a group or machine entry, only the owner of the group and members of *system:administrators* may list the members of the group. If it is a hyphen and this is a user entry, only the user himself/herself and members of *system:administrators* may list the groups to which the user belongs.
- If it is a lowercase `m` and this is a group entry, privilege extends to members of the group. If it is a lowercase `m` and this is a user or machine entry, the meaning is equivalent to the hyphen.
- If it is an uppercase `M`, privilege extends to everyone on the system.

The **pts setfields** command sets the privacy flags, and the **pts examine** command displays them.

MORE INFORMATION

pts examine

pts setfields

pts removeuser — remove users from groups in Protection Database.

pts removeuser -user <*user name*>+ **-group** <*group name*>+
[-cell <*cell name*>**] [-noauth] [-test] [-force] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts rem -u <*user name*>+ **-g** <*group name*>+ **[-c** <*cell name*>**] [-n] [-t] [-f] [-h]**

DESCRIPTION

Removes each user and/or machine entry from membership in the specified *group name* entry. It also removes each *group name* from the list of memberships stored with each user and/or machine entry.

WARNING

The effects of removing a user from a group are not immediate. Before an user member of a group is denied the access rights of that group, he or she must re-authenticate (issue **klog**). As long as a user retains old tokens, he or she retains the access rights acquired with those tokens.

ARGUMENTS

-name	specifies the user name of each user entry or the IP address of each machine entry to be removed.
-group	specifies the name of each group from which each user/machine entry should be removed.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following removes *smith* from the groups *staff* and *staff:finance*. Note that no switches are necessary because only a single instance is provided for the first argument (the user name).

```
% pts rem smith staff staff:finance
```

The following removes the three machine entries in the Transarc Corporation subnet from the group *bin-prot*:

```
% pts rem 158.98.0.0 192.55.207.0 192.54.226.0 -gr bin-prot
```

PRIVILEGE REQUIRED

Depends on how the group's fifth privacy flag is set:

- If it is the hyphen, only the group's owner and members of *system:administrators* may remove members.
- If it is lowercase `r`, privilege extends to members of the group.

It is not possible to set the fifth flag to uppercase `R`.

The **pts setfields** command sets the privacy flags and the **pts examine** command displays them.

MORE INFORMATION

pts adduser

pts membership

pts setfields

pts rename — change name of entry in Protection Database.

pts rename -oldname <*old name*> **-newname** <*new name*>
[-cell <*cell name*>**] [-noauth] [-test] [-force] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts ren -o <*old name*> **-ne** <*new name*> **[-c** <*cell name*>**] [-no] [-t] [-f] [-h]**

DESCRIPTION

Changes the name of a user or group entry called *old name* to *new name*. It is not possible to change a user name to be like a regular group name (with a colon in it). Members of *system:administrators* may change regular group names into prefix-less names and vice versa. When changing a prefix-less group name into a regular group name or a regular group name to another regular group name, the owner-name field of the new name must correctly reflect the group's ownership.

WARNING

By convention, many aspects of an AFS user account have the same name as the user's Protection Database entry, including the Authentication Database entry, volume, and mount point. If this command is used to change a user name, it is recommended for consistency's sake that the names of all relevant related entities also be changed. See the chapter on user accounts in the *AFS System Administrator's Guide* for complete instructions.

ARGUMENTS

-oldname	specifies the current full name of the entry.
-newname	specifies the new full name for the entry. For regular groups, the *owner-name* (pre-colon) part of the new name must reflect the true ownership of the group.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following changes the name of the group *staff*, owned by the privileged user *admin*, to *admin:staff*:

% pts ren staff admin:staff

The following changes the name of the group *admin:finance* to *finance*.

% pts ren admin:finance finance

PRIVILEGE REQUIRED

To change a regular group name to a prefix-less name or vice versa, the issuer must belong to *system:administrators*. For regular-to-regular or prefix-less-to-prefix-less changes, the issuer must be the group's owner or belong to the *system:administrators* group. To change user entries, the issuer must belong to the *system:administrators* group.

MORE INFORMATION

pts chown

pts setfields — set privacy flags or group creation quota for a protection Database entry.

pts setfields **-nameorid** <*user or group name or id*>+
[-access <*set privacy flags*>**]**
[-groupquota <*set limit on group creation*>**]**
[-cell <*cell name*>**]** **[-noauth]** **[-test]** **[-force]** **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts setf **-na** <*user or group name or id*>+ **[-a** <*set privacy flags*>**]**
[-g <*set limit on group creation*>**]** **[-c** <*cell name*>**]** **[-no]** **[-t]** **[-f]** **[-h]**

DESCRIPTION

Sets group-creation quota and/or the privacy flags for each entry indicated. The **pts examine** command shows the current settings of the quota and flags.

Note: Group-creation quota is meaningless for machines and groups, because it is impossible to authenticate as a group or machine.

ARGUMENTS

-nameorid specifies the name or AFS UID of each user or group, or the IP address or AFS UID of each machine for which privacy flags or group creation quota is to be set. Users, machines, and groups can be specified on the same command line, as can names (IP address for machines) and AFS UIDs. Precede the AFS UID of each group with a hyphen to indicate that it is negative.

-access specifies a list of five characters, one for each of the protection rights on a Protection Database entry. The default rights for user and machine entries are `S----` and for group entries `S-M--`. It is necessary to provide this argument only if the default settings are not satisfactory.

There are five slots, one for each of the five privacy flags. The flags always appear in the order listed here, and something always appears in all five slots. The meaning of the privacy flags and the single letter corresponding to each follows:

STATUS **(s)**: list status information about an entry (issue the **pts examine** command).

OWNED **(o)**: list the groups a user or group owns (issue the **pts listowned** command).

MEMBERSHIP **(m)**: list the groups to which a user or machine belongs or the users and machines that belong to a group (issue the **pts membership** command).

ADD **(a)**: add a user or machine to a group (issue the **pts adduser** command). This is meaningful only for group entries.

REMOVE **(r)**: remove a user or machine from a group (issue the **pts removeuser** command). This is meaningful only for group entries.

The issuer of the command determines who receives each right by typing one of three types of characters in each slot, as follows:

- To grant the right only to the owner of the entry and members of *system:administrators*, type a ***hyphen*** in the slot. For user entries, the hyphen also gives the user this right.
- To grant the right only to members of the group, type the ***lowercase version*** of the letter associated with the right (as defined just above). Do not type lowercase letters for user or machine entries; they are meaningless for them.
- To grant the right to everyone, type the ***uppercase version*** of the letter associated with the right. This is similar to putting *system:anyuser* on standard directory ACLs.

The following restrictions define how the issuer may set the rights:

- As mentioned above, it is always necessary to specify a value for all five flags.
- Do not set the flags (other than the first flag) in user or machine entries to lowercase letters. Lowercase letters have no sensible interpretation for these entries, because these entries cannot have members. The hyphen is equivalent to lowercase letters for self-owned group entries, too, since all the members of a self-owned group are owners of the group itself.
- Legal values for the first flag ("check status") are lowercase `s` and uppercase `S`. The hyphen is not legal because the command interpreter cannot distinguish an initial hyphen from the indicator of a switch or flag. For user and machine entries, the lowercase `s` is equivalent to the hyphen.
- Legal values for the second flag ("list owned") are the hyphen and uppercase `O`. For groups, the hyphen is equivalent to the disallowed lowercase `o`, because members of a group own any groups owned by the group.
- Legal values for the third ("membership") flag are the hyphen, lowercase `m`, and uppercase `M`.
- Legal values for the fourth flag ("add members") are the hyphen, lowercase `a`, and uppercase `A`. This flag has no sensible interpretation for user and machine entries, but must be set nonetheless, preferably to the hyphen.
- Legal values for the fifth flag ("remove members") are the hyphen and lowercase `r`. This flag has no sensible interpretation for user and machine entries, but must be set nonetheless, preferably to the hyphen.

-groupquota specifies the number of additional groups a user can create (it does not matter how many they have created already). This number is meaningless for groups and machines, since it is not possible to authenticate as a group or machine; there is no reason to set or change this value for those entries.

-cell specifies the cell in which to run the command. See section 6.3 for more details.

-noauth tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.

-test tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.

-force directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following changes the privacy flags on the group *operators*. The group retains the default "S" and "M" flags, but it acquires the "a" and "r" flags so that its members can add and remove members.

```
% pts setf operators -a S-Mar
```

The following changes the privacy flags and sets group quota on the user entry *admin*. The default "S" privacy flag is retained, and the "O" and "M" flags are enabled. The group quota is adjusted to 50.

```
% pts setf admin -a SOM-- -g 50
```

PRIVILEGE REQUIRED

For group entries, the issuer must be the group's owner or belong to the *system:administrators* group. For user entries, setting group-creation quota requires that the issuer belong to the *system:administrators* group; setting the privacy flags requires that the issuer be the owner of the group or belong to the *system:administrators* group.

MORE INFORMATION

pts adduser

pts examine

pts listowned

pts membership

pts removeuser

pts setmax — set value of GroupMax and/or UserMax counter.

pts setmax [-group <*group max*>**] [-user** <*user max*>**] [-cell** <*cell name*>**] [-noauth] [-test] [-force] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

pts setm [-g *group max*>**] [-u** <*user max*>**] [-c** <*cell name*>**] [-n] [-t] [-f] [-h]**

DESCRIPTION

Sets the GroupMax counter (for the group AFS UID) to *group max* and/or the UserMax counter (for the user/machine AFS UID) to *user max*. The **pts listmax** command displays the current value of the two counters.

ARGUMENTS

-group	specifies the value of the GroupMax counter. Precede the value with a hyphen to indicate that it is negative. When a system administrator next creates a group entry with **pts creategroup**, the Protection Server will assign it an AFS UID one less (that is, more negative) than GroupMax.
-user	specifies the value of the UserMax counter. When an administrator next creates a user or machine entry with **pts createuser**, the Protection Server will assign it an AFS UID one greater than UserMax.
-cell	specifies the cell in which to run the command. See section 6.3 for more details.
-noauth	tells the Protection Server to assign the unprivileged identity *anonymous* to the issuer. See section 6.3 for more details.
-test	tells the command interpreter to contact the machine listed in */usr/afs/etc/CellServDB* for execution of the command. See section 6.3 for more details.
-force	directs the command to continue performing as much of an operation as it can, even if it encounters problems while executing. See section 6.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 6.3 for more details.

EXAMPLES

The following sets `GroupMax` to -500 and `UserMax` to 1000. The switches are not actually required, since the arguments appear in the prescribed order, but help clarify which argument is which.

```
% pts -g -500 -u 1000
```

PRIVILEGE REQUIRED

Issuer must belong to the *system:administrators* group.

MORE INFORMATION

pts creategroup

pts createuser

pts listmax

7. The uss Commands and Template Lines

The **uss** commands help administrators create user accounts in their system more easily and efficiently. Without them, administering accounts can become time consuming. Creating an account "by hand" requires issuing at least six separate commands to five different AFS servers, whereas a single **uss add** command can accomplish all the equivalent actions once the issuer sets up the necessary template. Even better, a single issue of the **uss bulk** command can create multiple accounts at once, after the necessary preparations have been made.

Refer to chapter 11 for a complete summary list of **uss** commands and their syntax.

7.1. Common Arguments and Flags

Most **uss** commands accept the following optional arguments and flags. They are listed in the command descriptions where they apply, and are described here in detail:

[-cell <*cell name*>**]**

This argument specifies that the command should be run in the cell specified by *cell name*. The issuer may abbreviate *cell name* to the shortest form that distinguishes it from the other cells listed in */usr/vice/etc/CellServDB* on the client machine on which the command is issued. By default, commands are executed in the local cell, as defined:

- First, by the value of the environment variable AFSCELL.
- Second, in */usr/vice/etc/ThisCell* on the client machine on which the command is issued.

[-admin <*administrator to authenticate*>**]**

This argument names the user name whom the Authentication Servers should authenticate for purposes of creating an Authentication Database entry. If the issuer does not provide it, the authentication is attempted for the effective user name (the one under which the issuer is logged into the local machine's UNIX file system). With or without this argument, UNIX commands invoked by this command (for instance, **/etc/chown**) are executed under the effective user name.

[-dryrun]

This flag indicates that the command interpreter should not actually execute the command, but should report all the actions it would perform if executing it. This flag is useful for verifying that the account will be configured as the issuer wishes. It does not, however, necessarily reveal failures that will result if the **uss** command interpreter encounters any circumstances that make it impossible to carry out a requested action. Combining this flag with **-verbose** yields more information.

[-help]

This flag has the same function as the **bos help** command: it prints the command's online help message on the screen. No other arguments or flags should be provided at the same time. Even if they are, this flag overrides them, and the only effect of issuing the command is that the help message appears.

7.2. The Privilege Required for uss Commands

The issuer of a **uss** command must have all the rights required for performing the equivalent actions individually. See the PRIVILEGE REQUIRED section of each command description for details.

uss add — create user account.

uss add -user <*login name*> **[-realname** <*full name in quotes*>**]**
[-pass <*initial passwd*>**] [-server** <*FileServer for home volume*>**]**
[-partition <*FileServer's disk partition for home volume*>**]**
[-mount <*home directory mount point*>**] [-uid** <*uid to assign the user*>**]**
[-template <*pathname of template file*>**]**
[-verbose] [-var <*auxiliary argument pairs (Num val)*>$^{+}$**]**
[-cell <*cell name*>**] [-admin** <*administrator to authenticate*>**]**
[-dryrun] [-skipauth] [-overwrite]
[-pwexpires <*password expires in [0..254] days (0 => never)*>**]**
[-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

uss ad -us <*login name*> **[-r** <*full name in quotes*>**]**
[-pas <*initial passwd*>**] [-se** <*FileServer for home volume*>**]**
[-par <*FileServer's disk partition for home volume*>**]**
[-m <*home directory mount point*>**] [-ui** <*uid to assign the user*>**]**
[-t <*pathname of template file*>**] [-ve] [-va** <*auxiliary argument pairs (Num val)*>$^{+}$**]**
[-c <*cell name*>**] [-a** <*administrator to authenticate*>**] [-d] [-sk] [-o]**
[-pw <*password expires in [0..254] days (0 => never)*>**] [-h]**

DESCRIPTION

Creates entries in the Protection Database and Authentication Database for the user name *login name*. If *initial passwd* is provided, it is stored as the user's password in the Authentication Database after conversion into a form suitable for use as an encryption key. By default, the Protection Server automatically allocates an AFS UID for the new user; the issuer may specify an alternate with the **-uid** argument.

The other results of the command depend on what appears in the template file specified with *pathname of template file*. The issuer must provide any argument whose corresponding variable appears in the template (the ARGUMENTS section below maps the arguments and variables). If the issuer provides an argument for which the corresponding variable does not appear in the template, the value is ignored. Failure to provide a value for a variable causes the account creation to fail at the point where the command interpreter first encounters the variable in the template.

Syntax definitions for the lines in the template follow descriptions of the **uss** commands.

WARNING

If this command is issued on a machine running Ultrix and the template file being used is not 0-length, the user must use the **-admin** argument to adopt a privileged AFS identity while remaining "root" in the local machine's UNIX file system. Ultrix allows only "root" to issue the **/etc/chown** command that **uss add** invokes to set the owner of files and directories created by template lines. At the same time, AFS allows only a privileged administrator to issue the AFS commands invoked; "root" is not normally a privileged AFS administrator.

Other operating systems allow users other than "root" to issue **/etc/chown**, but users may still find it convenient to adopt different identities in the AFS and UNIX file systems. Being authenticated in AFS as a privileged user is required under all operating systems.

ARGUMENTS

-user specifies the user name that the user will type at the login prompt. It also becomes the name of the user's Protection and Authentication Database entries.

Corresponding variable in template: `$USER`.

-realname specifies the user's real name. Surround it with double quotes as delimiters, since it contains spaces and possibly punctuation. If not provided, it defaults to the user name provided with **-user**.

Corresponding variable in template: `$NAME`. The most common use for this variable is when creating a file to be incorporated as an entry in the cell's */etc/passwd* file.

-pass defines the user's initial password. Many UNIX applications and utilities require that this be no longer than eight characters; the AFS commands that handle passwords accept strings of virtually unlimited length. If not provided, this argument defaults to the string *changeme*.

Corresponding variable in the template: none.

-server specifies the file server machine on which the user's volume will reside.

Corresponding variable in template: `$SERVER`.

-partition specifies the partition on *file server for home volume* on which the user's volume will reside.

Corresponding variable in template: `$PART`.

-mount specifies the pathname to the user's home directory. If the issuer does not provide a full pathname, it is interpreted relative to the working directory.

Corresponding variable in template: `$MTPT`, but on the "V" line only. Occurrences of `$MTPT` on lines following the "V" line take their value from the "V" line's <*mount point*> field. Thus the exact value of this command line argument becomes the value for post-"V" line occurrences of `$MTPT` only if `$MTPT` appears alone in the <*mount point*> field.

-uid specifies the AFS UID to be assigned to the new user. It should be a positive integer. The issuer may wish to pre-verify with **pts listmax** that the intended number is greater than the current largest AFS UID: if the requested AFS UID is already in use, the account creation will terminate with an error. If the issuer does not provide this argument, the Protection Server assigns an AFS UID automatically; as with manual account creation, automatic allocation is preferred.

Corresponding variable in template: `$UID`.

-template specifies which template the command interpreter should consult. If the issuer does not provide this argument, the command interpreter looks for a template called *uss.template* in the following directories, which it searches in this order:

1. The working directory
2. */afs/**cellname**/common/uss*, where ***cellname*** names the local cell
3. */etc*

If the issuer provides a name other than *uss.template* without a pathname, the command interpreter looks for it in the directories just listed, in the same order. If the issuer provides a full or partial pathname, the command interpreter consults the specified file without consulting the directories just listed; it interprets partial pathnames relative to the working directory.

If the template is 0-length, the new account will consist solely of Protection and Authentication Database entries.

-verbose causes the command interpreter to produce a more detailed trace of the actions it is executing. By default, only warnings and error messages appear.

-var specifies values for each of the number variables `$1` through `$9` found in the template. The number variables allow the issuer to provide values for variables not built-in to the **uss add** command.

Corresponding variables in template: `$1` through `$9`.

Each instance of this argument has two parts, separated by a space:

- An integer between 1 and 9, not preceded by the dollar sign. Each number matches the variable in the template.
- A value.

See the chapter on **uss** in the *AFS System Administrator's Guide* for further explanation.

-cell specifies the cell in which to run the command. See section 7.1 for more details.

-admin names the user whom the Authentication Servers should authenticate for purposes of creating an Authentication Database entry. See section 7.1 for more details.

Note: Issuers of this command on machines running Ultrix must use this argument in order to adopt an privileged AFS identity while remaining "root" in the local machine's UNIX file system. See the preceding WARNING.

-dryrun indicates that the command interpreter should not actually execute the command, but should report all the actions it would perform if executing it. See section 7.1 for more details.

-skipauth directs the command to bypass the Authentication Server, allowing a site using MIT Kerberos to use the command with that form of authentication.

-overwrite instructs the command interpreter to overwrite any directories, files and links that exist in the file system and for which it also finds definitions on template "D", "E", "F", "L" and "S" lines. If this flag is omitted, the command interpreter prompts once for confirmation that the issuer really wants to overwrite all such elements.

-pwexpires defines the number of days for which *name of user*'s password will be valid from the date that the password is changed. Beyond the specified number of days, the password will be

invalid (that is, the user cannot use it to authenticate with his or her account). For up to 30 days after password expiration, the user can use his or her old password to change passwords.

The range of valid values is 0 through 254. Use 0 to indicate that the password never expires; use 1 through 254 to specify the number of days until expiration. By default, a user's password will never expire.

Corresponding variables in template: `$PWEXPIRES`.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 7.1 for more details.

EXAMPLES

When the template is 0-length, the following creates a "dummy" account called *admin* with entries in the Protection and Authentication Databases only.

% uss add admin

The combination of the following "V" line in the template *uss.tpl* and **uss add** command creates Protection and Authentication Database entries for user name *smith*. It would also create a volume called `user.smith` with a quota of 2500 kilobyte blocks, mounted at */afs/transarc.com/usr/smith*, the ACL for which gives *smith* all rights. See the description of the "V" line for an explanation of its fields. Note that only the template's name is provided, not its pathname, as it is assumed it resides in one of the three expected directories.

```
V user.$USER $SERVER.transarc.com /vice$PART $1
     /afs/transarc.com/usr/$USER $UID $USER all
```

% uss add smith "John Smith" js_pswd fs2 b -t uss.tpl -v 1 2500

The chapter on **uss** in the *AFS System Administrator's Guide* presents more extended examples of template lines and commands.

PRIVILEGE REQUIRED

Issuer (or person named with **-admin** flag) must belong to the *system:administrators* group in the Protection Database and must have the `ADMIN` flag in his or her Authentication Database entry. If the template contains a "V" line, the issuer must be listed in */usr/afs/etc/UserList* and must have at least ADMINISTER and INSERT rights in the directory where the mount point is to be created. If the template includes lines that create other types of objects (directories, files or links), the issuer must have the privilege(s) necessary to create them.

Under Ultrix, only "root" can issue the **/etc/chown** command invoked when a file or directory is created. See the WARNING section.

uss apropos — show each help entry containing keyword.

uss apropos -topic <*help string*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

uss ap -t <*help string*> **[-h]**

DESCRIPTION

Displays the first line of the help entry for any **uss** command that has *help string* in its name or short description.

ARGUMENTS

-topic	specifies the keyword string to search for. If it is more than a single word, surround it with double quotes or other delimiters. Type all *help string*s for **uss** commands in all lowercase letters.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 7.1 for more details.

OUTPUT

The first line of a command's online help entry names the command and briefly describes what it does. The **uss apropos** command displays that first line for any **uss** command where *help string* is part of the command name or first line.

To see the remaining lines in a help entry, which provide the command's alias (if any) and syntax, use the **uss help** command.

EXAMPLES

The following lists all **uss** commands that have the word "*create*" in their operation code or short online description.

```
% uss apropos create
add: create a new user
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

uss help

uss bulk — execute multiple **uss** commands.

uss bulk -file <*bulk input file*> **[-template** <*pathname of template file*>**]**
[-verbose] [-cell <*cell name*>**]**
[-admin <*administrator to authenticate*>**] [-dryrun] [-skipauth]**
[-overwrite] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

uss b -f <*bulk input file*> **[-t** <*pathname of template file*>**] [-v] [-c** <*cell name*>**]**
[-a <*administrator to authenticate*>**] [-d] [-s] [-o]**

DESCRIPTION

Executes the **uss** commands listed in *bulk input file*, which must already exist. If *bulk input file* has `add` commands in it that create complete (rather than authentication-only) accounts, then the issuer must also specify a template using *pathname of template file*.

WARNING

The following warning applies when all of the following are true:

- This command is issued on machines running Ultrix.
- The bulk file contains `add` lines.
- The template file used for the account creations is not 0-length.

In this set of circumstances, the issuer must use the **-admin** argument to adopt a privileged AFS identity while remaining "root" in the local machine's UNIX file system. Ultrix allows only "root" to issue the **/etc/chown** command that the `add` command invokes to set the owner of files and directories created by template lines. At the same time, AFS allows only a privileged administrator to issue the AFS commands invoked; "root" is not normally a privileged AFS administrator.

Other operating systems allow users other than "root" to issue **/etc/chown**, but users may still find it convenient to adopt different identities in the AFS and UNIX file systems. Being authenticated in AFS as a privileged user is required under all operating systems.

ARGUMENTS

-file names the file containing the **uss** commands to execute. If the issuer does not provide a pathname, the command interpreter assumes the file is in the working directory. The BULK FILE FORMAT section details the proper format for command lines in the bulk file.

-template names the template file for any `add` commands that appear in the *bulk input file*. If the issuer does not provide a pathname, the command interpreter assumes the template file is in the working directory. See the *AFS System Administrator's Guide* for a definition of template format.

-verbose causes the command interpreter to produce (on stdout) a more detailed trace of the actions it is executing. By default, only warnings and error messages appear.

-cell specifies the cell in which to run the command. See section 7.1 for more details.

-admin names the user whom the Authentication Servers should authenticate for purposes of creating an Authentication Database entry. See section 7.1 for more details.

Note: When the bulk file contains `add` lines, issuers of the **uss bulk** command on machines running Ultrix must use this argument in order to adopt an privileged AFS identity while remaining "root" in the local machine's UNIX file system. See the preceding WARNING.

-dryrun indicates that the command interpreter should not actually execute the command, but should report all the actions it would perform if executing it. See section 7.1 for more details.

-skipauth directs the command to bypass the Authentication Server, allowing a site using MIT Kerberos to use the command with that form of authentication.

-overwrite instructs the command interpreter (when `add` lines appear in the bulk file) to overwrite any directories, files and links that exist in the file system related to the user for which it also finds definitions on template "D", "E", "F", "L" and "S" lines. If this flag is omitted, the command interpreter prompts, once for each `add` line in the bulk file, for confirmation that the issuer really wants to overwrite those elements. This flag should not be used if the bulk file does not contain `add` lines.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 7.1 for more details.

BULK FILE FORMAT

Five types of command lines can appear in the bulk file: `add`, `delete`, `exec`, `savevolume`, and `delvolume`. Each command line should have a carriage return only at the end, even though it may cover several lines on the screen.

The `add` ***line***

Each `add` line is the equivalent of a **uss add** command issued on the command line. Begin the line with "`add`" only, not "`uss add`", and provide the arguments in the same order they would appear on the **uss add** command line, separating each with a colon. Only the first argument, corresponding to the command line argument **-user**, is required. To omit a value for an argument (presumably because it is optional or the template specifies a constant value for it), type nothing between two colons. After the last argument provided, end the line with either a colon and carriage return, or a carriage return alone.

Fields nine through eighteen are for assigning values to the number variables, with the fields listed in increasing numerical order. The issuer must indicate either a value or an empty field (nothing between two colons) for every variable that precedes the last one being assigned an actual value. It is acceptable, but not necessary, to indicate empty fields for any number variables following the last one actually assigned a value.

The following representation uses the same identifiers for arguments as the **uss add** definition. The "{ }" notation indicates that each entry between vertical bars is one possible choice. Remember that an instruction like this would appear on a single line in the actual bulk input file.

```
add <login name>[:<full name>][:<initial passwd>][:<password expires>]
[:<file server for home volume>][:<file server's disk partition for home volume>]
[:<home directory mount point>][:<uid to assign the user>]
[:{<var1> | :<var1>:<var2> | :<var1>:<var2>:<var3> |
    . . . . et cetera through. . . .
| :<var1>:<var2>:<var3>:<var4>:<var5>:<var6>:<var7>:<var8>:<var9>}]
```

Do not surround *full name* with double quotes in the bulk input file, even though its counterpart on the regular **uss add** command line must be so surrounded.

The EXAMPLES section below may help to clarify the format of the `add` line.

The `delete` *line*

Each `delete` line is the equivalent of a **uss delete** command issued on the command line. Begin the line with "`delete`" only, not "`uss delete`", and provide the arguments in the same order they would appear on the **uss delete** command line, separating each with a colon. Provide values for at least the first two arguments (corresponding to **-user** and **-mountpoint** on the command line). The third field, corresponding to **-savevolume**, is optional; there are three possible values:

- If the word `savevolume` appears, the volume and VLDB entry will not be removed.
- If the word `delvolume` appears, the volume and VLDB entry will be removed.
- If the field is blank, then the volume will be treated according to the prevailing default (see the following description of the `savevolume` and `delvolume` lines to learn how to set the default).

After the last argument provided, end the line with either a colon and carriage return or a carriage return alone.

The following concrete representation uses the same identifiers for arguments as does the **uss delete** definition. The "{ }" notation indicates that each entry between a vertical line is one possible choice. Remember that an instruction like this would appear on a single line in the actual bulk file. The EXAMPLES section illustrates use of this line.

```
delete <account name>:<full mount point pathname>
       [:{savevolume | delvolume | }]
```

The `exec` *line*

This line causes the indicated command to be executed. The command can be a UNIX shell script or command, a program, or an AFS command. Its format is

```
exec <command string to execute>
```

The savevolume *and* delvolume *lines*

The savevolume and delvolume lines set the default treatment of volumes in delete lines that follow them in the bulk file. The savevolume line prevents the removal of volume and VLDB entry for all subsequent delete lines. The delvolume line causes the removal of volume and VLDB entry for all subsequent delete lines. The default treatment if neither line appears in a bulk file is to remove the volume and the VLDB entry; delete lines that appear before the first savevolume line are also treated this way.

An explicit savevolume or delvolume instruction in the third field of an individual delete line overrides the default in effect at the time. If nothing appears in the third field, the default is obeyed.

Neither command takes arguments (the word savevolume or delvolume should appear by itself on the line). The effect of either line lasts until the end of the bulk file, or until its opposite appears in the file. Multiple instances of each command can be used to toggle back and forth between removal and non-removal of volumes.

EXAMPLES

The following shows the proper format for an add line in the bulk file when only the first (required) argument is provided, and when the first three arguments are provided. In the first case, the user's "real name" is set to *anderson* and password to *changeme* (the defaults).

```
add anderson
add smith:John Smith:js_pswd
```

The following add line example supposes that the Transarc Corporation cell uses a template with the following "V" line in it:

```
V user.$USER $SERVER.transarc.com /vicep$PART 2000
       /afs/transarc.com/usr/$3/$USER $UID $USER all
```

To create an account for users John Smith from the Marketing Department and Pat Jones from the Finance Department, the appropriate **add** commands in the bulk file might be:

```
add smith:John Smith:::fs1:a:::::marketing
add jones:Pat Jones:::fs3:c:::::finance
```

The effect would be to establish an account called *smith* in the Protection and Authentication Databases, with an initial password *changeme* and a value for $UID provided by the Protection Server. His volume, user.smith, would reside on partition "/vicepa" of file server machine "fs1.transarc.com" and would be mounted at */afs/transarc.com/usr/marketing/smith*. He would own his home directory and have full access to it. The account for *jones* would be similar, except that it would reside on partition "/vicepc" of file server machine "fs3.transarc.com" and would be mounted at */afs/transarc.com/usr/finance/jones*.

Notice that the fields corresponding to <*home directory mount point*>, <*uid to assign the user*>, <*var1*> and <*var2*> are empty (between a and marketing on the first example line) since their corresponding variables do not appear in the template file. The <*initial passwd*> field is also empty.

The issuer could, if he or she choose, specify values/empty fields for all nine number variables. In this case, the bulk file lines shown above would look like:

```
add smith:John Smith:::fs1:a:::::marketing::::::
add jones:Pat Jones:::fs3:c:::::finance::::::
```

The following shows a complete bulk file containing a set of `delete` lines combined with a `savevolume` line. Because *smith*, *pat*, and *rogers* appear before the `savevolume` command and their third field is blank, the volume will be removed. The volume for *terry* will not be removed, but *johnson*'s will be because the `delvolume` in the third field overrides the current default.

```
delete smith:/afs/transarc.com/usr/smith
delete pat:/afs/transarc.com/usr/pat
delete rogers:/afs/transarc.com/usr/rogers
savevolume
delete terry:/afs/transarc.com/usr/terry
delete johnson:/afs/transarc.com/usr/johnson:delvolume
```

The following `exec` example supposes that the bulk file contains a set of `add` lines followed by `delete` lines. The operator places this line between the two sets to flag when the additions are finished and the deletions beginning.

```
exec echo "Additions completed; beginning deletions..."
```

PRIVILEGE REQUIRED

Issuer (or person named by **-admin** argument) must have the privileges necessary to run all of the commands in the bulk file individually.

uss delete — delete user account from system.

uss delete -user <*login name*> **-mountpoint** <*mountpoint for user's volume*> **[-savevolume] [-verbose] [-cell** <*cell name*>**] [-admin** <*administrator to authenticate*>**] [-dryrun] [-skipauth] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

uss d -u <*login name*> **-m** <*mountpoint for user's volume*> **[-sa] [-v] [-c** <*cell name*>**] [-a** <*administrator to authenticate*>**] [-d] [-sk] [-h]**

DESCRIPTION

Deletes an AFS user account from the system by deleting the user's

- Protection Database entry.
- Authentication Database entry.
- Volume and VLDB entry, unless the issuer includes the **-savevolume** flag to prevent the deletion. If the volume is to be deleted, the issuer may first wish to transfer its contents to tape for possible restoration.
- Home directory mount point from the file tree. This happens even if the **-savevolume** flag is provided.

ARGUMENTS

-user	specifies the user name to remove from the Protection and Authentication Databases.
-mountpoint	specifies the pathname to the user's home directory. The volume mounted at that location is deleted from the system. If the issuer does not provide a full pathname, it is interpreted relative to the working directory.
-savevolume	prevents the user's volume and VLDB entry from being deleted from the system.
-verbose	causes the command interpreter to produce on standard output (stdout) a more detailed trace of the actions it is executing. By default, only warnings and error messages appear.
-cell	specifies the cell in which to run the command. See section 7.1 for more details.
-admin	names the user whom the Authentication Servers should authenticate for purposes of deleting the Authentication Database entry. See section 7.1 for more details.
-dryrun	indicates that the command interpreter should not actually execute the command, but should report all the actions it would perform if executing it. See section 7.1 for more details.
-skipauth	directs the command to bypass the Authentication Server, allowing a site using MIT Kerberos to use the command with that form of authentication.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 7.1 for more details.

EXAMPLES

The following removes *smith*'s user account from the *transarc.com* cell, but prevents deletion of the `user.smith` volume.

% uss del smith /afs/transarc.com/usr/smith -save

PRIVILEGE REQUIRED

Issuer (or person named by **-admin** argument) must belong to the *system:administrators* group in the Protection Database, must have the `ADMIN` flag in his or her Authentication Database entry, and must have at least ADMINISTER and DELETE rights in the directory from which the mount point is to be removed. To remove volumes (that is, if the **-savevolume** flag is not used), the issuer must be listed in */usr/afs/etc/UserList*.

uss help — show syntax of specified **uss** command(s) or list functional description for all of them.

uss help [-topic <*help string*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

uss h [-t <*help string*>+**] [-h]**

DESCRIPTION

Displays the first line (name and short description) of every **uss** command's help entry, if no *help string* is provided. For each operation code specified with **-topic**, it outputs the entire help entry. See the **Output** section below.

ARGUMENTS

-topic	specifies the operation code(s) for which syntax is to be provided. If it is omitted, the output instead provides a short description of all **uss** commands.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 7.1 for more details.

OUTPUT

The online help entry for each **uss** command consists of two or three lines.

- The first line names the command and briefly describes what it does.
- If the command has aliases, they will appear on the next line.
- The final line, which begins with "`Usage:`", lists the command's arguments and flags in the prescribed order. Online help entries use the same symbols (brackets, etc.) as the command definitions in this manual. For an explanation of their meaning, see page x of the introductory *About This Manual* chapter.

EXAMPLES

The following displays the online help entry for the **uss bulk** command.

% uss help bulk

```
uss bulk: bulk input mode
Usage: uss bulk -file <bulk input file> [-template <path
of template file>] [-verbose] [-cell <cell name>] [-dryrun]
[-help]
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

uss apropos

Template "A" line — improve authentication/password security for an account.

A <*username*> <*password lifetime*> <*reuse/noreuse*> <*failures*> <*locktime*>

DESCRIPTION

Specifies the following optional authentication/password security improvements for the specified *username*:

- Limiting the valid lifetime of the user's password. The user can normally change his or her password any time prior to the expiration of that password; however, once a password has expired, he or she will not be able to authenticate in his or her account.
- Restricting the reuse of recently used passwords. With this option turned on, a user will be prohibited from using any of his or her most recent twenty passwords.
- Limiting the number of consecutive times that a user can incorrectly enter his or her password during authentication. If the limit is exceeded, the user account will be locked (no one will be able to authenticate with the account) for the specified amount of time. (Only a system administrator will be able to unlock a locked account by using the **kas unlock** command.) For most cells, Transarc Corporation recommends setting the limit on authentication to 5 attempts and the lockout time to 25 minutes.

See the *AFS System Administrator's Guide* for more information on the purpose behind these limits and restrictions.

ARGUMENTS

A should be a capital letter and tells the command interpreter that this is an authentication security improvement instruction.

username specifies the Authentication Database entry to be affected.

password lifetime
defines the number of days for which *name of user*'s password will be valid. Beyond the specified number of days, the password will be invalid (that is, the user will be unable to authenticate with his or her account). The range of valid values is 0 through 254. Use 0 to indicate that the password never expires; use 1 through 254 to specify the number of days that the password will be valid. By default, a user's password will never expire.

reuse/noreuse specifies whether or not the user can reuse a password (during a password change) that he or she has used at any time during the last twenty password changes. The valid values for this field are *yes* (the user can immediately reuse an old password) and *no* (the user can only reuse an old password after 20 password changes). By default, the user can reuse passwords (*yes*).

failures specifies the maximum number of times that a user can consecutively fail to correctly type in his or her password during authentication (for example, using AFS **login** or **klog**), before being locked out of his or her user account for the amount of time specified by the *locktime* field. The range of valid values is 0 through 254. Use 0 to indicate that the user will never be locked out of his or her account based on the number of consecutive failed authentication attempts; this is the default.

locktime specifies the amount of time that a user will be locked out of his or her user account if he or she exceeds the maximum number of failed authentication attempts specified by the *failures* field.

You can specify a time in either hours and minutes (hh:mm) or minutes only (mm). If a 0 is specified, the account will be locked indefinitely (until a system administrator unlocks it using **kas unlock**). If a time is entered (other than 0), it will be rounded up to the next highest increment of 8.5 minutes; so all locked accounts will be locked for some multiple of 8.5 minutes. The maximum finite value is 36 hours; for example, if *99:99* is specified, the locktime will be 36 hours.

EXAMPLES

The follow line sets a password lifetime of 254 days, restricts password reuse, sets the number of failed consecutive authentication attempts to *4* and the corresponding locktime to *45* minutes (which will be rounded up to 51 minutes) for *username* supplied by the **-user** argument of **uss add** command or the `add` line of a **uss bulk** command:

```
A $USER 254 noreuse 4 45
```

Template "D" line — create a directory.

D <*pathname*> <*mode*> <*owner*> <*access list*>

DESCRIPTION

Creates a directory anywhere in the file system. It is used to create a subdirectory in the home directory defined on the preceding "V" line.

The "D" line should appear after the "V" line if any variables in it take their values from the "V" line (notably, `$MTPT`). Any number of "D" lines may appear in a template.

WARNING

Using the "D" line to create directories outside AFS (for example, in the UNIX file system of the workstation where the **uss** command is being issued) introduces two complications:

- The first complication arises if the issuer wishes to set the <*owner*> field to something other than his or her own user name or UNIX UID. The default owner of a UNIX file system directory is the user who created it (in this case, the issuer of the **uss** command). Changing this default, which **uss** will attempt if anything other than the issuer's name appears in the <*owner*> field, requires issuing the **chown** command, which in turn requires being the super-user "root". If the issuer wishes to specify an alternate owner, he or she must be "root" in the UNIX file system and use the **-admin** flag in order to be recognized in AFS as a privileged administrator. See the description of the **-admin** argument for **uss add** and **uss bulk**.
- The second complication is that it is not possible to associate an ACL with a non-AFS directory, so **uss** will always print a warning message about that. It will create the directory nonetheless, and the <*access list*> field must always be provided.

The preferred method for automatic placement of directories on the local disk is the **package** program, which runs as "root".

ARGUMENTS

D	should be a capital letter and tells the command interpreter that this is a directory-creation instruction.
pathname	is the full pathname of the directory.
mode	sets the UNIX mode bits for the directory. Provide a four-digit decimal number in this field, such as *0755* or *0744*. Keep in mind that the "user" `x` bit must be turned on for a directory.
owner	names the directory's owner (in the UNIX file system sense). If the directory is being placed in AFS, the `$UID` variable should appear in this field. If the directory is being placed into the UNIX file system rather than AFS, this field must contain the user name or UNIX UID of the **uss** command's issuer, or the account creation will abort at that point.
access list	defines the access control list for the new home directory. It must be included for non-AFS directories, even though it is ignored in that case. The list may define one or more pairs, each consisting of

- A user name or Protection Database group name.
- The access rights.

separated by a space. See the **fs setacl** command to learn about the access rights available with AFS.

At the least, the new user needs to be given all access rights. This could be achieved with

$USER all

Unlike the "V" line, there are no default pairs on the ACL of a directory created with a "D" line. However, as with the "V" line it is not possible to grant any rights to the issuer of the **uss** command. As the last step in account creation, the **uss** command interpreter automatically deletes that person from any access control lists set during the creation process.

EXAMPLES

The following creates a *public* directory in the user's home directory, makes the user own the directory and gives him/her all access rights.

```
D $MTPT/public 0755 $UID $USER all
```

Template "E" line — create a single-line file.

E <*pathname*> <*mode*> <*owner*> "<*contents*>"

DESCRIPTION

Creates a file anywhere in the file system by echoing what appears in the <*contents*> field into the new file. Its intended use is the creation of files in the new user home directory or its subdirectories.

The "E" line's advantage over the "F" line is that variables may appear in the <*contents*> field. The command interpreter replaces the variables with appropriate values before creating the file. (In contrast, the contents of a file created using the "F" line are the same for everyone.) However, the "E" line restricts the contents of the new file to a single line, since no carriage returns (newline characters) can occur in the <*contents*> field.

An "E" line should appear in the template after the "D" line that creates the directory the file is to reside in. Any number of "E" lines may appear in a template.

WARNING

Using the "E" line to create a file outside AFS (for example, in the UNIX file system of the client machine where the **uss** command is being issued) introduces a potential complication if the issuer wishes to set the <*owner*> field to something other than his or her own user name or UNIX UID. The default owner of a UNIX file system directory is the user who created it (in this case, the issuer of the **uss** command). Changing this default, which **uss** will attempt if anything other than the issuer's name appears in the <*owner*> field, requires issuing the **chown** command, which in turn requires being the super-user "root". If the issuer wishes to specify an alternate owner, he or she must be "root" in the UNIX file system and use the **-admin** flag in order to be recognized in AFS as a privileged administrator. See the description of the **-admin** argument for **uss add** and **uss bulk**.

The preferred method for automatic placement of files on the local disk is the **package** program, which runs as "root".

ARGUMENTS

E	should be a capital letter and tells the command interpreter that this is a file-creation instruction.
pathname	is the full pathname of the file to be created.
mode	sets the UNIX mode bits for the file. Provide a four-digit decimal number in this field, such as *0666* or *0644*.
owner	names the file's owner (in the UNIX file system sense). If the file is being placed in AFS, the `$UID` variable should appear in this field. If the file is being placed into the UNIX file system rather than AFS, this field must contain the user name or UNIX UID of the **uss** command's issuer, or the account creation will abort at that point.

contents defines the one-line string that is to become the contents of the new file. Surround it with double quotes in case it contains spaces. It may contain variables, which the command interpreter will resolve before creating the file, but may not contain carriage returns (newline characters).

EXAMPLES

The following creates an file in the current working directory called ***user_name***.*etcp* containing an entry suitable for incorporating into the cell's global */etc/passwd* file.

```
E  $USER.etcp  0644 root "$USER:X:$UID:10:$NAME:$MTPT:/bin/csh"
```

Template "F" line — create a file by copying a prototype.

F *<pathname> <mode> <owner> <prototype>*

DESCRIPTION

Creates a file anywhere in the file system by copying the contents of an existing "prototype" file into the new file. Its intended use is creation of standard files in the new user's home directory or a subdirectory of it. Include one "F" line for each file to be created.

"F" lines should appear in the template after the "D" lines that define the subdirectories they go in. Any number of "F" lines may appear in a template.

WARNING

Using the "F" line to create a file outside AFS (for example, in the UNIX file system of the client machine where the **uss** command is being issued) introduces a potential complication if the issuer wishes to set the *<owner>* field to something other than his or her own user name or UNIX UID. The default owner of a UNIX file system directory is the user who created it (in this case, the issuer of the **uss** command). Changing this default, which **uss** will attempt if anything other than the issuer's name appears in the *<owner>* field, requires issuing the **chown** command, which in turn requires being the super-user "root". If the issuer wishes to specify an alternate owner, he or she must be "root" in the UNIX file system and use the **-admin** flag in order to be recognized in AFS as a privileged administrator. See the description of the **-admin** argument for **uss add** and **uss bulk**.

The preferred method for automatic placement of files on the local disk is the **package** program, which runs as "root".

ARGUMENTS

F	should be a capital letter and tells the command interpreter that this is a file-creation instruction.
pathname	is the full pathname of the file to be created.
mode	sets the UNIX mode bits for the file. Provide a four-digit decimal number in this field, such as *0666* or *0644*.
owner	names the file's owner (in the UNIX file system sense). If the directory is being placed in AFS, the `$UID` variable should appear in this field. If the directory is being placed into the UNIX file system rather than AFS, this field must contain the user name or UNIX UID of the **uss** command's issuer, or the account creation will abort at that point.
prototype	specifies the complete pathname of the file whose contents are to be copied into the new file. It may reside in either AFS or the local UNIX file system.

EXAMPLES

The following, appropriate for the Transarc Corporation cell, copies a standard *.login* file (kept in */afs/transarc.com/common/uss/skel*) into user home directories. This would appear on a single line in the actual template.

```
F $MTPT/.login 0644 $UID /afs/transarc.com/common/uss/skel/.login
```

Template "G" line — define directory for even distribution of home directories.

G *<directory>*

DESCRIPTION

Defines a directory to be considered as a value for the $AUTO variable. Define a separate "G" line for each *directory* to be considered. All "G" lines, and there may be any number, must appear in the template before any occurrences of the $AUTO variable.

The intended use of the "G" line is to distribute user accounts randomly among several directories, rather than using divisions that reflect real-world divisions (such as departmental affiliation).

Creating multiple "*usr*" directories in this fashion (perhaps numbered as *usr1*, *usr2* and so on) is useful mostly in a very large cell where having all user home directories together in one *usr* directory could cause slow lookup. In such a case, the $AUTO variable goes in the <*mount point*> field in the "V" line. The command interpreter then selects from among the directories defined on "G" lines the one with the fewest entries. See the chapter on **uss** in the *AFS System Administrator's Guide* for more information.

ARGUMENTS

G	should be a capital letter and tells the command interpreter that this line defines a directory to be considered as a value for the $AUTO variable.
directory	defines a pathname. It may be either a complete pathname or only the final element (the directory itself). The choice affects the form of the "V" line <*mount-point*> field's value, as shown in the EXAMPLES section.

EXAMPLES

If the Transarc Corporation cell's administrators decided to distribute user home directories evenly into three directories, they would define three "G" lines:

```
G usr1
G usr2
G usr3
```

and then put

```
/afs/transarc.com/$AUTO/$USER
```

in the "V" line <*mount-point*> field.

Alternatively, they could put the entire pathname for the *usr* directories on the "G" line:

```
G /afs/transarc.com/usr1
G /afs/transarc.com/usr2
G /afs/transarc.com/usr3
```

in which case the "V" line <*mount-point*> field would specify simply

```
$AUTO/$USER
```

Template "L" line — create hard link.

L <*existing-file*> <*link*>

DESCRIPTION

Creates a "hard link" between two files, as achieved by the standard Unix **ln** command. An explanation of links is beyond the scope of this document, but the basic effect is to create a second name for an existing file, so that it can be accessed via either name. It does not create a second copy of the file.

AFS allows hard links only between files that reside in the same directory. This restriction is necessary to eliminate the confusion that would result from associating two potentially different ACLs (those of the two directories) with the same file. However, symbolic links are legal between two files that reside in different directories and even volumes (see the section on the template "S" line).

"L" lines should appear in the template after "D" lines; they should also appear after "F" and "E" lines if either of the files being linked were created on such a line. Any number of "L" lines may appear in a template.

ARGUMENTS

L	should be a capital letter and tells the command interpreter that this instruction creates a hard link.
existing-file	is the complete pathname of the existing file.
link	is the complete pathname of the second name for the file.

EXAMPLES

The following links the file *mail* to the file *mbox* in the home directory.

```
L $MTPT/mbox $MTPT/mail
```

Template "S" line — create symbolic link.

S <*existing-file*> <*link*>

DESCRIPTION

Creates a "symbolic link" between two files, as achieved by the standard Unix **ln -s** command. An explanation of links is beyond the scope of this document, but the basic effect is to create a second name for an existing file, so that it can be accessed via either name. It does not create a second copy of the file.

AFS allows symbolic links to cross mount points (that is, the linked files may reside in different volumes).

"S" lines should appear in the template after "D" lines; they should also appear after "F" and "E" lines if either of the files being linked were created on such a line. Any number of "S" lines may appear in a template.

ARGUMENTS

S	should be a capital letter and tells the command interpreter that this instruction creates a symbolic link.
existing-file	is the complete pathname of the existing file.
link	is the complete pathname of the second name for the file.

EXAMPLES

The following, appropriate for the Transarc Corporation cell, links the file *outgoing* in the *Mail* subdirectory to the file */afs/transarc.com/common/mail/outgoing*.

```
S /afs/transarc.com/common/mail/outgoing $MTPT/Mail/outgoing
```

Template "V" line — create a volume.

V <*volume name*> <*server*> <*partition*> <*quota*> <*mount point*> <*owner*> <*access list*>

DESCRIPTION

Has several effects:

- Creates a volume.
- Creates a VLDB entry for the volume.
- Mounts the volume in the file system at the indicated place in the file tree, effectively creating the user's home directory.
- Sets the owner of the home directory.
- Sets the home directory's access control list.

A template file may contain only one "V" line, and must contain one unless the file is 0-length. No other lines are necessary. (This does not imply that the "V" line must be the first line in the template when there are others. If the `$AUTO` variable appears on the "V" line, then the "G" lines that provide values for it must appear above the "V" line.)

ARGUMENTS

V should be a capital letter and tells the command interpreter that this is a volume-creation instruction.

volume name is the volume's name, under which it will be listed in the VLDB.

By convention, AFS user volume names have a `user.` prefix followed by the user name. This is achieved by specifying

user.$USER

in this field.

server is the file server machine that will house the volume. It is safest to provide a complete Internet-style host name in this field. The interpretation of a shortened form depends on the state of the cell's name server when the **uss add** command is executed.

partition defines which partition on *server* will house the volume.

Acceptable variants of the standard */vicep**x***-style name are:

/vicepa = vicepa = a = 0

/vicepb = vicepb = b = 1

and so on up to

/vicepz = vicepz = z = 25

quota defines the maximum amount of disk space the volume will be able to take up. Specify a number of kilobyte blocks (so a value of *1024* means a megabyte).

mount point defines the directory to be created as a mount point for the volume. The specified directory serves as the root directory for the volume named in the <*volume name*> field.

By convention, AFS cells call home directories by the user's user name, so the `$USER` variable may be part of this field.

Note: This field defines the value of any occurrences of the `$MTPT` variable on subsequent lines in the template.

owner names the owner (in the UNIX file system sense) of the home directory named in the <*mount point*> field. This field should contain the `$UID` variable.

access list defines the access control list for the new home directory. The list may define one or more pairs, each consisting of

- A user name or Protection Database group name.
- The access rights.

separated by a space. See the **fs setacl** command to learn about the access rights available with AFS.

At the least, the new user needs to be given all access rights. This could be achieved with

$USER all

The pairs specified are added to a default pair that grants *system:administrators* the ALL rights. Do not attempt to grant any rights to the issuer of the command. As the last step in account creation, the **uss** command interpreter automatically deletes that person from any access control lists set during the creation process.

EXAMPLES

The following, appropriate for the Transarc Corporation cell, creates a volume called `user.`*user name* on the /vicepa partition of the specified file server machine, assigning it a quota of 3000 kilobyte blocks. The volume is mounted in */afs/transarc.com/usr* as the value of `$USER`. The user owns the home directory and has all access rights to it. This line would appear on a single line in the actual template.

```
V user.$USER $SERVER.transarc.com /vicepa 3000
        /afs/transarc.com/usr/$USER $UID $USER all
```

Template "X" line — execute command.

X "<*command*>"

DESCRIPTION

Executes the indicated command, which may be a standard UNIX or AFS command. The command may include any standard template variables, which the **uss** command interpreter will resolve before passing the command on to the appropriate other command interpreter.

"X" lines should appear in the template after the definition of any elements they affect or variables they refer to. Any number of "X" lines may appear in a template. Each line must be a single line only (cannot contain carriage returns or new line characters within the <*command*> field).

ARGUMENTS

X	should be a capital letter and tells the command interpreter that this instruction executes a command.
command	is the command to be executed. Surround it with double quotes in case it contains spaces. It may contain variables, which the command interpreter will resolve before executing the command, but may not contain carriage returns (newline characters).

EXAMPLES

The following, appropriate for the Transarc Corporation cell, mounts the Backup version of the user's volume at the *OldFiles* subdirectory.

```
X "fs mkm /afs/transarc.com/usr/$USER/OldFiles user.$USER.backup"
```

8. The vos Commands

This chapter defines the **vos** commands that system administrators use to contact the Volume Server and Volume Location (VL) Server. It assumes the reader is familiar with the concepts described in *AFS System Administrator's Guide*.

System administrators use **vos** commands to instruct the Volume Server to create, move, delete, replicate, and backup volumes. The Volume Location (VL) Server automatically records in the Volume Location Database (VLDB) any changes in volume status and location that result from **vos** commands. Operators can use other **vos** commands to obtain information from VLDB records. The **vos** commands also help operators dump copies of volumes to disk and restore them to the file system if necessary.

Note that **vos** commands are "idempotent." This means that if execution of a certain command is interrupted in the middle by a server or process failure, then a subsequent execution of the same command picks up at the interruption point, rather than at the very beginning of the operation. (This does not apply if the issuer explicitly interrupts the operation with **^C** or another interrupt signal. In that case, the volume is left locked and the issuer must unlock it with **vos unlock** before proceeding.) Idempotency implies that before executing a command, the Volume and VL Servers check to see if running it will have any effect (that is, whether the state that will result from running the command already holds). If the desired end-state already holds, there is no need to run the command again, no matter how many times it is subsequently issued. If the end-state does not yet hold, the command should pick up where necessary to achieve it.

Refer to the *Command Summary* at the end of this document for a complete list of **vos** commands and their syntax.

8.1. The Information in VLDB Entries

The Volume Location Database (VLDB), maintained by the Volume Location (VL) Server, contains information about all the volumes in a cell. A separate copy of the VLDB resides on each database server machine. To keep the copies synchronized, the VL Server uses Ubik -- the AFS library of database management facilities.

It is important that the information in the VLDB correspond to the status of the actual volumes on the servers as much of the time as possible. For this reason, any **vos** command that affects volume status also changes the corresponding VLDB entry automatically, as noted in the command descriptions below. (It is possible for the VLDB and servers to disagree if **vos** operations are interrupted before completion; see the **vos syncserv** and **vos syncvldb** commands.)

There is an entry in the VLDB for each ReadWrite volume. The entry also records information about the ReadOnly and Backup version of the volume; ReadOnly and Backup volumes do not have their own VLDB entries. (The one exception to this rule is that a ReadOnly volume may have its own VLDB entry if its ReadWrite source has been removed.)

The following information is available from a VLDB entry:

- The ***volume name*** of the ReadWrite version of the volume. The ReadOnly version automatically has the same name with a `.readonly` extension, the Backup version with a `.backup` extension.
- The ***ReadWrite (RWrite) volume ID number***, if one exists. A volume ID number is simply an identification number guaranteed to be unique within a cell. In almost all cases, the VL Server allocates volume ID numbers automatically, but some commands allow the issuer to assign volume IDs explicitly. Each of the three versions of a volume has a different volume ID number; they often run consecutively, especially if the VL Server assigned them, but this is not a requirement.
- The ***ReadOnly (ROnly) volume ID number***, if one or more exist. All copies of the ReadOnly version share the same ID.
- The ***Backup volume ID number***, if one exists.
- The ***ReleaseClone (RClone) volume ID number***, if one exists. See the description of **vos release** for more about the ReleaseClone.
- A ***site count***, which tells how many ReadWrite and ReadOnly volumes exist. There can be only one ReadWrite volume, but as many as eleven ReadOnly copies may exist.
- One or more ***site definitions***, each of which specifies a ***file server machine name*** and ***partition name***. Site definitions specify the location of the ReadWrite version and each copy of the ReadOnly version (if any). No site definition is necessary for the Backup version because it always resides at the same site as its ReadWrite source. There can be only one ReadWrite definition but up to eleven ReadOnly definitions; the number of site definitions should equal the site count.
- A ***type flag*** associated with each site. The type flag indicates the type of volume residing at the site, and can have one of the following values: `RW` for ReadWrite or `RO` for ReadOnly.

- A ***release flag*** associated with each site. The presence of a release flag indicates that an error occurred as new ReadOnly clones were being distributed to their sites. See the description of the **vos release** command for further explanation.
- An indication if the VLDB entry is locked. Since being unlocked is the default state, there is no explicit indicator if the VLDB entry is unlocked. See the descriptions of the **vos lock** and **vos unlock** commands.

The **vos listvldb** command displays much of the information from the VLDB. The **vos examine** command also displays VLDB information, in combination with volume header information.

8.2. The Information in Volume Headers

While there is only a single VLDB entry for each ReadWrite volume and all of its (ReadOnly and Backup) clones, there is a separate ***volume header*** at the site of each copy of the volume, no matter its version. The volume header is the volume in a sense — it is a data structure that records the physical memory addresses on the partition that store the files in the volume. The volume header binds all the files into a logical unit without requiring that they be stored in contiguous memory blocks. In addition to data location, the volume header stores other information about the volume, some of it redundant with the VLDB, so that the Volume Server can access the information even when the VLDB is unavailable.

The information in the volume header includes the following:

- The ***volume name*** of this copy of the volume, with a `.readonly` or `.backup` extension if appropriate.
- Its ***volume ID number***.
- Its ***type flag*** that identifies the type of volume (`RW` for ReadWrite, `RO` for ReadOnly, `BK` for Backup).
- Its ***size*** in kilobytes (`1024` equals a megabyte).
- Its ***status*** at the site, which is unrelated to the locked/unlocked status of the VLDB entry. There are three possibilities:
  - `On-line` means that the volume is fully accessible through the file system.
  - `needs salvage` means that the volume is probably corrupted. Run the Salvager using the **bos salvage** command.
  - `Off-line` means that the volume is not accessible, but there is no reason to suspect that it is corrupted.
- A ***ReadWrite (RWrite) volume ID number***, if one exists; if no ReadWrite volume exists, a zero is displayed. This is the volume ID of this volume's ReadWrite source. If this volume is the ReadWrite version itself, this ID should match the volume ID number.
- A ***ReadOnly (ROnly) volume ID number***, if one exists; if no ReadOnly volumes exist, a zero is displayed. If this volume is a ReadOnly version itself, this ID should match the volume ID number.

- A ***Backup volume ID number***, if one exists; if no Backup volume exists, a zero is displayed. This is the volume ID of the Backup version of this volume. If this volume is the Backup version itself, this ID should match the volume ID number.
- A ***ReleaseClone (RClone) volume ID number***, if one exists; if no ReleaseClone exists, no RClone label is displayed in the output. This is the volume ID number of the last clone made from this volume's ReadWrite source for the purposes of replication. If a ReleaseClone volume ID is displayed, it indicates that the replication failed.
- A ***MaxQuota***, which is the maximum amount of disk space the ReadWrite version of the volume may occupy. It does not apply sensibly to ReadOnly or Backup volumes, but is reported for convenience anyway.
- Its ***Creation date***, which is the date and time when this copy of the volume was created (for ReadOnly and Backup copies, this means cloned/released). If the volume has been restored with **backup diskrestore**, **backup volrestore**, or **vos restore**, this is the restore time.
- Its ***Last Update***, which is the date and time when the contents of the volume last changed. For ReadOnly and Backup volumes, this should match the creation date, since they are not allowed to change.
- The ***number of times the volume has been accessed*** since the later of
  - 12:00 a.m. on the day the command is issued.
  - The last time the volume changed location.

  An access is defined as a fetch or store operation on any file system object stored in the volume.
- The data necessary to generate two **statistical tables** entitled *Raw Read/Write Stats* and *Writes Affecting Authorship*. These tables provide the following information:
  - The *Raw Read/Write Stats* table contains information on the number of reads (fetches) and writes (stores) made on the specified *volume name or id*. This information is categorized by reads and writes initiated by a client on the same network as the server(Same Network) versus those initiated from a client on a different network than the server different network (Diff Network). Both of these categories are further broken down into all reads and writes versus those initiated by authenticated users and processes.

    Note: Diff Network versus Same Network is determined by comparing the network portion of the IP address based on the class of network (A, B, or C). If the network portion of the IP addresses for the client and server match, it is the same network; otherwise, it is considered a different network.
  - The *Writes Affecting Authorship* table contains information on writes made to files and directories in the specified *volume name or id*. The table categorizes the data to compare writes to files (File Authorship) and writes to directories (Directory Authorship). Both of these categories are further broken down to compare writes made by the owner of the files or directories (Same), and writes made by users/processes that were not the owners of the files or directories (Diff). The time divisions indicate the time between the recorded write and the previous update to the affected file (or directory).

The **vos listvol** command displays information from the volume header. The **vos examine** command also displays information from the volume header, in combination with VLDB information.

8.3. Common Arguments and Flags

Most **vos** commands accept the following optional arguments and flags. They are listed in the command descriptions where they apply, and are described in detail below:

[-cell <*cell name*>]

This argument specifies that the command should be run in the cell specified by *cell name*. The issuer may abbreviate *cell name* to the shortest form that distinguishes it from the other cells listed in */usr/vice/etc/CellServDB* on the client machine on which the command is issued. By default, commands are executed in the local cell, as defined:

- First, by the value of the environment variable AFSCELL.
- Second, in */usr/vice/etc/ThisCell* on the client machine on which the command is issued.

Note: Do not use the **-cell** argument and the **-localauth** flag on the same **vos** command because they are functionally incompatible. The **-cell** argument instructs a **vos** command to execute with respect to a foreign cell; the **-localauth** flag instructs a **vos** command to use the cell name given in */usr/afs/etc/ThisCell*.

[-noauth]

This flag instructs the Volume and/or Volume Location Server not to authenticate the user of the command, and thus establishes an unauthenticated connection between the user and the server (the user is recognized as the unprivileged user *anonymous*). It is useful only when authorization checking is disabled on the file server machine (during the installation of a file server machine or when **bos setauth** has been used during other unusual circumstances). In normal circumstances, the Volume and VL Servers allow only authorized (privileged) users to issue commands that change the status of a volume or VLDB entry, and will refuse to perform such an action even if the **-noauth** flag is used.

[-localauth]

This flag instructs the **vos** command interpreter running on the local machine to construct a server ticket using the server encryption key with the highest key version number in */usr/afs/etc/KeyFile* on the local machine. The command interpreter presents the ticket to the Volume and/or Volume Location Server to use in mutual authentication.

This flag is only useful for commands issued on file server machines, since client workstations do not have a *KeyFile*. It is intended for ***cron***-type processes or jobs included in the machine's */usr/afs/local/BosConfig* file. An example might be a command that automatically runs the **vos backup** command on certain volumes in preparation for archival backups. See the chapter in the *AFS System Administrator's Guide* about backing up the system. The flag can also be used if the user is unable to authenticate but is logged into the local UNIX file system as "root".

Note: Do not use the **-localauth** flag and the **-cell** argument on the same **vos** command because they are functionally incompatible. The **-cell** argument instructs a **vos** command to execute with respect to a foreign cell; the **-localauth** flag instructs a **vos** command to use the cell name given in */usr/afs/etc/ThisCell*.

[-verbose]

This flag tells the Volume Server and/or VL Server to display messages about the operations they are performing while executing the command. Useful mainly for debugging and tracing purposes.

[-help]

This flag has the same function as the **vos help** command: it prints the command's online help message on the screen. No other arguments or flags should be provided at the same time. Even if they are, this flag overrides them, and the only effect of issuing the command is that the help message appears.

8.4. The Privilege Required for vos Commands

Most **vos** commands require privilege; only those that list volume-related information do not. The Volume and Volume Location Servers consider privileged those users listed in the file */usr/afs/etc/UserList* on a file server machines' local disk. This list is maintained on each file server machine's local disk using **bos** commands.

vos addsite — add definition of ReadOnly site to VLDB entry.

vos addsite **-server** <*machine name for new site*>
-partition <*partition name for new site*>
-id <*volume name or ID*> **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-verbose] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

vos ad **-s** <*machine name for new site*> **-p** <*partition name for new site*>
-i <*volume name or ID*> **[-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Defines future sites that will receive copies of the ReadOnly version of a volume, in preparation for actually creating and distributing the ReadOnly version using **vos release**. A site is defined as a certain file server machine and partition.

If defining more than one replication site for a given ReadWrite source, issue this command repeatedly.

WARNING

Do not define more than eleven ReadOnly sites with this command. Each VLDB entry can store up to twelve site entries (the ReadWrite site and up to eleven ReadOnly sites).

ARGUMENTS

-server	names the file server machine where the copy is to reside. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	names the particular partition on that file server machine where the copy is to reside. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-id	specifies either the complete name or volume ID number of the ReadWrite source volume.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following, appropriate in the Transarc Corporation cell, defines a ReadOnly site for the cell's `root.afs` volume.

% vos ad fs7.transarc.com /vicepb root.afs

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos listvol

vos release

vos apropos — show each help entry containing keyword.

vos apropos -topic <*help string*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos ap -t <*help string*> **[-h]**

DESCRIPTION

Displays the first line of the help entry for any **vos** command that has *help string* in its name or short description.

ARGUMENTS

-topic	specifies the keyword string to search for. If it is more than a single word, surround it with double quotes or other delimiters. Type all *help string*s for **vos** commands in all lowercase letters, except the word "*VLDB*."
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

OUTPUT

The first line of a command's online help entry names the command and briefly describes what it does. The **vos apropos** command displays that first line for any **vos** command where *help string* is part of the command name or first line.

To see the remaining lines in a help entry, which provide the command's alias (if any) and syntax, use the **vos help** command.

EXAMPLES

The following lists all **vos** commands that have the word "*lock*" in their operation code or short online description.

```
% vos ap lock
lock: lock VLDB entry for a volume
unlock: release lock on VLDB entry for a volume
unlockvldb: unlock all the locked entries in the VLDB
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

vos help

vos backup — create Backup volume version of one ReadWrite volume.

vos backup -id <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos backup -i <*volume name or ID*> **[-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Clones the indicated ReadWrite volume to create a Backup version. Names the Backup version by adding a `.backup` extension to the ReadWrite source's name. It places the Backup version at the same site as the ReadWrite. If a Backup version already exists, this new clone replaces it.

ARGUMENTS

-id	specifies either the complete name or volume ID number of the ReadWrite source volume.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following creates a Backup version of the volume `user.smith`.

% vos backup user.smith

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos backupsys

vos backupsys — create Backup volume version of all indicated ReadWrite volumes.

vos backupsys [-prefix <*common prefix on volume(s)*>**]**
[-server <*machine name*>**] [-partition** <*partition name*>**]**
[-exclude] [-cell <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

vos backups [-pr <*common prefix on volume(s)*>**] [-s** <*machine name*>**]**
[-pa <*partition name*>**] [-e] [-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Clones each indicated ReadWrite volume to make a Backup version of it. Names each Backup version by adding a `.backup` extension to the name of its ReadWrite source. It places each Backup at the same site as its source. If a Backup version already exists for a volume, the new clone replaces it.

By combining the **-prefix**, **-server**, and **-partition** arguments in different ways, it is possible to create Backup copies of varying numbers of volumes. The possibilities are listed here from most to least inclusive.

To create a Backup version of:

- Every ReadWrite volume in the system, omit all three arguments. This could take a rather long time to execute, depending on the number of volumes.
- Every ReadWrite volume whose name begins with a certain character string (for example, `sys.` or `user.`), regardless of site, use **-prefix**.
- Every ReadWrite volume on a file server machine, specify the file server name with **-server**.
- Every ReadWrite volume that resides on a partition of the same name (for instance, on */vicepa* on any file server machine), specify the partition name with **-partition**.
- Every ReadWrite volume on a certain partition of a file server machine, specify both **-server** and **-partition**.
- Every ReadWrite volume with a specific prefix that resides on a file server machine, combine **-prefix** and **-server**; every ReadWrite volume with a specific prefix that resides on a specific server and partition, combine **-prefix**, **-server**, and **-partition**.
- Every ReadWrite volume that resides on a file server machine (except those with a specific prefix), combine **-prefix** and **-server** and attach the **-exclude** flag.
- Every ReadWrite volume that resides on a specific server and partition (except those with a specific prefix), combine **-prefix**, **-server**, and **-partition** and attach the **-exclude** flag.
- A single ReadWrite volume, give its complete name as **-prefix**. This is actually better done with the **vos backup** command, which employs a more streamlined technique for finding a single volume.

ARGUMENTS

-prefix	specifies a character string of any length. Every volume whose name begins with this exact string will be cloned (subject to modulations from **-server** and **-partition**). Include field separators (such as periods) if appropriate. This argument may be combined with **-server** and/or **-partition**.
-server	names the file server machine where the ReadWrite source volume(s) reside. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter. This argument may be combined with **-prefix** and/or **-partition**.
-partition	names the particular partition where the ReadWrite source volume(s) reside. In addition to the full */vicepx* form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter. This argument may be combined with **-prefix** and/or **-server**.
-exclude	indicates that all volumes except those indicated with the **-prefix** argument are to be backed up. If this flag is used, the **-prefix** argument also must be used.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following creates a Backup version of every volume in the cell's file system whose name begins with "`user`".

% vos backups user

The following, appropriate in the Transarc Corporation cell, creates a Backup version of every volume on the file server machine *fs3.transarc.com*.

% vos backups -s fs3.transarc.com

The following, appropriate in the Transarc Corporation cell, creates a Backup version of every volume on the */vicepc* partition of the file server machine *fs5.transarc.com*.

% vos backups -s fs5.transarc.com -p c

The following, appropriate in the Transarc Corporation cell, creates a Backup version of every volume on the file server machine *db1.transarc.com* whose name begins with "`sys`".

% **vos backups sys db1.transarc.com**

The following, appropriate in the Transarc Corporation cell, creates a Backup version of every volume on the file server machine *db1.transarc.com* with the exception of those whose name begins with "`sys`".

% **vos backups sys db1.transarc.com -e**

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos backup

vos create — create (empty) ReadWrite volume and associated VLDB entry.

vos create -server <*machine name*> **-partition** <*partition name*> **-name** <*volume name*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos c -s <*machine name*> **-p** <*partition name*> **-na** <*volume name*> **[-c** <*cell name*>**] [-no] [-l] [-v] [-h]**

DESCRIPTION

Creates a ReadWrite volume, names it *volume name*, and places it at the site specified by *machine name* and *partition name*. The volume automatically receives a volume ID number, recorded in both the VLDB and the volume header.

If this command succeeds, then the volume is available for use, though it must be mounted in the file system before its contents are accessible. Use the **fs mkmount** command to mount a volume.

This command creates a default ACL associated with the volume's "root directory" (which takes the same name as volume's mount point when the volume is mounted with **fs mkmount**). The ACL grants all seven access rights to *system:administrators*. The volume's space quota is set to 5000 kilobyte blocks by default.

The Volume Location Server also allocates, and records in the VLDB, volume ID numbers for the ReadOnly and Backup versions that may be created later. It does not actually create those types of volumes or place anything at a ReadOnly or Backup site.

ARGUMENTS

-server	names the file server machine on which to create the new ReadWrite volume. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	names the particular partition where the ReadWrite volume is to reside. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-name	specifies a name for the ReadWrite volume, preferably descriptive of its contents. It may be no longer than 22 characters, but may contain upper- and lowercase letters, numbers and punctuation. By convention, periods separate the fields in a name. Do not use the extension *.backup* or *.readonly* on ReadWrite volume names; the Volume Server automatically adds these extensions when creating those volume types.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.

-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following creates the ReadWrite volume `user.pat` on the */vicepf* partition of *fs4.transarc.com*.

% vos c fs4.transarc.com /vicepf user.pat

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

vos delentry — remove specified entry from VLDB.

vos delentry -id <*volume name or ID*>+
[-prefix <*prefix of volume whose VLDB entry is to be deleted*>**]**
[-server <*machine name*>**] [-partition** <*partition name*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos de -i <*volume name or ID*>+
[-pr <*prefix of volume whose VLDB entry is to be deleted*>**]**
[-s <*machine name*>**] [-pa** <*partition name*>**] [-c** <*cell name*>**]**
[-n] [-l] [-v] [-h]

DESCRIPTION

Removes the VLDB entry for each specified volume. A specified volume may be any of the three types (ReadWrite, ReadOnly, or Backup), but the entire entry is removed no matter which type is provided. The command has no effect on the actual volumes on file server machines, if they exist.

This command is useful if the system administrator is certain that a volume removal was not recorded in the VLDB (perhaps the **vos zap** command was used), and does not want to take the time to use **vos syncserv** and **vos syncvldb** to synchronize an entire file server machine.

By using the **-id** argument alone, or combining the **-prefix**, **-server** and **-partition** arguments in different ways, it is possible to remove the VLDB entry for varying numbers of volumes. To remove the VLDB entry for:

- A single volume, provide its complete name or volume ID number using the **-id** argument.
- Every volume whose name begins with a certain character string (for example, `sys.` or `user.`), regardless of site, use **-prefix**.
- Every volume listed as residing on a certain file server machine, specify the file server name with **-server**.
- Every volume listed as residing on a partition of the same name (for instance, on */vicepa* on any file server machine), specify the partition name with **-partition**.
- Every volume on a certain partition of a file server machine, specify both **-server** and **-partition**.
- Every volume with a certain prefix that resides on a file server machine, combine **-prefix** and **-server**. The **-prefix** argument may also be combined with **-partition**, or both **-server** and **-partition**, in this way.

WARNING

This command should not be used as the standard way to remove a volume, as it is likely to put the VLDB out of sync with the volumes on servers. Use **vos remove** instead.

ARGUMENTS

-id	specifies either the complete name or the volume ID number of each volume, which may be ReadWrite, ReadOnly, or Backup, whose VLDB entry is to be removed. The entire entry is removed, no matter which type is specified. Provide this argument or some combination of **-prefix**, **-server**, and **-partition**.
-prefix	specifies a character string of any length. Every VLDB listing a volume whose name begins with this exact string will be removed (subject to modulations from **-server** and **-partition**). Include field separators (such as periods) if appropriate. This argument may be combined with **-server** and/or **-partition**.
-server	names a file server machine. If the VLDB entry mentions this file server machine as the site for a volume, the entry will be removed. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter. This argument may be combined with **-prefix** and/or **-partition**.
-partition	names a partition. If the VLDB entry mentions this partition as the site for a volume, the entry will be removed. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter. This argument may be combined with **-prefix** and/or **-server**.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following removes the VLDB entry for the volume *user.temp*.

% vos del user.temp

The following removes all VLDB entries that describe volumes stored on *fs3.transarc.com* whose names begin with the string "`test`".

% vos del -pr test -s fs3.transarc.com

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos remove

vos remsite

vos syncserv

vos syncvldb

vos zap

vos dump — convert volume into ASCII format and place in a file.

vos dump -id <*volume name or ID*> **-time** <*dump from time*> **[-file** <*arg*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos du -i <*volume name or ID*> **-t** <*dump from time*> **[-f** <*arg*>**] [-c** <*cell name*>**]**
[-n] [-l] [-v] [-h]

DESCRIPTION

Converts the contents of the indicated volume, which may be of any type, into ASCII format. If the **-file** argument is provided, the converted volume contents are placed in it; if it is not provided, the contents are directed to standard output (stdout), from which the issuer may pipe them elsewhere if desired.

It is possible to dump either the entire volume, or only those files in it modified since a specified time (the latter is referred to as a "incremental" dump), by specifying the proper value for **-time**.

Dumping does not affect any status or site flags in the volume header or VLDB. It does, however, make the volume inaccessible during the duration of the dump.

ARGUMENTS

-id specifies either the complete name or volume ID number of the volume, which may be ReadWrite, ReadOnly or Backup.

-time determines whether the dump is full or incremental, and in the latter case, from what date the dump is done. There are three types of legal values:

- *0* creates a full dump.
- *mm/dd/yy* specifies 12:00 a.m. on the indicated date (month/day/year) and creates an incremental dump including only elements with modification time stamps later than this date. Examples : *1/23/90*, *10/7/89*.
- *"mm/dd/yy hh:mm"* specifies a time in "hours:minutes" on the indicated date (month/day/year) and creates an incremental dump including only elements with modification time stamps later than this time. The time should be in 24-hour format (for example, 20:30 is 8:30 p.m.) Surround the entire instance with quotes because it contains a space. Examples : *"1/23/90 22:30"*, *"10/7/89 3:45"*.

-file directs the dump into the indicated file, which is created in the current working directory if no full or relative pathname is provided. If the issuer does not provide this argument, the dump is directed to standard output (stdout).

-cell specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.

-noauth tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.

-verbose tells the Volume and Volume Location Servers to trace their execution of the command. See section 8.3 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following executes a full dump of the volume `user.terry` into the file */afs/transarc.com/common/dumps/terry.dump* in the current working directory.

% vos dump user.terry 0 terry.dump

The following executes an incremental dump of the volume `user.smith` into the file *smith.900131.dump* in the current working directory. Only those files in the volume with modification time stamps later than 6:00 p.m. on 31 February 1990 are included in the dump.

% vos dump user.smith "1/31/90 18:00" smith.090131.dump

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**, and must have WRITE access in the directory where the dump file is to reside.

MORE INFORMATION

vos restore

vos examine — show information from volume header and VLDB about one volume.

vos examine -id <*volume name or ID*> **[-extended] [-cell** <*cell name*>**]**
[-noauth] [-localauth] [-verbose] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

vos e -i <*volume name or ID*> **[-e] [-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

vos volinfo -i <*volume name or ID*> **[-e] [-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Formats and displays information from both the VLDB entry and the volume header for the specified *volume name or ID*, which may be of any of the three types. The **vos listvol** command displays information from the volume header only. The **vos listvldb** command displays information from the VLDB only.

ARGUMENTS

-id	specifies either the complete name or volume ID number of the volume, which may be ReadWrite, ReadOnly, or Backup.
-extended	causes the command to display statistics about files and directories in the specified volume. The statistics include information about the number of reads of a file in a volume, the number of writes to files in the volume, and the number of writes to files and directories in a volume by the owner and by non-owner users and processes.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

OUTPUT

The first seven lines of the output show information from the volume header, and the remaining lines come from the VLDB. The command displays the following information from the volume's volume header:

- The first line of output shows information about the specified *volume name or ID* from the volume header, in the following order:

  - The ***volume name***.
  - Its ***volume ID number***.
  - Its ***type flag*** (`RW` for ReadWrite, `RO` for ReadOnly, `BK` for Backup).
  - Its ***size*** in kilobytes (`1024` equals a megabyte).
  - Its ***status*** at the server. The possible values are `On-line`, `Off-line`, and `needs salvage`, the meanings of which are explained fully in section 8.2.

- The second line identifies the volume's file server machine and partition. This information does not come from either the VLDB or the volume header; it is determined by the command at execution.
- The third line shows the ***ReadWrite*** (RWrite), ***ReadOnly*** (ROnly), ***Backup***, and ReleaseClone (RClone) volume ID numbers associated with this volume. One of these volume IDs should match the volume ID number that appears on the first line. A 0 indicates that no volume exists for the corresponding volume type (for ReadWrite, ReadOnly, and Backup); if no ReleaseClone version exists, the RClone label will not appear in the output.
- The fourth line shows the maximum quota ***MaxQuota*** allotted to the ReadWrite copy of the volume, expressed in kilobyte blocks.
- The fifth line shows the ***Creation date*** and time associated with this volume. If the volume has been restored with **backup diskrestore**, **backup volrestore**, or **vos restore**, this is the restore time.
- The sixth line shows the ***Last Update*** date and time associated with this volume, indicating the last time that the contents of this volume changed. For ReadOnly and Backup volumes, this timestamp should match the ***Creation*** timestamp.
- The seventh line reports the **number of times the volume has been accessed** since the later of the two following times:
  - 12:00 a.m. on the day the command is issued.
  - The last time the volume changed location.

If the command is run with the **-extended** flag, two tables appear next. Entitled *Raw Read/Write Stats* and *Writes Affecting Authorship*, these tables provide the following information:

- The *Raw Read/Write Stats* table contains information on the number of reads (fetches) and writes (stores) made on the specified *volume name or id*.
- The *Writes Affecting Authorship* table contains information on writes made to files and directories in the specified *volume name or id*.

Following a blank line, information from the VLDB entry appears:

- If the VLDB entry is locked, a `LOCKED` indicator appears alone on a line at the top of this part of the output.
- The ***volume ID numbers*** allocated to the ReadWrite (RWrite), the Backup, any ReadOnly (ROnly), and the ReleaseClone (RClone) versions of the volume are listed, if they exist. If the ReleaseClone ID appears here, then one or more `Old release` and `New release` flags should appear on the site definition lines discussed below.

- The next line provides a **site count** indicating the number of separate sites (partition/file server machine pairs) where copies of the volume exist. It should match the number of site definition lines that follow.
- Each of the remaining **site definition** lines show where the ReadWrite copy (and by implication, the Backup copy) and each ReadOnly copy of the volume resides, specified by ***server***, ***partition***, and ***type*** (`RW` or `RO`). The presence of `New release` or `Old release` flags on these lines indicates a failed release; they should appear only if a ReleaseClone volume ID number also appears above (see page 8-45).

Note: Sections 8-2 and 8-3 provide more complete explanations of these fields.

EXAMPLES

The following shows output for Transarc Corporation's volume called `usr` with two ReadOnly replication sites (this volume is mounted at */afs/transarc.com/usr*). For the sake of illustration, the output shows the volume as locked. If it weren't, the `LOCKED` line would not appear.

% **vos examine usr**

```
usr                          536870981 RW   3459 K On-line
     fs2.transarc.com /vicepb
     RWrite 5360870981   ROnly 536870982   Backup 536870983
     MaxQuota      40000 K
     Creation    Mon Jun 12 15:22:06 1989
     Last Update Fri Jun 16 09:34:35 1989
     5719 accesses in the past day (i.e., vnode references)
        LOCKED
     RWrite: 5360870981   ROnly: 536870982   Backup: 536870983
     number of sites -> 3
        server fs1.transarc.com partition /vicepa RO Site
        server fs3.transarc.com partition /vicepa RO Site
        server fs2.transarc.com partition /vicepb RW Site
```

The following shows the output for the volume `user.terry` using the **-extended** flag. The volume has no ReadOnly replication sites.

% vos e user.terry -extended

```
user.terry         354287190 RW    2302 K used 119 files On-line
    fs4.transarc.com /vicepc
    RWrite 354287190 ROnly          0 Backup 354287192
    MaxQuota       5000 K
    Creation    Wed Nov 25 17:38:57 1992
    Last Update Tue Dec 15 10:46:20 1992
    598 accesses in the past day (i.e., vnode references)

                    Raw Read/Write Stats
            |-----------------------------------------|
            |    Same Network     |    Diff Network   |
            |----------|----------|----------|--------|
            |  Total   |   Auth   |   Total  |  Auth  |
            |----------|----------|----------|--------|
Reads       |       55 |       55 |       38 |      38 |
Writes      |       95 |       95 |        0 |       0 |
            |-----------------------------------------|

                    Writes Affecting Authorship
            |-----------------------------------------|
            |   File Authorship   | Directory Authorship|
            |----------|----------|----------|----------|
            |   Same   |   Diff   |    Same  |   Diff   |
            |----------|----------|----------|----------|
0-60 sec    |       38 |        0 |       21 |        1 |
1-10 min    |        2 |        0 |        7 |        0 |
10min-1hr   |        0 |        0 |        1 |        0 |
1hr-1day    |        1 |        0 |        5 |        1 |
1day-1wk    |        0 |        0 |        0 |        0 |
> 1wk       |        0 |        0 |        0 |        0 |
            |-----------------------------------------|

    RWrite: 354287190    Backup: 354287192
    number of sites -> 1
       server fs4.transarc.com partition /vicepc RW Site
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

vos diskrestore

vos listvldb

vos listvol

vos restore

vos volrestore

vos help — show syntax of specified **vos** command(s) or list functional description for all of them.

vos help [-topic <*help string*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos h [-t <*help string*>+**] [-h]**

DESCRIPTION

Displays the first line (name and short description) of every **vos** command's help entry, if no *help string* is provided. For each operation code specified with **-topic**, it outputs the entire help entry. See the **Output** section below.

ARGUMENTS

-topic	specifies the operation code(s) for which syntax is to be provided. If it is omitted, the output instead provides a short description of all **vos** commands.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

OUTPUT

The online help entry for each **vos** command consists of two or three lines:

- The first line names the command and briefly describes what it does.
- If the command has aliases, they will appear on the next line.
- The final line, which begins with "`Usage:`", lists the command's arguments and flags in the prescribed order. Online help entries use the same symbols (brackets, etc.) as the command definitions in this manual. For an explanation of their meaning, see page x of the introductory *About This Manual* chapter.

EXAMPLES

The following displays the online help entry for the **vos create** command.

% **vos help create**

```
vos create: create a new volume
Usage: vos create -server <machine name> -partition <partition name>
-name <volume name> [-cell <cell name>] [-noauth] [-localauth]
[-verbose] [-help]
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

vos apropos

vos listpart — show all AFS partitions on specified file server machine.

vos listpart -server <*machine name*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos listp -s <*machine name*> **[-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Lists all of the valid AFS partitions on the indicated file server machine, without consulting the VLDB. The **vos partinfo** command reports the size of a partition and the available space on that partition.

ARGUMENTS

-server	names the file server machine for which to list the partitions. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

OUTPUT

The output consists of a list of partition names of the form `/vicep`*x*, following the header:

```
The partitions on the server are:
```

The last line of the output reports the total number of partitions.

EXAMPLES

The following lists the partitions on *fs1.transarc.com*:

% vos listpart fs1.transarc.com

```
The partitions on the server are:
    /vicepa     /vicepb     /vicepc     /vicepd

Total:  4
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

vos partinfo

vos listvldb — show information from the VLDB.

vos listvldb [-name <*volume name or ID*>**] [-server** <*machine name*>**]**
[-partition <*partition name*>**] [-locked] [-quiet]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos listvl [-na <*volume name or ID*>**] [-s** <*machine name*>**] [-p** <*partition name*>**]**
[-lock] [-q] [-c <*cell name*>**] [-no] [-loca] [-v] [-h]**

DESCRIPTION

Formats and displays information from the VLDB entry for each volume specified. The output depends on the combination of arguments and/or flags that are supplied when the command is issued. To display:

- Every entry in the VLDB, provide no arguments. This can take a very long time, depending on the number of entries.
- Every VLDB entry that mentions a certain file server machine as the site of a ReadWrite or ReadOnly version of a volume, specify the machine's name with **-server**.
- Every VLDB entry that mentions a certain partition on any file server machine as the site of a ReadWrite or ReadOnly version of a volume, specify the partition name with **-partition**.
- Every VLDB entry that mentions a certain partition on a certain file server machine as the site of a ReadWrite or ReadOnly version of a volume, combine **-server** and **-partition**.
- A single VLDB entry, specify a volume name or ID number with **-name**.
- The VLDB entry only for the volumes with locked VLDB entries found at a certain site, combine the **-locked** flag with any of arguments that define sites.

ARGUMENTS

-name	specifies either the complete name or volume ID number of a volume of any of the three types.
-server	names a file server machine. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter. This argument may be combined with **-partition**.
-partition	names a partition. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter. This argument may be combined with **-server**.
-locked	indicates that the output will show only locked VLDB entries. May be used alone, or combined with one or more of **-name**, **-server** or **-partition**.
-quiet	suppresses the lines that appear at the beginning and end of the output, which specify the machine/partition the listing is for and the total number of entries reported.

-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

OUTPUT

The VLDB entry for each volume includes the following information:

- If the VLDB entry is locked, a `LOCKED` indicator appears alone on a line at the top of this part of the output.
- The **volume name**.
- The ***volume ID numbers*** allocated to the ReadWrite (RWrite), the Backup, any ReadOnly (ROnly), and the ReleaseClone (RClone) versions of the volume are listed, if they exist. If the ReleaseClone ID appears here, then one or more `Old release` and `New release` flags should appear on the site definition lines discussed below.
- The next line provides a **site count** indicating the number of separate sites (partition/file server machine pairs) where copies of the volume exist. It should match the number of site definition lines that follow.
- Each of the remaining **site definition** lines show where the ReadWrite copy (and by implication, the Backup copy) and each ReadOnly copy of the volume resides, specified by ***server***, ***partition***, and ***type*** (`RW` or `RO`). The presence of `New release` or `Old release` flags on these lines indicates a failed release; they should appear only if a ReleaseClone volume ID number also appears above (see page 8-45).

Note: Section 8-2 provide more complete explanations of these fields.

If the output includes more than one VLDB entry, the very last line of output shows the total number of entries reported. This line does not appear if the **-quiet** flag is used.

EXAMPLES

The following displays VLDB information for the Transarc Corporation volume called `usr`, which has two ReadOnly replication sites:

% vos listvldb usr

```
usr
 RWrite: 5360870981   ROnly: 536870982   Backup: 536870983
 number of sites -> 3
    server fs1.transarc.com partition /vicepa RO Site
    server fs3.transarc.com partition /vicepa RO Site
    server fs2.transarc.com partition /vicepb RW Site
```

The following shows entries for two of the volumes that reside on *fs4.transarc.com*. The first displayed VLDB entry is currently locked. The total number of entries for the server is 508.

% vos listvldb -s fs4.transarc.com

```
VLDB entries for server fs4.transarc.com
    .       .           .        .
    .       .           .        .
        LOCKED
user.smith
 RWrite: 278541326   ROnly: 278541327   Backup: 278542328
 number of sites -> 1
   server fs4.transarc.com partition /vicepg RW Site

user.terry
 RWrite 354287190   ROnly 354287191   Backup 354287192
 number of sites -> 1
   server fs4.transarc.com partition /vicepc RW Site
   .       .           .        .
   .       .           .        .

Total entries: 508
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

vos examine

vos listvol

vos lock

vos unlock

vos unlockvldb

vos listvol — show information from volume header(s).

vos listvol -server <*machine name*> **[-partition** <*partition name*>**]**
[-fast] [-long] [-quiet] [-extended] [-cell <*cell name*>**]**
[-noauth] [-localauth] [-verbose] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

vos listvo -s <*machine name*> **[-p** <*partition name*>**] [-f] [-lon] [-q] [-e]**
[-c <*cell name*>**] [-n] [-loc] [-v] [-h]**

DESCRIPTION

Using information from the volume header, this command formats and displays the following information for each specified volume: volume name, volume ID, volume type, size, and status at the server. The actual information displayed depends on the combination of arguments supplied when the command is issued. To display volume header information for:

- Every volume on a file server machine, specify the machine's name with **-server**.
- Every volume at a particular site, provide the file server machine name with **-server** and the partition with **-partition**.

ARGUMENTS

-server	names a file server machine. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter. This argument may be combined with **-partition**.
-partition	names a partition. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter. The **-server** argument must be provided along with this argument.
-fast	indicates that the output should display only the volume ID numbers of volumes at the indicated location. Do not use this flag with the **-extended** flag.
-long	indicates that the output should include the IDs of all volumes associated with the volume, the ReadWrite volume's quota, creation date and update date. See the **Output** section below.
-quiet	suppresses the lines that appear at the beginning and end of the output, which total the number of volumes listed.
-extended	causes the command to display statistics about files and directories in the specified volume. The statistics include information about the number of reads of a file in a volume, the number of writes to files in the volume, and the number of writes to files and directories in a volume by the owner and by non-owner users and processes. Do not use this flag with the **-fast** flag.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.

-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

OUTPUT

The output of this command is arranged in alphabetical order by volume name and provides the following information for each volume by default:

- The ***volume name***.
- Its ***volume ID number***.
- Its ***type flag*** (`RW` for ReadWrite, `RO` for ReadOnly, `BK` for Backup).
- Its ***size*** in kilobytes (`1000` equates to a megabyte).
- Its ***status*** at the server. The possible values are `On-line`, `Off-line`, and `needs salvage`, the meanings of which are explained fully in section 8.2.

The first line of output specifies the number of volumes in the listing; the last line specifies the number of volumes that are on-line, off-line, and busy. These lines do not appear if the **-quiet** flag is used.

If the **-fast** flag is added, the output lists only the volume ID number of each volume, arranged in increasing numerical order. The last line of, specifying the number of on-line, off-line, and busy volumes, is omitted.

If the **-long** flag is added, the output includes all of the information in the default listing plus the following for each volume:

- The volume's file server machine and partition. This information does not come from either the VLDB or the volume header; it is determined by the command at execution.
- The ***ReadWrite*** (RWrite), ***ReadOnly*** (ROnly), ***Backup***, and ***ReleaseClone*** (RClone) volume ID numbers associated with this volume. One of these volume IDs should match the volume ID number that appears on the first line. A 0 indicates that no volume exists for the corresponding volume type (for ReadWrite, ReadOnly, and Backup); if no ReleaseClone version exists, no RClone label appears in the output.
- The maximum quota (**MaxQuota**) allotted to the ReadWrite copy of the volume, expressed in kilobyte blocks.
- The ***Creation date*** and time associated with this volume. If the volume has been restored with **backup diskrestore**, **backup volrestore**, or **vos restore**, this is the restore time.
- The ***Last Update*** date and time associated with this volume, indicating the last time that the contents of this volume changed. For ReadOnly and Backup volumes, this date and time should match the ***Creation date*** and time.

- The **number of times the volume has been accessed** since the later of the two following times:
  - 12:00 a.m. on the day the command is issued.
  - The last time the volume changed location.

If the command is run with the **-extended** flag, two tables appear. Entitled *Raw Read/Write Stats* and *Writes Affecting Authorship*, these tables provide the following information:

- The *Raw Read/Write Stats* table contains information on the number of reads (fetches) and writes (stores) made on the specified *volume name or id*.
- The *Writes Affecting Authorship* table contains information on writes made to files and directories in the specified *volume name or id*.

Note: Section 8-3 provides more complete explanations of these fields.

EXAMPLES

The following shows selected parts of the default listing for the */vicepb* partition on *fs2.transarc.com*:

% vos listvol fs2.transarc.com b

```
Total number of volumes on server fs2.transarc.com
                          partition /vicepb : 66
sys                  1969534847 RW       1582 K On-line
sys.backup           1969535105 BK       1582 K On-line
      .                    .     .         .   .    .
      .                    .     .         .   .    .
user.pat             1969534536 RW      17518 K On-line
user.pat.backup      1969534538 BK      17537 K On-line
Total volumes onLine 66 ; Total volumes offLine 0 ;  Total busy 0
```

The following shows the output when the **-fast** flag is added:

% vos listvol fs2.transarc.com b -fast

```
Total number of volumes on server fs2.transarc.com
                            partition /vicepb : 66
 1969516782
 1969516784
     .
     .
 1969535796
```

The following shows two volumes from the output that appears when the **-long** flag is added:

% vos listvol fs2.transarc.com b -long

```
Total number of volumes on server fs2.transarc.com
                                    partition /vicepb: 66
      .                    .      .          .   .    .
      .                    .      .          .   .    .

user.pat              1969534536 RW      17518 K On-line
     fs2.transarc.com /vicepb
     RWrite 1969534536 ROnly 0        Backup 1969534538
     MaxQuota      20000 K
     Creation    Mon Jun 12 09:02:25 1989
     Last Update Thu Jan  4 17:39:34 1990
     1573 accesses in the past day (i.e., vnode references)

user.pat.backup       1969534538 BK      17537 K On-line
     fs2.transarc.com /vicepb
     RWrite 1969534536 ROnly 0        Backup 1969534538
     MaxQuota      20000 K
     Creation    Fri Jan  5 06:37:59 1990
     Last Update Fri Jan  5 06:37:59 1990
     0 accesses in the past day (i.e., vnode references)

       .                    .      .          .   .    .
       .                    .      .          .   .    .

Total volumes onLine 66 ; Total volumes offLine 0 ;
                                              Total busy 0
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

backup diskrestore

backup volrestore

vos examine

vos listvldb

vos restore

vos lock — lock a volume entry in the VLDB.

vos lock -id <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos lo -i <*volume name or ID*> **[-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Locks the VLDB entry for the indicated volume, blocking any operation that requires a write to that entry. Note that this blocks operations on all of the volumes associated with that entry, not just the one specified as **-id**.

The **vos unlock** command unlocks a single VLDB entry. The **vos unlockvldb** command unlocks several entries at once, up to every entry in the VLDB.

WARNING

This command should not be used in normal circumstances. It is useful if the system administrator wishes to guarantee that no one else manipulates the volume until the lock is released, and there is reason to believe that locking will not happen automatically.

ARGUMENTS

-id	specifies either the complete name or volume ID number of a volume of the any of the three types.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following locks the VLDB entry for *user.terry*.

% vos lo user.terry

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos unlock

vos unlockvldb

vos move — move volume to specified other site.

vos move -id <*volume name or ID*> **-fromserver** <*machine name on source*>
-frompartition <*partition name on source*>
-toserver <*machine name on destination*>
-topartition <*partition name on destination*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos m -i <*volume name or ID*> **-froms** <*machine name on source*>
-fromp <*partition name on source*> **-tos** <*machine name on destination*>
-top <*partition name on destination*> **[-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Moves the indicated ReadWrite volume from its current site (specified with **-fromserver** and **-frompartition**) to the destination site (specified with **-toserver** and **-topartition**). This command automatically removes the Backup copy from the current site, if it exists. To create a new Backup at the destination site, use **vos backup**.

It is not possible to actually move a ReadOnly or Backup volume. For ReadOnly volumes, the corresponding action is to create a new ReadOnly site (with **vos addsite** and **vos release**) and then remove an existing one (with **vos remove**). For Backup volumes, the corresponding action is to move its ReadWrite source and then issue **vos backup**.

When you attempt to move a volume using this command, a check will be made to ensure that the *partition name on destination* is not full and that it contains enough space to house the new volume. If you attempt to move a volume and the command detects that there is not enough space, the following message will appear:

`vos: no space on target partition` *dest part* `to move volume` *volume*

Where *dest part* is the *partition name on destination* and *volume* is *volume name or ID*.

WARNINGS

Unless you have a very good reason for doing so, you should not interrupt a **vos move** command in progress. Interrupting a move can cause one or more of the following problems:

- You may end up with two versions of the volume, one version at the old site and one version at the new site. (If this happens, you should retain the version identified by the VLDB and remove the other version using **vos zap**.)
- A backup version of the volume being moved may remain in place at the old site. (If this happens, you should remove the backup using **vos zap**.)
- The volume you are moving may be off line. (If this happens, run **bos salvage** to bring it back on line.)

If you attempt to interrupt a move (using a **ctrl-C**), the following message will appear:

```
SIGINT handler: vos move operation in progress
WARNING: may leave AFS storage and metadata in indeterminate state
enter second control-c to exit
```

To continue the interrupt, press **ctrl-C** a second time; press any other key to abort the interrupt.

ARGUMENTS

-id	specifies either the complete name or volume ID number of a ReadWrite volume.
-fromserver	names the file server machine where the volume currently resides. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-frompartition	names the partition where the volume currently resides. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-toserver	names the file server machine to which the volume should move. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-topartition	names the partition to which the volume should move. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following moves the volume *user.smith* from */vicepb* on *fs3.transarc.com* to */vicepg* on *fs7.transarc.com*.

% vos move user.smith fs3.transarc.com b fs7.transarc.com g

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on both **-fromserver** and **-toserver**.

MORE INFORMATION

vos addsite

vos listvol

vos remove

vos backup

vos release

vos partinfo — show available and total space on specified partition(s).

vos partinfo -server <*machine name*> **[-partition** <*partition name*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos p -s <*machine name*> **[-p** <*partition name*>**] [-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Lists the amount of space available and total size on either all of the partitions on the indicated file server machine (if **-partition** is omitted) or the specified partition on that file server machine. The VLDB is not consulted.

ARGUMENTS

-server	names the file server machine for which to inspect the partitions. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	names the particular partition to inspect. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

OUTPUT

Note: The total partition size reported in this command may not agree with the same figure in the output of the standard UNIX **df** command. The **df** total size includes some reserved space that does not show up in this report, and so is likely to be about 10% larger.

The output reports the amount of space available and total space for each specified partition.

EXAMPLES

The following lists all the partitions on *fs2.transarc.com*.

% vos p fs2.transarc.com

```
Free space on partition /vicepa: 27301 K blocks out of total 549197
Free space on partition /vicepb: 13646 K blocks out of total 69194
Free space on partition /vicepc: 31798 K blocks out of total 320315
Free space on partition /vicepd: 33302 K blocks out of total 494954
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

vos listpart

vos release — place ReadOnly versions of a ReadOnly volume at the sites indicated in the VLDB entry.

vos release -id <*volume name or ID*> **[-f] [-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos rel -i <*volume name or ID*> **[-f] [-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Clones the indicated ReadWrite volume and places a copy of the clone at each ReadOnly site indicated in the VLDB entry. Each copy is named after the ReadWrite source, with the addition of a `.readonly` extension. The issuer must already have defined the ReadOnly sites using **vos addsite**.

The **vos release** command is no more difficult to use than any other **vos** command, but exactly what happens internally during its execution is somewhat complicated. The complexity is necessary in order to ensure that all copies of the volume's ReadOnly version match both the ReadWrite source and each other. If all the ReadOnly copies are not the same, then users might see different data depending on which copy of the volume they happen to access--obviously not a satisfactory situation. To make sure that all ReadOnly copies match each other and the ReadWrite source, releases should be "all-or-nothing"--either all ReadOnly sites receive the new clone, or all sites keep the ReadOnly version they currently have.

The "all-or-nothing" requirement has two main implications that affect the issuer:

- He or she needs to be alert for error messages that indicate an unsuccessful release, and/or check the VLDB entry to make sure certain error-signaling flags are not present
- He or she needs to decide whether to use the **-f** flag

The following two subsections discuss these two implications.

Flags that indicate a failed release

If the **vos release** command fails before the ReadOnly volume is in place at every defined site, an error message will specify which sites did not receive the ReadOnly volume. To give the issuer a backup method for determining if a release has completed (and for its own internal use), the Volume Server and VL Server set various flags while executing the following **vos release** steps. The presence of some of these flags after an apparent completion signals failure. After determining the cause of the failure, the issuer should attempt to eliminate the cause and then continue to issue the **vos release** commands as many times as necessary to make sure the release completes successfully.

The steps during a release, and the flags set, are:

1. Before cloning begins, the VL Server sets the site flag for the present ReadOnly entries in the VLDB to `Old release`.

2. The VL Server sets the site flag for the ReadWrite source to `New release`.
3. The Volume Server clones the ReadWrite source, if required. It assigns the clone a temporary volume ID number and the VL Server puts that number in the ***releaseClone*** field in the source's VLDB entry. (The discussion below on the use of the **-f** flag describes when the Volume Server should make a new clone, and how it uses the releaseClone ID in case a release is not completely successful.)
4. The Volume Server distributes a copy of the ReleaseClone to each ReadOnly site previously defined in the VLDB (using **vos addsite**). The VL Server changes the site flag for each ReadOnly site from `Old release` to `New release` as soon as the site successfully receives the new clone.
5. When all the ReadOnly copies are successfully released, the VL Server clears all the `New release` site flags, leaving that part of the site flag field empty. Because it is no longer needed, the Volume Server deletes the ReleaseClone from the system and its ID from the VLDB.

The presence of `New release` and/or `Old release` site flags in the VLDB after the "completion" of a release indicates that it was not successful. As mentioned above, an unsuccessful release makes it possible for Cache Managers to access different versions of a volume, depending on which File Server they contact. To prevent this possibility, the system administrator should take whatever steps are necessary to make the release successful.

Using the* -f *flag

If the issuer wants to make sure that the Volume Server releases a brand new clone to the ReadOnly sites, he or she can include the **-f** flag. The flag "forces" the Volume Server to make a new clone of the ReadWrite source volume and distribute it to all the possible ReadOnly sites.

If the issuer does not include the **-f** flag, the Volume Server's course of action depends on whether all of the ReadOnly sites already have identical copies of the volume:

- If all the sites currently have the same copy, the Volume Server infers that the previous **vos release** command must have completed successfully, and that no new ReadOnly sites have been defined since. Assuming that the issuer wants to release a brand new clone, the Volume Server makes one and distributes it to all the defined sites.
- If all the sites do not have the same copy, then the Volume Server concludes that either the previous **vos release** command failed in the middle or else a system administrator has defined a new ReadOnly site since the last release. (A possible reason for failure of a release is that one or more sites were inaccessible at the time.) The Volume Server does not need to make an entirely new clone, however. Instead, it distributes the clone with the *releaseClone* volume ID to the sites that do not have it yet (step 3 above defines the ReleaseClone). If this release also fails, the operator needs to issue more **vos release** commands until the same copy exists at all sites.

WARNING

There should be a maximum of twelve sites defined in the VLDB entry for the volume (one ReadWrite site and up to eleven ReadOnly sites).

ARGUMENTS

-id	specifies either the complete name or volume ID number of a ReadWrite volume.
-f	determines whether the Volume Server makes a new clone before distributing it to the ReadOnly sites, in interaction with the state of the ReadOnly copies already at sites. The section entitled ***Using the -f flag*** describes all the issues.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following clones the ReadWrite volume *usr* and releases it to the ReadOnly sites defined in its VLDB entry.

```
% vos release usr
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos addsite

vos listvol

vos syncserv

vos syncvldb

vos remove — remove specified volume from a site.

vos remove **-server** <*machine name*> **-partition** <*partition name*>
-id <*volume name or ID*> **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-verbose] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

vos remo **-s** <*machine name*> **-p** <*partition name*> **-i** <*volume name or ID*> **[-c** <*cell name*>**]**
[-n] [-l] [-v] [-h]

DESCRIPTION

Removes the indicated volume from the site specified by **-server** and **-partition**. The VLDB records the removal, with the precise results described below.

Use this command to remove any of the three types of volumes; the exact effect differs for the types:

- Removing a ReadWrite volume automatically removes its associated Backup copy as well. The site information for both is removed from the VLDB entry, but their volume IDs are still recorded. ReadOnly sites, if any, are not affected. The whole VLDB entry is removed if there are no ReadOnly sites.
- Removing ReadOnly copies must be done on a site-by-site basis, as specified by **-server** and **-partition**. The specified site is erased from the VLDB entry. Removing a Backup copy does not erase its volume ID.

This command is appropriate in almost all circumstances. Other commands (**vos delentry**, **vos remsite** and **vos zap**) are available for removing volumes or VLDB entries, but by definition they can put the volumes and VLDB out of sync. Use them only in the special circumstances mentioned in their command descriptions.

ARGUMENTS

-server names the file server machine where the volume resides. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.

-partition names the particular partition to inspect. In addition to the full */vicepx* form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.

-id specifies either the complete name or volume ID number of a volume of the any of the three types. When removing a ReadOnly or Backup volume, avoid accidentally removing the ReadWrite source by:

either specifying the appropriate extension (`.readonly` or `.backup`) if providing a name

or using the appropriate volume ID number if providing a number.

-cell specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.

-noauth tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.

-localauth constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.

-verbose tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following removes the ReadWrite volume *user.terry* and its Backup version, if any, from their site (*/vicepc* on *fs3.transarc.com*).

% vos remo fs3.transarc.com c user.terry

The following removes the ReadOnly volume *root.afs.readonly* from one of its sites, */vicepa* on *fs1.transarc.com*.

% vos remo fs1.transarc.com a root.afs.readonly

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos delentry

vos remsite

vos zap

vos remsite — remove ReadOnly site definition from a VLDB entry.

vos remsite -server <*machine name*> **-partition** <*partition name*> **-id** <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos rems -s <*machine name*> **-p** <*partition name*> **-i** <*volume name or ID*> **[-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Removes the ReadOnly replication site specified by **-machine** and **-partition** from the VLDB entry for the indicated volume, which is ReadWrite.

This command is useful for removing ReadOnly sites that were mistakenly created with the **vos addsite** command, before the **vos release** command actually releases them. If a ReadOnly copy already exists at the site, it is not affected. However, if this ReadOnly site was the last site housing any version of the volume, then the entire VLDB entry will disappear, even if a copy of the ReadOnly version still actually exists at the site. The discrepancy will not be noticed until someone runs the **vos syncserv** and **vos syncvldb** commands.

WARNING

This command should not be used as the standard way to remove a ReadOnly volume, as it can put the VLDB out of sync with the volumes on servers. Use **vos remove** instead.

ARGUMENTS

-server	specifies the file server machine part of the site definition to be removed. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	specifies the partition name part of the site definition to be removed. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-id	specifies either the complete name or volume ID number of a ReadWrite volume.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.

-verbose tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following removes the mistakenly defined ReadOnly site *viceph* on *fs5.transarc.com* from the VLDB entry for volume `root.cell`.

% vos remsite fs5.transarc.com h root.cell

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos delentry

vos remove

vos zap

vos rename — rename a volume.

vos rename -oldname <*old volume name*> **-newname** <*new volume name*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos ren -o <*old volume name*> **-ne** <*new volume name*> **[-c** <*cell name*>**]**
[-no] [-l] [-v] [-h]

DESCRIPTION

Changes the name of the ReadWrite volume specified with **-oldname** to the name specified with **-newname**. The names of the ReadWrite's ReadOnly copies and Backup copy, if any, change automatically to match.

After issuing this command, remember to correct any mount points that refer to the old volume name, by removing the old mount point with **fs rmmount** and creating a new one with **fs mkmount**.

ARGUMENTS

-oldname	is the current name of the ReadWrite volume.
-newname	is the desired new name for the volume.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following changes the mistaken volume name `sun4_411.afsws` to the correct alternative `sun4_411.usr.afsws`.

% vos ren sun4_411.afsws sun4_411.usr.afsws

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

vos restore — convert ASCII file into proper volume format and place it into the file system.

vos restore -server <*machine name*> **-partition** <*partition name*> **-name** <*name of volume to be restored*> **[-file** <*dump file*>**] [-id** <*volume ID*>**] [-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos res -s <*machine name*> **-p** <*partition name*> **-na** <*name of volume to be restored*> **[-f** <*dump file*>**] [-i** <*volume ID*>**] [-c** <*cell name*>**] [-no] [-l] [-v] [-h]**

DESCRIPTION

Converts the dump file indicated with **-file** from ASCII into the volume format appropriate for the machine specified as **-server**, and restores it as a ReadWrite volume to the site specified by **-server** and **-partition**. It assigns it the name indicated with **-name**, and resets the volume's "creation date," stored in the volume header, to match the restore time. If **-file** is not provided, the issuer must provide input from standard input (stdin), presumably through a pipe.

The issuer may optionally specify a volume ID number as **-id**, but this is not generally recommended; the Volume Server allocates one otherwise. If used, it should be a number that the issuer knows is available and has a particular reason for using.

If the name indicated is the name of an existing (ReadWrite) volume, the contents of the dump file will overwrite the existing volume, after the command interpreter prompts for confirmation that this is acceptable. The volume retains its current volume ID number.

Note: If this command is used to restore a volume whose name ends with a **.restored** extension, the extension is not stripped from the volume name prior to restoration. The command will strip **.backup** and **.readonly** extensions from a volume name, where applicable.

ARGUMENTS

-server	names the file server machine onto which to restore the volume. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	names the partition onto which to restore the volume. In addition to the full */vicepx* form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-name	specifies the name under which to restore the volume. If a volume already exists under that name, it will be overwritten after the issuer is prompted for confirmation.
-file	names the dump file the Volume Server should restore. If the issuer provides a full or relative pathname, the Volume Server looks for the file there; otherwise it looks in

	the current working directory. If the issuer does not provide this argument, he or she must provide input through standard input (stdin).
-id	specifies the volume ID number to assign to the restored volume.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following restores the contents of the dump file */afs/transarc.com/common/dumps/terry.dump* to the */vicepc* partition on *fs3.transarc.com*. The restored volume is called `user.terry`.

```
% cd /afs/transarc.com/common/dumps
% vos res terry.dump fs3.transarc.com c user.terry
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos dump

vos examine

vos listvol

vos status — report activity of Volume Server.

vos status -server <*machine name*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos st -s <*machine name*> **[-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Reports on what the Volume Server on a certain file server machine is doing at the moment the command is issued. If there is no activity, the message "No active transactions on *machine name*" is returned. This command is useful mainly if there is concern that the Volume Server is not performing requested actions.

ARGUMENTS

-server	names a file server machine. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

OUTPUT

There are two possible types of output.

The following message indicates that the Volume Server is not currently performing any actions.

```
No active transactions on machine name
```

The other possible output is a set of information which is probably more useful to programmers than to system administrators. A full understanding of all the fields requires familiarity with the code for the Volume Server, as many of the fields report ID numbers and flag values that the Volume Server sets for internal use.

Among the fields of possible interest to an administrator are:

- `created` on the first line, which indicates the time at which this transaction started
- `attachFlags` on the second line, where a value of `offline` indicates that the volume is not available for other read or write operations during this transaction
- `volume` on the third line, which specifies the affected volume's ID number
- `partition` on the third line, which indicates where the affected volume resides (at the beginning of the transaction if this is a move)
- `procedure` on the third line, which indicates the internal subprocedure being executed

A fourth line may appear during certain transactions, and includes the following fields:

- `packetRead` tracks whether information is being read into the volume. Its absolute value is not informative, but the way it changes shows whether the **vos restore** command is executing properly. As **vos status** is issued repeatedly during a restore, `readNext` should increase monotonically to indicate that information is being read into the volume.
- `packetSend` tracks whether information is being sent out of the volume. Its absolute value is not informative, but the way it changes shows whether the **vos dump** command is executing properly. As **vos status** is issued repeatedly during a dump, `transmitNext` should increase monotonically to indicate that information is being transferred from the volume into the dump file.

The `lastReceiveTime` and `lastSendTime` are for internal use.

EXAMPLES

The following illustrates the kind of output that might be seen if the Volume Server on *fs1.transarc.com* is executing a dump when this command is issued.

% vos status fs1.transarc.com

```
-------------------------------------------
transaction: 575  created: Tue Jan 2 8:34:56 1990
attachFlags: offline
volume: 536871080 partition: /vicepb procedure: Dump
packetRead: 2 lastReceiveTime: 113313 packetSend: 24588
    lastSendTime: 113317
-------------------------------------------
```

PRIVILEGE REQUIRED

None.

vos syncserv — synchronize VLDB entries that mention a given site with volume headers.

vos syncserv -server <*machine name*> **[-partition** <*partition name*>**] [-force] [-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos syncs -s <*machine name*> **[-p** <*partition name*>**] [-f] [-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Finds and inspects the VLDB entries for volumes (ReadWrite, ReadOnly, and Backup) residing on the file server machine specified by **-server** and, optionally, the partition specified by **-partition**. It checks that everything in the VLDB entry is correct, including the reported sites of all copies (even though that requires looking at volumes on servers other than **-server**). It can make several changes:

- If a volume header is marked `Off-line`, but the corresponding VLDB entry is normal, then the volume is brought on-line. (Note that the `Off-line` flag does not indicate the volume is corrupted.)
- If there are two copies of a given volume at different sites, both marked `Off-line`, the newer volume is brought on-line; if the **-force** flag was specified, the older volume is immediately deleted from its site and the VLDB; otherwise, the command issuer is prompted to confirm or override deletion of the older volume. (This situation could arise if the execution of a **vos move** command were interrupted).
- If a volume recorded in the VLDB does not exist at the indicated site, then the VLDB entry is changed.

Run this command, preferably on all file server machines, after **vos syncvldb** has been run on all file server machines.

ARGUMENTS

-server	names the file server machine for which to check entries in the VLDB. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	names the partition for which to check VLDB entries. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-force	directs the command to automatically remove redundant volumes; if this flag is not specified, the command issuer will be prompted to confirm or override any removals.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.

-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following alters the VLDB entries of volumes whose site definitions mention *fs3.transarc.com* to match the volume header. The **-force** flag is used to automatically remove any unnecessary volumes.

% vos syncserv fs3.transarc.com -f

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos move

vos syncvldb

vos syncvldb — synchronize VLDB entry with volume headers at given site.

vos syncvldb -server <*machine name*> **[-partition** <*partition name*>**] [-force] [-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos syncv -s <*machine name*> **[-p** <*partition name*>**] [-f] [-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Inspects the volumes housed on the file server machine specified by **-server** and, optionally, the partition specified by **-partition**. It checks that the VLDB records every volume whose volume header is marked `On-line`. In case of discrepancies, the command alters VLDB entries to reflect the true status of volumes on **-server**. It also performs the following tasks if necessary:

- If it finds a Backup volume whose ReadWrite source no longer exists at the same site, it removes the Backup immediately (if the command is run with the **-force** flag) or it prompts the user to remove or not remove the Backup volume (if the command is run without the **-force** flag). The issuer will have to issue the **vos backup** command to create a new Backup if desired.
- The VL Server keeps track of the next available volume ID number with a counter. This command changes the counter to the highest volume ID number found.
- If it encounters multiple ReadOnly copies of a volume at the same site, or multiple Backup copies, it removes all but the newest one from that site (if the command is run with the **-force** flag) or it prompts the user to remove or not remove all but the newest one (if the command is run with the **-force** flag).

After running this command, execute **vos syncserv**, preferably on all file server machines in the cell.

ARGUMENTS

-server	names the file server machine from which to compare volumes to VLDB entries. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	names the partition from which to compare volumes to VLDB entries. In addition to the full */vicepx* form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-force	directs the command to remove unnecessary volumes without prompting the command issuer to confirm or override the removal.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.

-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following makes sure that the VLDB matches the volume headers found at sites on *fs4.transarc.com*.

```
% vos syncvldb fs4.transarc.com
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos backup

vos syncserv

vos unlock — unlock an entry in the VLDB.

vos unlock -id <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos unlock -i <*volume name or ID*> **[-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Releases the lock on the VLDB entry for the indicated volume.

WARNING

This command should not be used under normal circumstances.

It is useful if the VLDB entry is locked but there is no reason to suspect inconsistency within the volume or between it and the VLDB. Note that it is possible to list information from locked VLDB entries, even though they cannot be manipulated in other ways.

The **vos unlockvldb** unlocks several VLDB entries at once, or even the entire VLDB. The **vos lock** command locks a VLDB entry so that no one else may perform an action that would require writing the VLDB.

ARGUMENTS

-id	specifies either the complete name or volume ID number of a volume of any of the three types.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following unlocks the VLDB entry for `user.terry`.

% vos unlock user.terry

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos lock

vos unlockvldb

vos unlockvldb — unlock all specified locked entries in the VLDB.

vos unlockvldb [-server <*machine name*>**] [-partition** <*partition name*>**] [-cell** <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

vos unlockv [-s <*machine name*>**] [-p** <*partition name*>**] [-c** <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Releases the lock on the VLDB entries indicated by the combination of arguments provided. To unlock:

- All entries in the VLDB, provide no arguments.
- All entries that mention a file server machine in a site definition, provide its name with **-server**.
- All entries that mention a partition on any file server machine in a site definition, provide the partition name with **-partition**.
- All entries that mention a specific site, provide both **-server** and **-partition**.
- A single volume, use the **vos unlock** command instead.

WARNING

This command should not be issued under normal circumstances.

It is useful if VLDB entries for volumes at a certain site are locked but there is no reason to suspect inconsistency within the volume or between it and the VLDB. Note that it is possible to list information from locked VLDB entries, even though they cannot be manipulated in other ways.

The **vos lock** command locks a VLDB entry so that no one else may perform an action that would require writing the VLDB.

ARGUMENTS

-server	names the file server machine for which to unlock VLDB entries. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	names the partition for which to unlock VLDB entries. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.

-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following unlocks all locked entries in the VLDB.

% vos unlockvldb

The following unlocks all locked VLDB entries that mention */vicepa* in a site definition.

% vos unlockvldb -p a

The following unlocks all locked VLDB entries that refer to volumes on the */vicepc* partition of *fs3.transarc.com*.

% vos unlockvldb fs3.transarc.com c

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos lock

vos unlock

vos zap — remove a volume from its site without writing to the VLDB.

vos zap -server <*machine name*> **-partition** <*partition name*>
-id <*volume ID*> **[-force] [-backup] [-cell** <*cell name*>**]**
[-noauth] [-localauth] [-verbose] [-help]

ACCEPTABLE ABBREVIATIONS/ALIASES

vos z -s <*machine name*> **-p** <*partition name*> **-i** <*volume ID*>
[-f] [-b] [-c <*cell name*>**] [-n] [-l] [-v] [-h]**

DESCRIPTION

Removes the volume with the specified *volume ID* from the site defined by **-server** and **-partition**, without attempting to change the corresponding VLDB entry. If the removal of a volume could result in incorrect data in the VLDB, a warning message is displayed.

The **-force** flag removes a volume even if it cannot be "attached" (brought online), which can happen either because the volume is extremely damaged or because the Salvager functioned abnormally. Without this flag, this command cannot remove volumes that are not attachable. See also the WARNINGS section.

If you also want to remove the Backup version of a specified ReadWrite volume, include the **-backup** flag.

WARNINGS

This command should not be used as the standard way to remove a volume, as it is likely to put the VLDB out of sync with the volumes on servers. Use **vos remove** instead.

This command is useful in situations where it is important to delete the volume, but for some reason the VLDB is unreachable (the VL Server may be down). The issuer may remove the VLDB entry later with **vos remove** or **vos delentry**, or it will be removed automatically when someone runs **vos syncserv** and **vos syncvldb**.

To remove a ReadOnly site defined in the VLDB by mistake, before a copy actually exists at the site, use **vos remsite**. To remove an entire VLDB entry without affecting volumes at their sites, use **vos delentry**.

Do not use the **-force** flag if the volume is online, but only when attempts to remove the volume with **vos remove** or **vos zap** have failed and/or the volume definitely cannot be attached. After using **-force**, make sure that the volume's VLDB entry is also removed (issue **vos delentry** if necessary).

Adding the **-force** flag makes the command take considerably longer—about as long as a salvage of the relevant partition—since the Volume Server examines all inodes on the partition for traces of the volume.

ARGUMENTS

-server	names the file server machine from which to remove the volume. Abbreviated forms of machine names may be allowed depending on the naming service available at the time the command is issued; see page xvi in the introductory *About This Manual* chapter.
-partition	names the partition from which to remove the volume. In addition to the full */vicep**x*** form of a partition name, three shorter forms are acceptable; see page xvi in the introductory *About This Manual* chapter.
-id	specifies the volume ID number of the volume to remove, which may be of any of the three types. The volume name is not acceptable.
-force	tells the Volume Server to delete the volume even though it cannot be attached (brought online). Use only after failed attempts to remove the volume with **vos remove** and **vos zap** without this flag.
-backup	tells the Volume Server to also remove the Backup version of the ReadWrite volume associated with *volume ID*. This flag should only be included if you are removing a ReadWrite volume.
-cell	specifies the cell in which to run the command. Do not use this argument with the **-localauth** flag. See section 8.3 for more details.
-noauth	tells the Volume and Volume Location Servers to assign the identity *anonymous* to the issuer. See section 8.3 for more details.
-localauth	constructs a server ticket using a key from */usr/afs/etc/KeyFile*. Do not use this flag with the **-cell** argument. See section 8.3 for more details.
-verbose	tells the Volume and Volume Location Servers to report on what they are doing as they execute the command. See section 8.3 for more details.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one. See section 8.3 for more details.

EXAMPLES

The following removes the volume with volume ID 536870988 from */vicepf* of *fs6.transarc.com*, without noting the change in the VLDB.

% vos zap fs6.transarc.com f 536870988

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* on **-server**.

MORE INFORMATION

vos delentry

vos remove

vos remsite

9. Miscellaneous AFS Commands

This chapter describes AFS commands that do not belong to a suite. The commands are

afsd
afsmonitor
bosserver
budb_convert
buserver
butc
dkload
fileserver
fms
kaserver
kdb
klog
knfs
kpasswd
package_test
pagsh
ptserver
runntp
salvager
scout
tokens
translate_et
unlog
up
upclient
upserver
vldb_convert
vlserver
volserver

afsd — initialize Cache Manager and start related daemons.

afsd [-blocks <*1024 byte blocks in cache*>**] [-files** <*files in cache*>**]**
[-rootvol <*name of AFS root volume*>**] [-stat** <*number of stat entries*>**]**
[-memcache] [-cachedir <*cache directory*>**]**
[-mountdir <*mount location*>**] [-daemons** <*number of daemons to use*>**]**
[-nosettime] [-verbose] [-rmtsys] [-debug]
[-chunksize <*log(2) of chunk size*>**] [-dcache** <*number of dcache entries*>**]**
[-volumes <*number of volume entries*>**]**
[-biods <*number of bkg I/O daemons (aix vm)*>**]**
[-prealloc <*number of 'small' preallocated blocks*>**]**
[-confdir <*configuration directory*> **[-logfile** <*Place to keep the CM log*>**]**
[-waitclose] [-help]

ACCEPTABLE ABBREVIATIONS

This command does not use the syntax conventions of the AFS command suites so the command name and its switches and flags must be typed in full.

DESCRIPTION

Initializes the Cache Manager on an AFS client machine by transferring AFS-related configuration information into kernel memory and starting several daemons. More specifically, **afsd**

- Sets a field in kernel memory that defines which cell the machine belongs to. Some Cache Manager-internal operations and system calls consult this field to learn which cell to execute in. (The AFS command interpreters refer to */usr/vice/etc/ThisCell* instead.) This information is transferred into the kernel from the file */usr/vice/etc/ThisCell* and cannot be changed until **afsd** runs again.
- Places in kernel memory the names and Internet addresses of the database server machines in the local cell and (optionally) foreign cells. The appearance of a cell's database server machines in this list enables the Cache Manager to contact them and to access files in the cell. Omission of a cell from this list, or incorrect information about its database server machines, prevents the Cache Manager from accessing files in it.

  This information is transferred into the kernel from the file */usr/vice/etc/CellServDB*. After initialization, use the **fs newcell** command to change the kernel-resident list without having to reboot.
- Defines the directory in the machine's file name space at which the AFS file tree is mounted. The first field in the file */usr/vice/etc/cacheinfo* is the source for the default directory name. The standard value is */afs*. Use the **-mountdir** argument to override the value from *cacheinfo*.
- Defines which volume corresponds to the root of the AFS file tree. The default is `root.afs`; use the **-rootvol** argument to override it. Note that although the volume name should be given in the "base" (ReadWrite) form, the Cache Manager retains its bias for accessing the ReadOnly version of the volume—in the default case, `root.afs.readonly`—if it is available.

- Determines whether the cache is on disk or in machine memory. The default is to use a disk cache. If the **-memcache** argument is provided, space is allocated in machine memory for caching, and disk space is not used even if the machine has a disk.

- Defines the name of the local disk directory devoted to caching, when **-memcache** is not used. If necessary, **afsd** creates the directory (as long as its parent directory exists). It does not remove the directory that formerly served this function, if one exists.

  The second field in the file */usr/vice/etc/cacheinfo* is the source for this name, and the standard value is */usr/vice/cache*. Use the **-cachedir** argument to override the value from *cacheinfo*.

- Sets the size of the cache, for both disk and memory caches. The value should not exceed 90% to 95% of the disk space available on the cache partition (with a disk cache) or of the machine's memory (with a memory cache). This is because the cache implementation itself requires a small amount of disk or machine memory.

  The **afsd** program consults the third field in the file */usr/vice/etc/cacheinfo* to learn the default cache size in kilobyte blocks. For either a disk or memory cache, use the **-blocks** argument to override the value from *cacheinfo*.

  For a disk cache, the value defined in *cacheinfo* or with **-blocks** is an absolute upper limit on cache size; values provided for other arguments cannot result in a larger cache.

  For a memory cache, the issuer can override the *cacheinfo* value by providing either

  – **-dcache** and **-chunksize**, to set both the number of dcache entries and the chunk size (see below for definition of these parameters). In this case, **afsd** derives cache size by multiplying the two values. Using this combination is not recommended, as it requires the issuer to perform the calculation beforehand to determine the resulting cache size.

  – **-dcache** by itself. In this case, **afsd** derives cache size by multiplying **-dcache** by the default chunk size of 8 kilobytes. Using this argument is not recommended, as it requires the issuer to perform the calculation beforehand to determine the resulting cache size.

  After initialization, use **fs setcachesize** to change the size of a disk cache without having to reboot; the value set with that command is overridden the next time **afsd** runs. The **fs setcachesize** command does not work for memory caches. Instead, the machine must be rebooted.

- Sets the size of each "chunk" of data in the cache, and by implication the amount of data that the Cache Manager requests at a time from the File Server (how much data per "fetch" RPC, since AFS uses partial file transfer).

  For a disk cache, each chunk is called a "V file", so this parameter sets the maximum size of each V file; the default is 64 kilobytes. See below for more on V files. For a memory cache, each chunk is a collection of memory blocks allocated together, so this sets the size of each collection; the default is 8 kilobytes.

  For both types of cache, use the **-chunksize** argument to change the default chunk size. To guarantee proper chunk sizes, the integer provided is used as an exponent on the number *2*; see the ARGUMENTS section for details. For a memory cache, if total cache size divided by chunk size leaves a remainder, **afsd** rounds down the number of dcache entries resulting in a slightly smaller cache (see below for more on dcache entries).

- Sets the number of chunks into which the cache is divided. With a disk cache, each chunk is called a "V file." With a memory cache, each chunk is a set of memory blocks allocated together.

  For a disk cache, the number of chunks is computed by the **afsd** command as the greater of one of the following (unless overridden by the **-files** argument):

  – 100.

  – cachesize/chunksize * 1.5.

  – cachesize/10,000.

  For a memory cache, the number of chunks is equal to the cachesize divided by the chunk size.

- Sets the number of "dcache entries" allocated in machine memory for storing information about the chunks in the cache.

  For a disk cache, the file */usr/vice/cache/CacheItems* contains one entry for each V file. By default, one half the number of these entries (but never more that 2000) are duplicated as dcache entries in machine memory for quicker access.

  For a memory cache, there is no *CacheItems* file so all information about cache chunks must be in memory as dcache entries. Thus, there is no default number of dcache entries for a memory cache; instead, **afsd** derives it by dividing cache size by chunk size.

  Use the **-dcache** argument to set the number of dcache entries. This is not recommended for either type of cache. Increasing the number of dcache entries for a disk cache may improve performance marginally (because more entries are retrieved from memory rather than from disk), but this is not generally necessary. Using this argument is not recommended for a memory cache because it requires the issuer to calculate cache size by multiplying this value by chunk size

  Note: Explicitly using either the **-dcache** argument or one of the Cache Manager configuration scripts available for the **afsd** command, precludes the Cache Manager from calculating a default value for **-dcache**. In addition, the limit of 2000 dcache entries does not apply if either of these two methods is used.

- Sets the number of "stat" entries available in machine memory for caching status information about cached AFS files. The default is 300; use the **-stat** argument to override the default.

- Randomly selects a file server machine in the local cell as the source for the "correct" time. Every five minutes thereafter, the local clock is adjusted (if necessary) to match the file server machine’s clock.

  Use the **-nosettime** flag to prevent **afsd** from selecting a time standard. This is recommended only on file server machines that are also acting as clients. File server machines maintain the correct time using the Network Time Protocol Daemon instead.

In addition to setting cache configuration parameters, **afsd** starts up the following three types of daemons. (On most system types, these daemons appear as nameless entries in the output of the **ps** command.)

- A "callback" daemon that handles callbacks. It also responds to the File Server’s periodic probes, which check that the client machine is still alive.

- A "maintenance" daemon that
  - Collects garbage.
  - Synchronizes files.
  - Probes the *fileserver* process on file server machines every few minutes.
  - Refreshes information from ReadOnly volumes once per hour.
  - Does delayed writes for NFS clients if the machine is running the NFS/AFS Translator.
  - Synchronizes the machine's clock with the chosen file server machine's clock.
- "Background" daemons that improve performance by pre-fetching files and performing background (delayed) writes of saved data into AFS.

  The default number of background daemons is 2, usually enough to handle up to 5 simultaneous users of the machine. Use the **-daemons** argument to increase the number of background daemons, if the machine serves more users. No more than 6 background daemons should ever be necessary.

  Background daemons dedicated to performing I/O operations are required on machines running a version of AIX with VM integration (system type rs_aix32). Known as I/O daemons or VM daemons, these daemons perform tasks such as moving data from disk to memory, and vice versa. By default, 5 VM daemons run by default if both the **-daemons** and **biods** arguments are omitted. Use the **-biods** argument to specify the number of these background daemons; however, this argument should only be used on machines with system type rs_aix32 that are integrated with VM.

The default number of daemons is four (one callback daemon, one maintenance daemon, and two background daemons). The issuer can alter only the number of background daemons; **afsd** always initializes one callback daemon and one maintenance daemon.

AFS includes three configuration scripts that can be used to modify some Cache Manager parameters on a client machine that uses a disk cache. Named *rc.afsd.small*, *rc.afsd.med*, and *rc.afsd.large*, the configuration scripts specify suitable, predefined values for the **afsd** command's **-stat**, **-dcache**, **-daemons**, and **-volumes** arguments. They define increasingly greater values for these arguments according to the configuration and usage patterns of the client machine on which the **afsd** command is run. Refer to the *AFS System Administrator's Guide* for more information about the scripts and how to use them.

ARGUMENTS

-blocks specifies the number of kilobyte blocks to be made available for caching in the machine's cache directory (for a disk cache) or memory (for a memory cache), overriding the default defined in the third field of */usr/vice/etc/cacheinfo*. It should not exceed 90% to 95% of the actual space available. If using a memory cache, do not combine this argument with **-dcache**, since doing so could result in a chunk size that was not an exponent of 2.

-files specifies the number of V files to be created in the cache directory for a disk cache, overriding the default that is calculated by the command based on the size of the cache

and the chunk size. Each V file can accommodate a "chunk" of data, which for a disk cache is 64 kilobytes by default. Do not combine this argument with **-memcache**.

-rootvol names the ReadWrite volume corresponding to the root directory for the AFS file tree (which is usually */afs*). This value overrides the default of `root.afs`.

-stat specifies the number of entries in the machine's memory for recording status information about the AFS files in the cache. This value overrides the default of 300.

-memcache causes **afsd** to initialize a memory cache rather than a disk cache. Do not combine this flag with **-files**.

-cachedir names the local disk directory to be used as the cache. This value overrides the default defined in the second field of */usr/vice/etc/cacheinfo* (typically, */usr/vice/cache*).

-mountdir names the local disk directory on which to mount the AFS file tree. This value overrides the default defined in the first field of */usr/vice/etc/cacheinfo* (typically, */afs*). If */afs* is not used, the machine cannot access the AFS global name space.

-daemons specifies the number of "background" daemons to run on the machine. These daemons improve efficiency by doing pre-fetching and background writing of saved data. This value overrides the default of 2, which is adequate for a machine serving up to five users. It does not change the number of "callback" or "maintenance" daemons, which is always one each. Do not specify a value of less than 2 or more than 6.

Note: On an *rs_aix32* system type (IBM RS/6000 machines running AIX 3.2) integrated with VM, the **-daemons** argument also affects the number of VM (or I/O) daemons that run. The number of VM daemons is calculated as twice the number of background daemons specified with the **-daemons** argument (though, this calculation can be overridden by including the **-biods** argument). If both the **-daemons** and **-biods** arguments are omitted, 5 VM daemons run by default.

-nosettime prevents the machine from selecting at random a local file server machine to act as a source for the "correct" time. If this flag is omitted, the machine selects a file server machine as a time standard, and every five minutes thereafter adjusts its clock to avoid drifting from the standard.

-verbose causes **afsd** to produce a more detailed trace of its activities than the default one. The trace displays on standard out (stdout) unless it is piped into a file.

-rmtsys initializes an additional "remote-system" daemon to execute AFS-specific system calls on behalf of NFS client machines. This flag is necessary only if the machine is an NFS/AFS translator machine, and if users on its NFS clients want to execute AFS commands.

-debug causes **afsd** to produce a highly detailed trace of its activities, potentially useful to a developer for debugging purposes. The trace goes to standard output (stdout) by default.

-chunksize sets the size of each cache chunk. The integer provided, which should be between *0* and *30*, is used as an exponent on the number 2. It overrides the default of 16 for a disk cache (2^{16} is 64 kilobytes) and 13 for a memory cache (2^{13} is 8 kilobytes). A value of *0* or less, or greater than *30*, sets chunk size to the appropriate default. Values less than *10* (which sets chunk size to a kilobyte, 2^{10}) are not recommended. Combining this argument with **-dcache** is not recommended because it requires that the issuer calculate the cache size that results.

-dcache sets the number of "dcache entries" in memory, which are used to store information about cache chunks. For a disk cache, this overrides the default of one-half of the number of *files in cache*. For a memory cache, this argument effectively sets the number of cache

chunks. Use of this argument is not recommended for a memory cache, because it requires the issuer to calculate the resulting total cache size (derived by multiplying this value by chunk size). Do not combine this argument with **-blocks**, since doing so could result in a chunk size that is not an exponent of 2.

-volumes specifies the number of volumes about which location information is to be cached in memory. By default, location information on 50 volumes is stored in memory.

-biods specifies the number of I/O background daemons dedicated to performing I/O operations on a machine running a version of AIX with VM integration (system type *rs_aix32*). I/O daemons, or VM daemons, on an AIX machine perform operations such as moving data from disk to memory, and vice versa. Use this argument only on machines with system type *rs_aix32* that are integrated with VM.

-prealloc specifies the number of blocks of memory that the Cache Manager is to preallocate for its own internal use. By default, this number is 400, which should be enough; on heavily used machines, especially those running AIX 3.2, additional blocks of memory may be needed. However, additional preallocation is unnecessary in most cases. Do not arbitrarily use this argument to increase the number of preallocated blocks.

-confdir directs the Cache Manager to use the specified configuration directory to obtain required configuration files (that is, *cacheinfo*, *ThisCell*, and *CellServDB*). By default, the Cache Manager looks for configuration files in */usr/vice/etc*.

-logfile directs the Cache Manager to use the specified directory to store the *AFSLog* file. By default, the Cache Manager stores this file in */usr/vice/etc*. This argument can also be used to specify a different name for the log file.

Note: Debugging of the Cache Manager is enabled or disabled with the **fs debug** command. Each time debugging is enabled, the log file is cleared. To truncate the file, turn debugging off and then turn it back on. Never removing the *AFSLog* file while debugging is on, as this can cause the Cache Manger to "panic."

-waitclose forces the Cache Manager to complete the transfer of a closed file to the file server before returning control to the application invoking the close. Without this flag, the Cache Manager may return control to a closing application before the final chunk of a file is completely transferred to the file server. Using this flag further reduces the amount of changed user data that can be lost during a server crash; however, it extracts a substantial performance penalty – users in particular will notice an increase in the amount of time it takes to save (close) a file.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

This command is normally included in an initialization file such as */etc/rc*, rather than typed at the command shell prompt. For most disk caches, the appropriate form is

```
/usr/vice/etc/afsd
```

The following is appropriate when enabling a machine to act as an NFS/AFS Translator machine serving more than five users.

```
/usr/vice/etc/afsd -daemons 4 -rmtsys
```

The following initializes a memory cache and sets chunk size to 16 kilobytes (2^{14}).

```
/usr/vice/etc/afsd -memcache -chunksize 14
```

PRIVILEGE REQUIRED

Issuer must be logged into the machine's UNIX file system as "root."

afsmonitor — monitor File Server and/or Cache Manager processes.

afsmonitor [initcmd] [-config <*configuration file*>**]**
[-frequency <*poll frequency, in seconds*>**]**
[-output <*storage file name*>**] [-detailed]**
[-debug <*turn debugging output on to the named file*>**]**
[-fshosts <*list of file servers to monitor*>$^{+}$**]**
[-cmhosts <*list of cache managers to monitor*>$^{+}$**]**
[-buffers <*number of buffer slots*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

afsmonitor [-co <*configuration file*>**] [-fr** <*poll frequency, in seconds*>**]**
[-o <*storage file name*>**] [-det] [-deb** <*turn debugging output on to the named file*>**]**
[-fs <*list of file servers to monitor*>$^{+}$**] [-cm** <*list of cache managers to monitor*>$^{+}$**]**
[-b <*number of buffer slots*>**] [-h]**

DESCRIPTION

Allows you to monitor the status and performance of specified File Server and Cache Manager processes by gathering statistical information on those processes. Using **afsmonitor**, you can

- Monitor any number of File Server and Cache Manager processes on any number of machines (in both local and foreign cells) from a single location.
- Set threshold values for any monitored statistic. When the value of a statistic exceeds the threshold, **afsmonitor** highlights it to draw your attention. Threshold levels can be set globally and customized for specific processes.
- Set up programs/scripts to run automatically when a statistic exceeds its threshold.

You can configure the AFSMonitor by creating a simple ASCII file and specifying that file upon startup with the **-config** argument. With a configuration file, you can

- Specify the File Servers and/or Cache Managers to be monitored.
- Specify the statistics to display. By default, all of the statistics that are gathered are displayed; there are 271 statistics for the File Servers screen and 570 statistics for the Cache Managers screen. See the section on **afsmonitor** in the *AFS System Administrator's Guide* for a complete listing of the available statistics.
- Assign threshold values to statistics and a script or program to execute if a threshold is exceeded. By default, no threshold values are defined and no scripts or programs are executed.

Each line in an **afsmonitor** configuration file must begin with one of the following words/symbol and use the specified format:

- **cm** <*host_name*>

This line indicates that the Cache Manager process on *host_name* is to be monitored. It will be added to the list of such processes and will appear on the System Overview and Cache Managers screens in the order specified in this file.

- **fs** <*host_name*>

  This line indicates that the File Server on *host_name* is to be monitored. It will be added to the list of such processes and will appear on the System Overview and File Servers screens in the order specified in this file.

- **thresh** <**fs**|**cm**> <*field_name*> <*thresh_val*> [<*cmd_to_execute*>] [<arg_1>] . . . [<arg_n>]

  This line assigns the threshold value <*thresh_val*> to <*field_name*> (that is, a statistic field) for either File Servers or Cache Managers (<**fs**|**cm**>). If the <*cmd_to_execute*> option appears, the string comprising this option will be executed each time the corresponding threshold value is exceeded; though it is not executed if both the new value and the previous value are above the threshold.

  When the command is executed, the following variables will be passed to the command as arguments:

  <*host_name*> <**fs**|**cm**> <*field_name*> <*threshold_val*> <*actual_val*> [<arg_1>] . . . [<arg_n>]

  The variables correspond to the **thresh** line and the affected host machine; the *actual_val* represents the actual value of *field_name* that forced it over the threshold value. The values arg_1 through arg_n represent additional optional arguments that can be used as needed; if any of these arguments contain spaces, enclose the argument in double quotes.

  A **thresh** line can be used to set either a global threshold value or a process-specific threshold value. To define a global threshold, place the **thresh** line before any **fs** or **cm** lines. A global **thresh** line applies to all monitored File Server or Cache Manager processes. To define a process-specific threshold value, place the **thresh fs** or **thresh cm** line after the appropriate **fs** or **cm** line, respectively and prior to any other **fs** or **cm** lines. A process-specific **thresh** line applies only to the preceding File Server or Cache Manager process. A process-specific threshold value always overrides a corresponding global value.

  Note: It is illegal to specify a **thresh fs** line after a **cm** line or a **thresh cm** line after a **fs** line.

- **show** <**fs**|**cm**> <*field/group/section*>

  The **show** line is used to customize the File Servers and/or Cache Managers screen so that the customized screen provides only the statistics of interest listed in a specified order. A **show** line instructs **afsmonitor** to display only the statistics fields identified by that and other **show** lines for the corresponding Cache Managers (**cm**) or File Servers (**fs**) screen. A **show** line can appear anywhere in the configuration file and you can include as many as necessary to customize the screens to your liking. The order of the statistics field on the corresponding screen (from left to right) will follow the order established in the configuration file (from top to bottom).

  If there are no **show** lines, all statistics fields for both Cache Managers and File Servers are displayed. If only **show fs** lines are included, the specified File Server statistics fields are displayed and all Cache Manager statistics fields are displayed. If only **show cm** lines are included, the specified Cache Manager statistics fields are displayed and all File Server statistics are displayed.

Each statistic monitored by **afsmonitor** is part of a statistics group and each group is a part of a statistics section. The *field/group/section* variable is used to specify a single field, group, or section to be displayed. If a field is specified, only that statistic field is displayed; if a group or section is specified, all of the statistics fields that are members of that group or section are displayed. See the section on **afsmonitor** in the *AFS System Administrator's Guide* for a complete listing of the available statistics and their groups and sections.

- **#** <*comments*>

  Begin comment lines with a pound symbol (#) in the very first column of the line.

REQUIREMENTS

The **afsmonitor** command can only be run on systems that

- Support the AFS xstat libraries which **afsmonitor** relies upon to gather its data.
- Contain the "curses" graphics package which is provided as a standard utility on most UNIX distributions.

The **afsmonitor** screens can be displayed on both "dumb" terminals and windowing systems that emulate terminals. The **afsmonitor** screens make use of reverse video and cursor addressing so for the screen to look its best, the display environment should support those features. You should set the environment variable `TERM` to the correct terminal type or one with characteristics similar to the type of terminal you are using. The display window or terminal should be at least 80 columns wide and 12 lines long.

Any number of instances of **afsmonitor** may run on a single machine; however, each instance runs in a separate dedicated window or on a dedicated terminal. Also, it may take up to three minutes to start an additional instance. Once an **afsmonitor** process is initiated, you will be unable to perform other activities in the dedicated window or on the dedicated terminal until the session is exited. The **afsmonitor** program must run in the foreground.

ARGUMENTS

initcmd is an optional string that accommodates the command's use of the AFS command parser. It can be omitted and ignored.

-config specifies the name of a configuration file to be used by **afsmonitor**; the configuration file contains a list of the processes to monitor, the threshold values to alert on, and the statistics to display. If a complete pathname is not provided the file name is interpreted relative to the current directory. Although this argument is optional, you must use either this argument or the **-fshosts** and/or **-cmhosts** arguments. See the section on **afsmonitor** in the *AFS System Administrator's Guide* for information on creating this file.

-frequency specifies the frequency in seconds with which File Server and Cache Manager processes are probed. You may use a value between 1 and 86400; the default value is 60. This frequency applies to both File Server and Cache Manager probes; however, File Server and Cache Manager probes are initiated and processed independent of each other. The actual interval between probes to a host is the probe frequency plus the time needed by all hosts to respond to the probe.

-output specifies the name of an output file to which **afsmonitor** will write all of the statistical data that it collects. By default, no output file is created. See the section on **afsmonitor** in the *AFS System Administrator's Guide* for information on this file.

-detailed indicates that the output file (specified with the **-output** argument) should be laid out in an easily readable format. This flag should only be used with the **-output** argument.

-fshosts identifies each File Server process to monitor by specifying the host it is running on. You can identify a host using either its complete Internet-style host name or an abbreviation acceptable to the cell's naming service. Do not use this argument with the **-config** argument; it may be used with the **-cmhosts** argument.

-cmhosts identifies each Cache Manager process to monitor by specifying the host it is running on. You can identify a host using either its complete Internet-style host name or an abbreviation acceptable to the cell's naming service. Do not use this argument with the **-config** argument; it may be used with the **-fshosts** argument.

-buffers specifies the number of circular buffer slots that the **afsmonitor** uses to store the data it collects. This facility is provided for future enhancements to the AFSMonitor and provides no functionality at this time.

-help prints the online help for this command. Do not provide any other arguments or flags with this one.

OUTPUT

The **afsmonitor** program displays its data on three screens:

- **System Overview screen**: The System Overview screen is the first screen displayed when **afsmonitor** is executed. It provides summary information on the monitored processes. The screen provides information on the number of processes monitored, the host associated with each process, and the alert status of each process (that is, the number of statistics above a threshold value and whether or not a process failed to respond to the last probe). The information on this screen is categorized into Cache Manager and File Server groupings.
- **File Servers screen**: This screen displays detailed statistical information on each File Server process being monitored. Processes that failed to respond to the last probe are identified; statistics that exceed threshold values are highlighted.
- **Cache Managers screen**: This screen displays detailed statistical information on each Cache Manager process being monitored. Processes that failed to respond to the last probe are identified; statistics that exceed threshold values are highlighted.

Standard Screen Features

Each screen displays four standard blocks of information, one in each corner:

- In the top left corner, the program name and version number are displayed.
- In the top right corner, the screen title, current and total page numbers, and current and total column numbers are displayed. The page numbers (e.g., p. 1 of 3) indicate the current page of information that is displayed followed by the total number of pages over which all available data is vertically displayed. The column numbers (e.g., c. 1 of 235) indicate the current left most column that is displayed followed by the total number of columns over which all available data is horizontally displayed. (The symbol >>> indicates that there is additional data to the right; the symbol <<< indicates that there is additional data to the left.)
- In the bottom left corner, a list of available commands is displayed. Typing the first letter in the command name will execute that command. Only those commands that can be executed will be displayed; for example, if there is only one page of data, there is no reason to scroll up or down, so the **n**ext and **p**rev commands will not appear. See the section on **afsmonitor** in the *AFS System Administrator's Guide* for information on the commands available on the AFSMonitor screens.
- In the bottom right corner, the probe count and probe frequency are displayed. The probe counts indicate the number of times that the File Servers (fs) and/or Cache Managers (cm) have been probed for statistics. These numbers can be different depending on the number of Cache Managers and File Servers being monitored and the probe frequency.

System Overview Screen

When **afsmonitor** is initialized, the System Overview screen is displayed. This screen shows the status of as many File Server and Cache Manager processes as can be initially displayed; additional data can be accessed by scrolling up and down.

The information on this screen is categorized into Cache Manager and File Server groupings. The left half of the screen displays File Server information and the right half displays Cache Manager information. At the top of each grouping there are two pieces of information:

- The number of processes (whether Cache Manager or File Server) monitored by this **afsmonitor** process.
- The number of alerts and the number of processes (machines) affected by those alerts. (Each alert represents a single instance of a statistic exceeding its threshold or of a process failing to respond to the last probe.)

Under this status information is a list of the hosts containing the monitored processes. If one or more of a process's statistics have exceeded a threshold, the number of above-threshold statistics are listed in square brackets to the left of that process. If a process failed to respond to the last probe, the letters "PF" (probe failure) appear in square brackets to the left of that process.

File Servers Screen

Executing the **fs** command option from either the System Overview or Cache Managers screen displays the File Server screen. This screen provides detailed statistical information collected from the File Servers during the last probe.

The File Servers screen is laid out as follows:

- A summary line is displayed at the top of the screen just below the standard program version and screen title blocks. It specifies the number of monitored processes, the number of alerts, and the number of processes (machines) affected by the alerts.
- The names of the hosts containing the monitored File Server processes are displayed in separate rows down the left side of the screen. These host names are always displayed, even when the screen is scrolled left and right.
- As many columns of data as can fit within the confines of the window are displayed to the right of the host names. Each column is labeled with the name of its corresponding statistic. The order and content of the statistical display area is configurable. As the screen is scrolled left and right, these columns change to display new data. If the last probe failed to contact a File Server, the fields in that row will contain dashes (--); otherwise, some value will be displayed. If a value exceeds its configured threshold, it will be highlighted in reverse video. The columns are 10 characters wide; if a particular value is too large to fit into one column, it overflows into the next row in the same column.

Cache Managers Screen

Executing the **cm** command from either the System Overview or File Servers screen displays of the Cache Managers screen. This screen provides detailed information collected from the Cache Managers for the latest probe.

The File Servers screen is laid out as follows:

- A summary line is displayed at the top of the screen just below the standard program version and screen title blocks. It specifies the number of monitored processes, the number of alerts, and the number of processes (machines) affected by the alerts.
- The names of the hosts containing the monitored Cache Manager processes are displayed in separate rows down the left side of the screen. These host names are always displayed, even when the screen is scrolled left and right.
- As many columns of data as can fit within the confines of the window are displayed to the right of the host names. Each column is labeled with the name of its corresponding statistic. The order and content of the statistical display area is configurable. As the screen is scrolled left and right, these columns change to display new data. If the last probe failed to contact a Cache Manager, the fields in that row will contain dashes (--); otherwise, some value will be displayed. If a value exceeds its configured threshold, it will be highlighted in reverse video. The columns are 10 characters wide; if a particular value is too large to fit into one column, it overflows into the next row in the same column.

Output File

All of the statistical information collected and displayed by AFSMonitor may be preserved by writing it to an output file. You can create an output file by using the **-output** argument when the you startup the **afsmonitor** process. The ability to record this information allows you to analyze process performance over long periods of time and to apply post-processing techniques to further analyze system trends.

The AFSMonitor output file is a simple ASCII file that records the information reported by the File Server and Cache Manager screens. The output file has the following format:

<*time*> <*host_name*> <**CM**|**FS**> <*list_of_measured_values*>

The line specifies the *time* at which the *list_of_measured_values* were gathered from the Cache Manager (**CM**) or File Server (**FS**) process housed on *host_name*. On those occasion where probes fail, a single *-1* is reported instead of the *list_of_measured_values*.

If you intend to view this file often, you may want to use the **-detail** flag to format the data into a more user-readable form. Of course, the more readable version of the file is much more difficult to use as input to programs or scripts.

EXAMPLES

See the section on **afsmonitor** in the *AFS System Administrator's Guide* for examples of AFSMonitor screens and an AFSMonitor configuration file.

PRIVILEGE REQUIRED

None.

bosserver — initialize BOS Server.

/usr/afs/bin/bosserver [-noauth] [-log]

ACCEPTABLE ABBREVIATIONS

This command does not use the syntax conventions of the AFS command suites, so the command name and its flags must be typed in full.

DESCRIPTION

Initializes the Basic OverSeer (BOS) Server; the bosserver must be run on every file server machine. The ***Basic OverSeer Server (BOS Server)*** reduces the demands on system administrators by performing the following tasks:

- Monitoring the other server processes (on the local machine) to make sure they are running correctly.
- Automatically restarting failed processes, without contacting a human operator. When restarting multiple server processes simultaneously, the BOS server takes interdependencies into account and initiates restarts in the correct order.
- Accepting requests from the system administrator via the **bos** command suite. Common reasons to contact the BOS server are to verify the statuses of server processes on file server machines, install and start new processes, stop processes either temporarily or permanently, and restart dead processes manually.
- Helping system administrators to manage system configuration information. The BOS server automates the process of adding and changing ***server encryption keys***, important in mutual authentication. The BOS Server also provides a simple interface for modifying two files that contain information about privileged users and certain special file server machines. See chapter 2 for more details about these configuration files.
- Providing information when illegal or nonexistent names are passed for any of the *fs*-related processes (**fileserver**, **volserver**, or **salvager**). An error occurs if incorrect names are specified.

The BOS Server creates a log file in */usr/afs/logs/BosLog*. Including the **-log** flag causes the BOS Server to record in *BosLog* the names of all users who perform **bos** commands that require the user be included in the local */usr/afs/etc/UserList* file. The *BosLog* file can be read using the **bos getlog** command

When the BOS Server is first initialized on a file server machine, it creates several files and directories under */usr/afs* on the local disk (such as */usr/afs/db* and */usr/afs/etc/KeyFile*). It sets the mode bits on files and directories according to the table that appears in the DESCRIPTION section of the **bos status** command in Chapter 2.

Each subsequent time the BOS Server restarts, it checks to make sure that the mode bits on the files and directories in */usr/afs* are still set correctly. If the mode bits do not match those listed in the table, the BOS Server adds a warning to *BosLog*. The warning also appears at the start of the output from the

bos status command if the **-long** flag is used. Because the BOS Server does not reset the mode bits, you can set them to alternate values as appropriate.

ARGUMENTS

-noauth	assigns the unprivileged identity *anonymous* to the issuer.
-log	causes the BOS Server to record in the */usr/afs/logs/BosLog* file the names of all users who successfully initiate a **bos** command that require the issuer be listed in the local */usr/afs/etc/UserList* file.

EXAMPLES

The following initializes the BOS Server and directs it to record in the */usr/afs/logs/BosLog* file the names of users who initiate **bos** commands that require the issuer to be in the */usr/afs/etc/UserList* file:

```
% bosserver -log &
```

PRIVILEGE REQUIRED

Issuer must be logged into the machine's UNIX file system as "root."

MORE INFORMATION

bos create
bos exec
bos getlog
bos getrestart
bos restart
bos shutdown
bos start
bos startup
bos status
bos stop

budb_convert — convert Backup Database from AFS 3.1 to AFS 3.2.

budb_convert

ACCEPTABLE ABBREVIATIONS/ALIASES

This command does not use the syntax conventions of the AFS command suites. It must be typed in full.

DESCRIPTION

Converts information from an AFS 3.1 Backup System for use with the AFS 3.2 Backup System. The AFS 3.1 Backup System consisted in part of a "Backup Database" in the form of a collection of files in the */usr/afs/backup* directory on the backup machine. The files that comprised the AFS 3.1 Backup Database contained information about volume sets and entries, dump schedules (the dump hierarchy), Tape Coordinators, and existing dump sets and the volumes they contain.

The AFS 3.2 Backup System includes a Backup Database replicated on all database server machines in a cell. There is a single version of a Backup Database per cell, with copies residing on each database server machine. The information formerly housed in the files in */usr/afs/backup* on a backup machine is now contained in the replicated Backup Database on the database server machines.

The **budb_convert** command converts information from the files on the former backup machine to the Backup Database on the database server machines. The command must be executed from the 3.1 backup machine. A Backup Server process must be running on each database server machine before the command is issued. (Each Backup Server automatically creates an empty Backup Database on its machine if one does not exist when it is started.) The binary file for the command resides in the */usr/afsws/etc* directory.

To convert 3.1 backup files to 3.2 Backup Database format, execute the **budb_convert** command. The command executes in four phases:

1. It reads the */usr/afs/backup/dumpschedule* file and writes a file named *dumpsched.sh* to the same directory. The *dumpsched.sh* file contains the **backup adddump** commands necessary to recreate the dump hierarchy.
2. It reads the */usr/afs/backup/volumeset* file and writes a file named *volset.sh* to the same directory. The *volset.sh* file contains the **backup addvolset** and **backup addvolentry** commands necessary to recreate the volume sets and volume entries.
3. It reads the */usr/afs/backup/tapehosts* file and writes a file named *hosts.sh* to the same directory. The *hosts.sh* file contains the **backup addhost** commands necessary to recreate Tape Coordinator entries.
4. It reads each dump file (*/usr/afs/backup/D*.db*) and creates a record of each dump ID in the Backup Database. The command does not create a *.sh* file during this phase (as it does in the previous three); rather, it reads the information from these files and adds it directly into the Backup Database. However, it does create a *convert.log* file in the directory, in which it lists the names of all dump sets for which information was either successfully or unsuccessfully added to the Backup Database. The command does not add duplicate dump

IDs to the database, nor does it add dump IDs for which inconsistencies were present in the 3.1 backup files. Inconsistencies can result from repeated or interrupted dump sets or from dump sets that contained no volumes.

When the four phases are done, examine the *convert.log* file to determine which dump sets were successfully added to the Backup Database and which were not. Verify that all error messages and their causes are clearly accounted for. (If possible, dispose of any inconsistencies before running the command.)

The three files (*dumpsched.sh*, *volset.sh*, and *hosts.sh*) generated by the **budb_convert** command are shell scripts that contain **backup** commands (as described previously). They can be executed from the command line to issue the commands they contain, entering the corresponding information into the database.

If the **budb_convert** command is being used to merge 3.1 backup information with an already existing 3.2 Backup Database, verify the contents of the three script files before executing them. Make sure the files do not contain commands to add dump levels, volume sets, volume entries, or Tape Coordinators that the database already contains (the database may already contain such information if **backup** commands were used to define it prior to using **budb_convert** to convert the 3.1 files). Remove or edit any commands in the files that will duplicate information already in the database. (Note that this is not an issue if the **budb_convert** command is being used to create a new Backup Database.)

Finally, execute the three script files to add the information extracted from the 3.1 backup files to the Backup Database. Execute the scripts from the former 3.1 backup machine on which they were created; they contact the Backup Server to add the information to the Backup Database on the database server machines. Commands from the **backup** suite (such as **listvolsets** and **listdumps**) can be used to verify that the information in the database is complete and current. All **backup** commands can then be used to execute operations using the new Backup Database.

Note that only information from the 3.1 backup files needs to be upgraded for use with the 3.2 Backup System. Tapes containing data dumped using the 3.1 Backup System are fully compatible with the 3.2 Backup System. After **budb_convert** and the three script files are run, the tapes can be used to restore information dumped with the 3.1 Backup System.

In addition, it is not necessary to copy the files in the */usr/afs/backup* directory to another location before running **budb_convert**. The command does not affect the backup files.

WARNING

The **budb_convert** command can take several hours to complete. The command generates the three shell scripts relatively quickly, but it takes some time to transfer the information about dump sets from the former backup files to the new Backup Database. (The command passes information about each dump set to the Ubik synchronization site for the database, and Ubik distributes a new copy of the database after each update.) Execute the command over a weekend or overnight.

The 3.1 Backup System did not record volume sizes; the 3.2 Backup System does. Because volume size information was not available from the 3.1 backup files, all volumes converted from the backup files are recorded in the 3.2 Backup Database with size `0`.

OUTPUT

During its first three phases, the **budb_convert** command writes the *dumpsched.sh*, *volset.sh*, and *hosts.sh* script files to the */usr/afs/backup* directory on the backup machine. When it has completed the fourth phase, it displays the following message on the screen:

```
Database conversion complete To add dump schedules, volume sets and
tape hosts.....  Run the shell scripts in /usr/afs/backup
```

The three script files can then be run to execute the **backup** commands they contain.

EXAMPLES

The following executes the **budb_convert** command on the 3.1 backup machine:

```
% budb_convert
```

PRIVILEGE REQUIRED

The issuer must be able to read, insert, and write to files in the */usr/afs/backup* directory on the 3.1 backup machine. The issuer must also be listed in */usr/afs/etc/UserList*.

buserver — initialize Backup Server.

/usr/afs/bin/buserver [-database <*database directory*>**]**
[-cellservdb <*cell configuration directory*>**] [-resetdb]**
[-noauth] [-smallht]
[-servers <*list of ubik database servers*>+**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

/usr/afs/bin/buserver [-d <*database directory*>**] [-c** <*cell configuration directory*>
[-r] [-b] [-sm] [-se <*list of ubik database servers*>+**] [-h]**

DESCRIPTION

Initializes the Backup Server which runs on all designated database server machines. The Backup Server is a Ubik server that manages multiple, replicated copies of the Backup Database.

Note: This command is not normally issued at the command shell prompt, but rather placed into a database server machine's */usr/afs/local/BosConfig* with the **bos create** command. If it is ever issued at the command shell prompt, the issuer must be working on a database server machine.

When executed, the Backup Server process creates the Backup Database in */usr/afs/db* if the database does not already exist. The database exists in the form of two files: a *bdb.DB0* database file and a *bdb.DBSYS1* database log file.

The Backup Database houses information about volume sets and entries, the dump hierarchy, Tape Coordinators, and previously performed dump sets. Requests to store and retrieve information in the Backup Database are sent to the Backup Server via the **backup** program. Different **backup** commands can be used to add, remove, or view information stored in the Backup Database. The **backup** program can be run on any machine in the cell. It locates the *buserver* processes for the cell via the */usr/afs/etc/CellServDB* file on the machine on which it is running.

When executed, the Backup Server also creates an activity log file named */usr/afs/logs/BackupLog* if the file does not already exist. This log file can be examined using the **bos getlog** command.

Note: When the Backup Server is started on multiple database server machines, it requires approximately three minutes for the server processes on the different machines to coordinate and prepare to receive requests.

WARNING

The *buserver* process reserves port 7021 for its use. Unexpected behavior can occur if another process tries to reserve this port while the *buserver* process is running.

ARGUMENTS

-database specifies the pathname of an alternate directory for the Backup Database. Include a slash after the final element of the directory's pathname. The Backup Server checks the specified directory for the Backup Database files. By default, the Backup Server looks in the */usr/afs/db* directory for the Backup Database files.

-cellservdb specifies the pathname of an alternate directory for the *CellServDB* file. Use this argument if you plan to start the Backup Server on a subset of the database server machines that is not a majority of the machines listed in the */usr/afs/etc/CellServDB* file. If an alternate CellServDB file is not specified in this case, the Backup Server starts but is unable to contact the Backup Database. By default, the Backup Server uses the */usr/afs/etc/CellServDB* file.

-resetdb truncates the existing Backup Database files in the */usr/afs/db* directory. The files are not removed from the directory, but all of the information they contain is deleted. All information previously contained in the Backup Database must be recreated before the Backup System can be used to backup or restore data. (Using this argument is the equivalent of removing the Backup Database files and restarting the Backup Server, in which case the server process creates new database files.)

-noauth tells the Backup Server to assign the unprivileged identity *anonymous* to the issuer. The flag instructs the Backup Server not to authenticate the issuer. Thus, it establishes an unauthenticated connection between the issuer and the Backup Server. It is useful only when authorization checking is disabled on the database server machine. In normal circumstances, the Backup Server allows only authorized (privileged) users to issue commands that affect or contact the Backup Database and will refuse to perform such an action even if the **-noauth** flag is used.

-smallht tells the Backup Server to use smaller internal hash tables for the Backup Database. The database uses hash tables internally to accelerate data access. Smaller hash tables reduce memory requirements but cause the database to take slightly longer to access data.

-servers specifies the database server machines on which the Backup Server is to be started. Use this argument if you plan to start the Backup Server on a subset of the database server machines that is a majority of the machines listed in the */usr/afs/etc/CellServDB* file. By default, the system assumes the Backup Server is to be started on all machines listed in the *CellServDB* file.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following **bos create** command creates a *buserver* process on *fs3.transarc.com*. Type the command on one line:

% bos create fs3.transarc.com buserver simple /usr/afs/bin/buserver

PRIVILEGE REQUIRED

If the */usr/afs/db/bdb.DB0*, */usr/afs/db/bdb.DBSYS1*, and */usr/afs/logs/BackupLog* files do not already exist on the database server machine, the issuer must be able to insert and write to files in the */usr/afs/db*

and */usr/afs/logs* directories on the machine. If the files do already exist, the issuer need only be able to write to them.

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

Issuer must be logged into the machine's UNIX file system as "root" to run this command from the command line.

MORE INFORMATION

bos create

bos getlog

butc — initialize Tape Coordinator process.

butc [-port <*port offset*>**] [-debuglevel** <*debug level*>**] [-cell** <*cell name*>**] [-aixscsi] [-noautoquery] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

butc [-p <*port offset*>**] [-d** <*debug level*>**] [-c** <*cell name*>**] [-a] [-n] [-h]**

DESCRIPTION

Initializes a Tape Coordinator process on a Tape Coordinator machine. The machine to which the tape drive associated with the Tape Coordinator is attached must contain a */usr/afs/backup/tapeconfig* file. The file must contain a single line for each tape drive, specifying the following information about the drive: the capacity of tapes used in the drive, the size of the end-of-file marker for the drive, the device name of the drive, and the port offset number of the drive's Tape Coordinator.

Issue the **butc** command over a connection to the Tape Coordinator machine on which the Tape Coordinator is to be started (when using a window manager, this corresponds to a separate window). A separate connection must be opened for each Tape Coordinator. Because the Tape Coordinator must run in the foreground, the connection on which the **butc** command is issued is unavailable for subsequent commands. The Tape Coordinator uses it as a dedicated "monitoring" window on which to display trace information or prompts for insertion of additional tapes into its associated drive. The Tape Coordinator window must remain open as long as the Tape Coordinator is running. To stop a Tape Coordinator process, enter an appropriate interrupt signal such as **^c** (Control-c) in the process's monitoring window.

The **butc** program also writes output to two files on the local disk of the Tape Coordinator machine:

- It writes information about its execution of operations to a log file named */usr/afs/backup/TL_*<*device_name*>. The level of detail to which each operation is described depends on the operation.
- It writes information on any problems it encounters during its execution of operations to an error file named */usr/afs/backup/TE_*<*device_name*>.

In the log and error file names, <*device_name*> is the device name of the tape drive with which the Tape Coordinator is associated. For example, if the device name of the drive is */dev/rmt0*, the log and error files are named *TL_rmt0* and *TE_rmt0*; if the device name of the drive is */dev/rmt/4m*, the files are named *TL_rmt_4m* and *TE_rmt_4m*.

Comparable information is written to both the log and error files. For example, if **backup dump** is used to back up a number of volumes, the log file lists both the names of volumes that were successfully dumped and the names of volumes that, for some reason, were omitted from the dump. In the same instance, the error file lists the names of only those volumes omitted from the dump.

The *TL_<device_name>* and *TE_<device_name>* files and the log and error files associated with a tape drive should be browsed periodically to ensure that operations such as dumps and restores are completing without problems.

By default, **butc** will initially prompt the operator to insert an appropriate tape and press **Return** when any of the following **backup** commands are executed: **backup diskrestore**, **backup dump**, **backup labeltape**, **backup readlabel**, **backup restoredb**, **backup savedb**, **backup scantape**, and **backup volrestore**. You can eliminate this initial query, to speed backup and increase the level of backup automation, by running the **butc** process with the **-noautoquery** flag. Of course for this feature to work properly, you must place the appropriate tape in the drive prior to the execution of any of the previously mentioned **backup** commands.

You can abort any **backup** operation involving a tape, prior to inserting a tape and pressing **Return**, by typing an **a** and pressing **Return** at the tape insertion prompt.

WARNING

If issuing this command on an AIX 2.2.1 machine using a driver that provides extended SCSI support, the issuer must use the **-aixscsi** flag to prevent the Tape Coordinator from doing a reset before rewinding tapes. Do not include this flag if the Tape Coordinator machine is running another operating system or is running AIX 2.2.1 without a driver that provides extended SCSI support.

ARGUMENTS

-port specifies the port offset number associated with the Tape Coordinator to be initialized. The issuer of **backup** commands uses this number when indicating that the Tape Coordinator is to execute a command. See Section 1.2 for more details on specifying port offsets.

-debuglevel specifies the types of messages the Tape Coordinator is to produce on its monitoring connection/window. Two values are legal:

- *0* indicates that the Tape Coordinator is only to prompt for new tapes, not report on its activities. This is the default value.
- *1* indicates that the Tape Coordinator is to report on its activities (for example, the volumes it is accessing and dumping), in addition to prompting the operator to place new tapes in the drive as necessary.

-cell specifies the cell with respect to which the Tape Coordinator is to run. You must be included in the */usr/afs/etc/UserList* file in the specified cell, and you must authenticate to the specified cell before using this argument. By default, the local cell of the Tape Coordinator machine is used.

The specified Tape Coordinator communicates with the Backup Server in the indicated cell. It can manipulate data in only the specified cell. An entry must be defined for the Tape Coordinator in the Backup Database of the specified cell; the entry must define a unique port offset number for the Tape Coordinator in that cell.

-aixscsi prevents the Tape Coordinator from doing a reset before rewinding tapes. Use this flag only on Tape Coordinator machines running AIX 2.2.1 with a driver that provides extended SCSI support.

-noautoquery disables the initial prompt for tape insertion given by **butc** when executing **backup** commands involving a tape drive. See the DESCRIPTION section for a complete list of the affected commands.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

EXAMPLES

The following starts the Tape Coordinator with port offset *7*. The debug level is specified as *1*, meaning the Tape Coordinator is to provide additional information as it executes commands.

```
% butc 7 1
```

PRIVILEGE REQUIRED

The issuer must be listed in */usr/afs/etc/UserList*. If the */usr/afs/backup/TL_<device_name>* and */usr/afs/backup/TE_<device_name>* files for the tape drive do not already exist on the Tape Coordinator machine, the issuer also must be able to insert and write to files in the */usr/afs/backup* directory on the machine; if the log and error files already exist, the issuer need only be able to write to them.

dkload — incorporate external libraries into kernel without rebooting.

dkload [-readOnly] [-quiet] [-verbose] [-syscallresult <*number*>**]**
[-path <*object path*>**] [-ld_cmd** <*loader path*>**]**
[-as_cmd <*assembler path*>**] [-nm_cmd** <*nm path*>**]**
[-libcommon <*dkload libcommon.a path*>**]**
[-kernel_alloc <*dkload kernel allocator path*>**]**
{-name <*mod*> | <*object files*>$^{+}$**} [-dir** <*scratch dir for dkload*>**]**
[-deletedir]

ACCEPTABLE ABBREVIATIONS

This command does not use the syntax conventions of the AFS command suites. Therefore, the command must be typed in full and its switches must always be included, though they may be shortened as indicated:

dkload [-r] [-q] [-v] [-s <*number*>**] [-p** <*object path*>**] [-ld** <*loader path*>**]**
[-a <*assembler path*>**] [-nm** <*nm path*>**] [-li** <*dkload libcommon.a path*>**]**
[-k <*dkload kernel allocator path*>**] {-na** <*mod*> | <*object files*>$^{+}$**}**
[-di <*scratch dir for dkload*>**] [-de]**

DESCRIPTION

Loads one or more libraries into the memory version of the local machine's kernel. It does not alter the disk version of the kernel (*/vmunix* or equivalent). Its intended use is loading the AFS routine library, *lib.afs*, into the kernel on client machines.

The dynamic loader begins by requesting (from a low-level kernel routine) allocation of a certain amount of memory in which to load the libraries. It resolves cross-references between procedures in the existing kernel and in the libraries (referred to as "linking" the two). The result is a list of memory addresses for the cross-referenced procedures, which the dynamic loader stores in a table. It generates an executable version of the libraries with correct addresses inserted for all of the necessary kernel variables and procedures, and loads the executable into the memory space allocated in the first phase.

By default, the command writes temporary files to */tmp* as it executes. Use the **-dir** flag to redirect these files to a different directory; the command will automatically change the mod bits of this directory to `700`.

REQUIREMENTS

The *dkload* binary file should be available in the current working directory, or the issuer must specify a pathname in the command name. The standard directory is */usr/vice/etc/dkload* on client machines and */usr/afs/bin/dkload* on file server machines.

The file *kalloc.o* should be available in the current working directory, or the issuer must specify a pathname with either the **-kernel_alloc** or **-path** argument. This file helps in the first phase of dynamic

loading: allocating kernel memory for the libraries. It is generated automatically during the compilation of the **dkload** program.

The file *libcommon.a* should be available in the current working directory, or the issuer must specify a pathname with either the **-libcommon** or **-path** argument. This file helps in the second phase of dynamic loading: resolving cross-references. It is generated automatically during the compilation of the **dkload** program.

The library file(s) to be loaded (such as *lib.afs*) should be available in the current working directory, or the issuer must use the **-path** argument to specify the correct path.

The binary files for the standard UNIX commands **ld**, **as**, and **nm** should be available in a local disk directory included in the issuer's $PATH environment variable. Otherwise, the issuer needs to use the **-ld_cmd**, **-as_cmd** and/or **-nm_cmd** arguments to specify the correct pathname.

ARGUMENTS

-readOnly directs the command interpreter to report the actions it would perform if executing the command, rather than actually performing it.

-quiet suppresses the trace of actions that by default appears on standard output (stdout).

-verbose increases the amount of information in the trace that appears on standard output (stdout). Multiple instances of this flag may be provided, resulting (up to a certain point) in a increasing level of detail in the trace.

-syscallresult specifies the memory address at which allocation begins when the **-readOnly** flag is provided. This is useful when the issuer wants to specify a memory address obtained during a previous aborted run of **dkload**.

Provide this argument only when using **-readOnly**. The value may be either a large decimal or hexidecimal number (the latter beginning with *0x*). If this flag is not provided, the value defaults to *0xc1456780*, which may not be acceptable on all machine types but has the advantage that it causes allocation to begin well above the addresses used by standard libraries.

-path specifies the directory in which the command interpreter can find *kalloc.o*, *libcommon.a*, and each library to be loaded, if they are not in the current working directory. The value of this argument is overridden by **-kernel_alloc** and **-libcommon**.

-ld_cmd specifies the pathname to the binary for the UNIX **ld** command, which the kernel dynamic loader uses during the linking phase.

This argument is necessary only if the issuer's $PATH environment variable will not lead to the correct binary file. The binary file cannot be in AFS, for instance, since the dynamic loader runs before the machine can access AFS.

-as_cmd specifies the pathname to the binary for the UNIX **as** command, which the kernel dynamic loader uses during the linking phase.

This argument is necessary only in the conditions specified under **-ld_cmd**.

-nm_cmd specifies the pathname to the binary for the UNIX **nm** command, which the kernel dynamic loader uses during the linking phase.

This argument is necessary only in the conditions specified under **-ld_cmd**.

-libcommon specifies the pathname of the *libcommon.a* file.

This argument is necessary only if the file does not reside in the current working directory or if the **-path** argument does not indicate the correct directory for it.

-kernel_alloc specifies the pathname of the *kalloc.o* file.

This argument is necessary only in the conditions specified under **-libcommon**.

-name specifies the variable part of the library name to be loaded: the command interpreter loads the library called "*lib*<***object files***>*.a*". For example, if <***object files***> is *afs*, then *libafs.a* gets loaded. The <*mod*> instance is not used.

-dir specifies the full pathname that **dkload** will use to store its temporary files. If this flag is not used, **dkload** will store its temporary files in */tmp*.

-deletedir indicates that the directory and temporary files created as a result of using the **-dir** argument should be deleted upon completion of the program. This flag can only be used in conjunction with the **-dir** argument.

EXAMPLES

The following loads in the AFS libraries. It assumes that all files can be found in the current directory or other expected place. The issuer desires extra trace information.

```
% dkload -verbose -name afs
```

PRIVILEGE REQUIRED

Issuer must be logged into the machine's UNIX file system as "root" or at least have `w` access to */dev/mem* and */dev/kmem*.

fileserver — initialize File Server component of *fs* process.

/usr/afs/bin/fileserver [-d <*debug level*>**] [-p** <*number of processes*>**]**
[-spare <*number of spare blocks*>**]**
[-pctspare <*percentage spare*>**] [-b** <*buffers*>**]**
[-l <*large vnodes*>**] [-s** <*small vnodes*>**]**
[-vc <*volume cachesize*>**] [-w** <*call back wait interval*>**]**
[-cb <*number of call backs*>**]**
[-banner <*print banner every 10 minutes*>**]**
[-hr <*number of hours between refreshing the host cps*>**]**
[-m <*min percentage spare in partition*>**]**
[-L <*large server conf*>**] [-S** <*Small server conf*>**]**
[-k <*stack size*>**] [-c** <*check interval*>**] [-help]**

ACCEPTABLE ABBREVIATIONS

This command does not use the syntax conventions of the AFS command suites, so the command and its switches and flags must be typed in full.

DESCRIPTION

Initializes the File Server component of the *fs* process.

Note: This command is not normally issued at the command shell prompt, but rather placed into a database server machine's */usr/afs/local/BosConfig* with the **bos create** command. If it is ever issued at the command shell prompt, the issuer must be working on a file server machine.

The arguments on **fileserver** allow the issuer to configure many aspects of the commands behavior, as detailed in the ARGUMENTS section. However, there are three standard configurations (small, medium, and large) that set the following parameters to the indicated values:

- The maximum number of lightweight processes (LWPs) the File Server uses to handle requests for data (set with **-p**). The File Server always uses a minimum of 32K for these processes. (**Small:** 6, **Medium:** 9, **Large:** 12)
- The maximum number of directory blocks the File Server caches (set with **-b**). Each cached directory block (buffer) uses 2K plus 44 bytes of memory. (**Small:** 70, **Medium:** 90, **Large:** 120)
- The maximum number of large vnodes the File Server caches for tracking directory elements (set with **-l**). Each large vnode cached requires 292 bytes of memory. (**Small:** 200, **Medium:** 400, **Large:** 600)
- The maximum number of small vnodes the File Server caches for tracking file elements (set with **-s**). Each small vnode cached requires 100 bytes of memory. (**Small:** 200, **Medium:** 400, **Large:** 600)
- The maximum number of call backs the File Server can cache (set with **-cb**). Each call back uses 16 bytes of memory. (**Small:** 20,000, **Medium:** 32,000, **Large:** 40,000)

- The maximum number of **Rx** packets the File Server uses. Each packet requires 1544 bytes of memory. (**Small:** 100, **Medium:** 150, **Large:** 200)
- The maximum volume cache size which defines the number of volumes which the fileserver can cache before having to retrieve data from disk (set with **-vc**). (**Small:** 200, **Medium:** 400, **Large:** 600)

By default, the **fileserver** command initializes with the medium configuration; the **-S** flag is used to specify the small configuration and the **-L** flag is used to specify the large configuration. The individual settings in each configuration size is the default; use the indicated flag to override these defaults.

The amount of memory required for the File Server varies, depending on the configuration size employed and whether or not any of the standard settings are overridden. The approximate default memory usage is 751K for small, 1.1M for medium, and 1.4M for large. If a machine has additional memory, consider boosting the number of callbacks and the volume cache size first (using the **-cb** and **-vc** arguments, respectively); these two parameters, more than any of the others, directly affect File Server performance.

While the medium (default) and large configurations should result in improved File Server performance, employing too large a configuration may not always result in enhanced performance. How you configure any File Server depends on factors such as how much data is stored on the file server machine and the amount of memory available on the machine.

In addition to making the above configuration decisions, issuer of this command must also determine how much extra storage users will have once they have exceeded their volume quotas. By default users can store up to an extra megabyte (1024 kilobytes) of data. This default can be changed by using **-spare** or **-pctspare** arguments:

- **-spare** specifies the number of kilobytes over quota the File Server will allow. To make users unable to store files after the quota is exceeded, specify a value of *0*.
- **-pctspare** expresses the amount of excess the File Server will allow as a percentage of the volume's quota.

Note: The user still cannot create new files after his or her quota is exceeded.

Because AFS has the ability to include IP addresses in protection groups, it is important that the File Server have some way of refreshing its list of current protection subgroups and the machines whose IP addresses they contain (its "host CPS"). Including a machine's IP address in a protection group does not provide the machine with access to data until the file server machine that distributes the data is aware of the inclusion. (The same limitation applies to a user who is added to a group; however, unlike a machine, a user can use the **klog** command to refresh his or her tokens and receive access to all data to which the group has access.

By default, the File Server checks protection groups to determine current machine membership every 2 hours. The **-hr** flag is used to increase or decrease the frequency with which the File Server will refresh its host CPS. (An interface also exists that allows for the creation of programs to manually refresh the host CPS.)

When executed, the File Server creates an activity log file named */usr/afs/logs/FileLog* if the file does not already exist. This log file can be examined using the **bos getlog** command.

WARNING

The **-k** and **-w** arguments are intended for use by AFS developers only. Changing them from their default values is strongly discouraged and may result in unpredictable File Server behavior.

Do not specify both **-spare** and **-pctspare**. Doing so causes the File Server to exit, leaving an error message in */usr/afs/logs/FileLog*.

ARGUMENTS

-d sets the detail level for the debugging trace the File Server produces in */usr/afs/logs/FileLog*. Provide a positive integer; higher values (up to a point) produce greater detail. The default level of *0* produces no trace.

-p sets the number of server lightweight processes (LWPs) to run. Provide a positive integer up to *16*; values above *16* are reduced to *16*. The (medium) default is 9.

-spare specifies the number of extra kilobyte blocks the File Server will allow users to store in a volume after its quota is exceeded. Provide a positive integer; a value of *0* prevents the volume from ever exceeding its quota. Do not combine this argument with **-pctspare**.

-pctspare specifies the amount by which the File Server will allow a volume to exceed its quota, as a percentage of the quota. Provide an integer between *0* and *99*. A value of *0* prevents the volume from ever exceeding its quota. Do not combine this argument with **-spare**.

-b sets the number of directory buffers, which are 2 kilobytes each. Provide a positive integer. The (medium) default is 90.

-l sets the number of large vnodes to use (for tracking directory elements). Provide a positive integer. The (medium) default is 400.

-s sets the number of small vnodes to use (for tracking file elements). Provide a positive integer. The (medium) default is 400.

-vc sets the number of volumes to cache. Provide a positive integer. The (medium) default is 400.

-w sets the interval at which the daemon spawned by the File Server performs its maintenance tasks. The default is 300 seconds (5 minutes). Do not use this argument; changing its default may cause unpredictable behavior.

-cb sets the number of callbacks the File Server can track. Provide a positive integer. The (medium) default is 32,000.

-banner causes the File Server to prints the following banner to */dev/console* about every 10 minutes.

```
File Server is running at time.
```

-hr specifies how often the File Server refreshes its knowledge of the machines that belong to protection groups (its "host CPS"). The File Server must update this

information to enable users from machines recently added to protection groups to access data for which those machines now have the necessary ACL permissions.

-m specifies the percentage by which the File Server allows partitions on the file server machine to exceed their quotas. The File Server leaves the specified percentage of a partition's total size free on the disk on which the partition resides. Specify an integer value between *0* and *30*; the default is *5*. A value of *0* prevents partitions from ever exceeding their quotas.

Note: This argument appears only on machines running an AIX operating system. This argument is useful if you plan to store very large files on a file server machine. The syntax of the **fileserver** command on other system types does not include this argument.

-L configures a "large" File Server (see the DESCRIPTION for details). If you use this flag, do not use the **-S** flag; omit both flags to configure a "medium" File Server.

-S configures a "small" File Server (see the DESCRIPTION for details). If you use this flag, do not use the **-L** flag; omit both flags to configure a "medium" File Server.

-k sets the LWP stack size in units of 1 kilobyte. Provide a positive integer. The default is 24 (kilobytes). Do not set this argument lower than the default.

-help prints the online help entry for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

The following **bos create** command creates an *fs* process on *fs2.transarc.com* that uses the large configuration size, and allows volumes to exceed their quota by 10%. Type the command on a single line:

```
% bos create fs2.transarc.com fs fs "/usr/afs/bin/fileserver -pctspare 10 -L"
                /usr/afs/bin/volserver  /usr/afs/bin/salvager
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

Issuer must be logged into the machine's UNIX file system as "root" to run this command from the command line.

MORE INFORMATION

backup diskrestore

backup dump

backup labeltape

backup readlabel

backup restoredb

backup savedb

backup scantape

backup volrestore

bos create

bos getlog

salvager

volserver

fms — determine tape capacity and end-of-file mark size.

fms -tape <*tape special file*> **[-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

fms -t <*tape special file*> **[-h]**

DESCRIPTION

Determines the total capacity and the size of the end-of-file (EOF) marker for the tape in the tape drive identified by **-tape**. EOF marker size can differ for each combination of tape and tape drive.

The AFS Backup System separates each dumped volume on a tape with an EOF marker, which is a section of blank tape. For some combinations of tape and tape drive, the marker is large enough (for example, one megabyte) that failure to consider it when writing a tape significantly overestimates the amount of space left before the end of the tape. A wrong estimate can cause the Backup System to reach the end of a tape while writing, which renders the entire tape useless for restoration purposes.

To make the Backup System's estimate of available space more accurate, each entry in the */usr/afs/backup/tapeconfig* file specifies tape capacity and EOF marker size, among other information. Use this command to determine the correct sizes.

Before issuing this command, insert a tape in the drive. The tape is overwritten while the command executes; use a blank tape, one that can be recycled, or one to be initialized with file marks. Because this command inserts file marks onto the entire tape, it can take from several hours to more than a day to complete.

WARNING

Because EOF marker size can differ for each combination of tape and tape drive, the capacity reported by this command should be decreased by 10-15% before being used in the *tapeconfig* file. Similarly, the EOF marker size should be increased by 10-15% before being used in the *tapeconfig* file. The chapter on the Backup System in the *AFS System Administrator's Guide* provides more information on the *tapeconfig* file.

This command can take from several hours to more than a day to complete. Also, the command overwrites the tape in the specified drive.

You should not use the **fms** command on data compressing tape drives because the command will give misleading numbers. Instead, you should configure the tape size to a number larger than the tape itself. The end-of-tape (EOT) handling capacity of the Tape Coordinator will prevent dumps from failing when they hit the EOT.

ARGUMENTS

-tape specifies the UNIX device name of the tape drive for which tape size and EOF mark size are to be determined; for example, */dev/sd0a*.

-help prints the online help for this command. Do not provide any other arguments or flags with this one. See section 1.2 for more details.

OUTPUT

The command sends output to standard output (stdout) and to an *fms.log* file, which it creates in the current working directory. The output specifies the tape capacity and EOF marker size for the tape in the specified tape drive.

The first few lines of output include status information about the execution of the command, including such information as the number of blocks and the number of file marks written to the tape by the command. More lines of output are written to the screen than to the *fms.log* file. However, the last two lines of both screen and file output provide the following information:

- "`Tape capacity is` *number* `bytes`" specifies the tape size, in bytes, for the tape drive.
- "`File marks are` *number* `bytes`" specifies the file mark size, in bytes, for the tape drive.

On machines running AIX 2.2.1, much more output is displayed on the screen (a separate line is displayed for each block written to the tape) than is displayed there on other machines. While information may scroll from the screen at a rapid pace while the command executes, the information that scrolls past is unimportant; important summary information is presented last. Note that only the customary number of lines are written to the *fms.log* file.

If a problem with the tape drive prevents execution of the command, no *fms.log* file is created; the message "`Can't open tape drive` *tape special file*" appears. If a problem prevents creation of the *fms.log* file, the message "`Can't open log file`" appears. In both cases, execution of the command stops when the message appears.

EXAMPLES

The following determines the capacity and the EOF mark size for the tape drive whose device name is */dev/rmt1h*:

% **fms /dev/rmt1h**

The command displays the following output on the screen:

```
wrote block: 130408
Finished data capacity test - rewinding
wrote 1109 blocks, 1109 file marks
Finished file mark test
Tape capacity is 2136604672 bytes
File marks are 1910205 bytes
```

It writes the following information to the *fms.log* file:

```
fms test started
wrote 9230 blocks
Finished file mark test
Tape capacity is 151224320 bytes
File marks are 2375680 bytes
```

PRIVILEGE REQUIRED

If an *fms.log* file does not already exist in the current directory, the issuer must be able to insert and write to files in the directory. If the log file already exists in the current directory, the issuer need only be able to write to the file.

kaserver — initialize Authentication Server.

/usr/afs/bin/kaserver [-noAuth] [-fastKeys] [-database *dbpath***] [-localfiles** *lclpath***] [-minhours** <*n*>**] [-servers** <*serverlist*>**]**

ACCEPTABLE ABBREVIATIONS

This command does not use the syntax conventions of the AFS command suites, so the command and its switches and flags must be typed in full.

DESCRIPTION

Initializes the Authentication Server, which runs on database server machines. The Authentication Server is responsible for several aspects of AFS security including:

- Maintenance of all AFS server encryption keys and user passwords in the Authentication Database.
- Creation of the *tickets*, via its Ticket Granting Service (TGS), that users and servers use to establish secure connections.

Note: This command is not normally issued at the command shell prompt but rather placed into a file server machine's */usr/afs/local/BosConfig* with the **bos create** command. If it is ever issued at the command shell prompt, the issuer must be working on a file server machine.

The Authentication Server creates a log file in */usr/afs/logs/AuthLog*. In this file, the Authentication Server logs information on actions, such as the creation of users, that involve interaction with the Authentication Server. This file can be read using the **bos getlog** command; the **kdb** command is used to read the protected files (*AuthLog.dir* and *AuthLog.pag*) associated with *AuthLog*.

ARGUMENTS

-noAuth tells the Authentication Server to assign the unprivileged identity *anonymous* to the issuer. Thus, it establishes an unauthenticated connection between the issuer and the Authentication Server. It is useful only when authorization checking is disabled on the database server machine. In normal circumstances, the Authentication Server allows only authorized (privileged) users to issue commands that affect or contact the Authentication Database and will refuse to perform such an action even if the **-noAuth** flag is used.

-fastKeys is a Transarc Corporation internal test flag; it serves no functional purpose.

-database specifies a path that the Authentication Server uses when accessing the Authentication Database; rather than the default of */usr/afs/db/kaserver*. If you use this argument, you should also use the **-localfiles** argument; otherwise, *lclpath* will be set to the path specified by this argument.

-localfiles specifies a path that the Authentication Server uses when accessing its local files; the default is */usr/afs/local/kaserver*. If you use the **-database** argument, you should also use the **-localfiles** argument; otherwise, *lclpath* will be set to the path specified by **-database**.

-minhours specifies the minimum number of hours that must pass between consecutive password changes made by any regular user. System Administrators (with the `ADMIN` flag in their Authentication Database entry) can change passwords as often as desired. Setting a minimum time between password changes is not recommended.

-servers tells the Authentication Server to use the server(s) specified with this argument rather than using those listed in the local version of *CellServDB*.

EXAMPLES

The following **bos create** command creates a **kaserver** process on *fs3.transarc.com*. Type the command on one line:

```
% bos create  fs3.transarc.com  kaserver  simple  /usr/afs/bin/kaserver
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

MORE INFORMATION

bos create

bos getlog

kdb

kdb — read logs associated with the Authentication Server.

kdb [-dbmfile <*dbmfile to use (default /usr/afs/logs/AuthLog)*>**]**
[-key <*extract entries that match specified key*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

kdb [-dbmfile <*dbmfile to use (default /usr/afs/logs/AuthLog)*>**]**
[-key <*extract entries that match specified key*>**] [-help]**

DESCRIPTION

Allows command issuer to read the protected *AuthLog.dir* and *AuthLog.pag* files associated with the file */usr/afs/logs/AuthLog*. These files contain information on privileged actions associated with the Authentication Database. The binary file for this command is */usr/afsws/etc/kdb*.

Note: The **kdb** command is not available on the supported Hewlett-Packard system types (*hp300_ux90*, *hp700_ux90*, or *hp800_ux90*) or on the *rt_aix221* system type.

ARGUMENTS

-dbmfile	specifies the complete pathname of the file to be displayed. By default, the command looks for the specified file in the */usr/afs/logs/AuthLog* directory. If the file to be displayed resides in a different directory, specify either the full pathname of the file or a relative pathname from the default directory.
-key	specifies each entry to be displayed from the indicated file.
-help	prints the online help for this command. Do not provide any other arguments with this one.

OUTPUT

Each entry in the output of the **kdb** command contains the following two fields:

- **user/server**: Identifies the user requesting the corresponding service and the server that performed that service. In cases where no user is directly involved, only the server is listed; in cases where no server is directly involved, only the user is listed.
- **service**: Identifies the service performed by the corresponding server. There are seven services that will be logged to this list:
  - auth: Got a ticket-granting ticket.
  - chp: Changed a password for a user.
  - cruser: Created an Authentication Database entry for a user.
  - delu: Deleted an Authentication Database entry for a user.
  - gtck: Got a ticket other than a ticket-granting ticket.

– setf: Set the fields for an Authentication Database entry.

– unlok: Unlocked an Authentication Database entry for a user.

EXAMPLES

The following shows the output of the **kdb** command in the Transarc Corporation cell (transarc.com):

% **kdb**

```
admin,krbtgt.TRANSARC.COM:auth
admin,afs:gtck
admin:cruser
admin:delu
```

PRIVILEGE REQUIRED

Issuer must be logged into the machine's UNIX file system as "root."

MORE INFORMATION

kaserver

bos getlog

klog — authenticate with Authentication Server to obtain token.

klog [-x] [-principal <*user name*>**] [-password** <*user's password*>**] [-tmp] [-cell** <*cell name*>**] [-servers** <*explicit list of servers*>$^{+}$**] [-pipe] [-lifetime** <*ticket lifetime in hh[:mm[:ss]]*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

klog [-x] [-pr <*user name*>**] [-pa** <*user's password*>**] [-t] [-c** <*cell name*>**] [-s** <*explicit list of servers*>$^{+}$**] [-pi] [-l** <*ticket lifetime in hh[:mm[:ss]]*>**] [-h]**

DESCRIPTION

Authenticates *user name* in the specified cell. By default, the issuer of the command is *user name*. If the issuer of the command successfully identifies *user's password*, he or she obtains a token accepted by the AFS server processes in that cell. This command does not change the identity under which the issuer is logged into the local UNIX file system.

The Cache Manager acting on behalf of *user name* stores the token in a credential structure associated with *user name*. If *user name* already has a token for the cell, the token resulting from this command replaces the existing token in the credential structure.

If a user receives the message `Unable to authenticate to AFS because ID is locked - see your system admin` it indicates that someone has attempted to authenticate with the user's account and has exceeded the authentication failure limit. Use the **kas unlock** command to unlock the user account.

Note: The **klog.krb** version of this command is intended for use by sites that employ standard Kerberos authentication for their clients. The **klog.krb** command provides all of the functionality of **klog**. In addition, **klog.krb** will automatically place the user's Kerberos tickets in the file specified by the KRBTKFILE environment variable. The KRBTKFILE environment variable is automatically defined by **pagsh.krb** to be */tmp/tktpX* (where *X* is the number of the user's PAG).

By default, the token generated by **klog** is appropriate for the local cell (the one to which the local machine belongs). The command interpreter contacts an Authentication Server in the local cell, chosen at random from the list in */usr/vice/etc/CellServDB*, and requests a token for the issuer logged into the local machine.

Use the **-principal** argument to acquire a token for an *user name* other than the name under which you are identified by the system. Use the **-cell** argument to acquire a token for a cell other than the local cell. Use the **-servers** argument to specify a specific list of authentication servers to be used by the command interpreter to obtain the token.

The issuer (or user indicated with **-principal**) does not have to appear in the local password file (*/etc/passwd* or equivalent) to issue this command; in previous versions of this command, users had to add the **-x** flag if they did not appear in that file.

During a single login on a given machine, a user can be authenticated in multiple cells simultaneously; however, a single user can have only one token at a time for a given cell (that is, can only authenticate under one identity per cell).

The lifetime of the token resulting from this command is the smallest of the following:

- The lifetime requested by the issuer with the **-lifetime** argument. If the issuer does not include this argument, the value defaults to 720 hours (30 days).
- The "maximum ticket lifetime" recorded in the "*afs*" entry in the Authentication Database. The default is 100 hours. Administrators can inspect this value using **kas examine**, and change it using **kas setfields**.
- The "maximum ticket lifetime" recorded in the issuer's Authentication Database entry. The default is 25 hours for user entries created by the AFS 3.1 or later versions of the Authentication Server, and 100 hours for user entries created by the AFS 3.0 version of the Authentication Server. Administrators and the user himself/herself can inspect this value using **kas examine**, and administrators can change it using **kas setfields**.
- The "maximum ticket lifetime" recorded in the "*krbtgt.**CELLNAME***" entry in the Authentication Database; this entry corresponds to the ticket-granting ticket used internally in generating the token. The default is 720 hours (30 days).

If none of these defaults have been changed, then the standard token lifetime is 25 hours for users whose Authentication Database entries were created by the AFS 3.1 or later version of the Authentication Server, and 100 hours for users whose Authentication Database entries were created by the AFS 3.0 version of the Authentication Server. The user can issue **klog** to request a token with a different lifetime.

The maximum lifetime for any token is 720 hours (30 days), and the minimum is 5 minutes. Between these values, token lifetimes are not granted on a linear scale; only certain values are possible.

Lifetimes between 5 minutes and 10 hours 40 minutes are granted at 5 minute intervals, rounding up. For example, if the issuer requests a lifetime of 12 minutes, the token's actual lifetime is 15 minutes.

For token lifetimes greater than 10 hours 40 minutes, consult the following table, which presents all the possible times in units of *hours:minutes:seconds*. The number in parentheses is an approximation of the corresponding time in days and hours (*d h*). For example, 282:22:17 means 282 hours, 22 minutes, and 17 seconds, which translates to approximately 11 days and 18 hours (11d 18h).

If the issuer requests a lifetime between two listed values, the token's lifetime is rounded up to the next higher value. For example, if 11:24:16 was specified, the **klog** command would use 12:11:34.

11:24:15 (0d 11h)	33:14:21 (1d 09h)	96:52:49 (4d 00h)	282:22:17 (11d 18h)
12:11:34 (0d 12h)	35:32:15 (1d 11h)	103:34:45 (4d 07h)	301:53:45 (12d 13h)
13:02:09 (0d 13h)	37:59:41 (1d 13h)	110:44:28 (4d 14h)	322:46:13 (13d 10h)
13:56:14 (0d 13h)	40:37:19 (1d 16h)	118:23:54 (4d 22h)	345:05:18 (14d 09h)
14:54:03 (0d 14h)	43:25:50 (1d 19h)	126:35:05 (5d 06h)	368:56:58 (15d 08h)
15:55:52 (0d 15h)	46:26:01 (1d 22h)	135:20:15 (5d 15h)	394:27:37 (16d 10h)
17:01:58 (0d 17h)	49:38:40 (2d 01h)	144:41:44 (6d 00h)	421:44:07 (17d 13h)
18:12:38 (0d 18h)	53:04:37 (2d 05h)	154:42:01 (6d 10h)	450:53:46 (18d 18h)
19:28:11 (0d 19h)	56:44:49 (2d 08h)	165:23:50 (6d 21h)	482:04:24 (20d 02h)
20:48:57 (0d 20h)	60:40:15 (2d 12h)	176:50:01 (7d 08h)	515:24:22 (21d 11h)
22:15:19 (0d 22h)	64:51:57 (2d 16h)	189:03:38 (7d 21h)	551:02:38 (22d 23h)
23:47:38 (0d 23h)	69:21:04 (2d 21h)	202:08:00 (8d 10h)	589:08:45 (24d 13h)
25:26:21 (1d 01h)	74:08:46 (3d 02h)	216:06:35 (9d 00h)	629:52:56 (26d 05h)
27:11:54 (1d 03h)	79:16:23 (3d 07h)	231:03:09 (9d 15h)	673:26:07 (28d 01h)
29:04:44 (1d 05h)	84:45:16 (3d 12h)	247:01:43 (10d 07h)	
31:05:22 (1d 07h)	90:36:53 (3d 18h)	264:06:34 (11d 00h)	

WARNING

This command does not create a new "process authentication group" (commonly abbreviated PAG); see the description of the **pagsh** command in this chapter to learn about PAGs. Users in cells not using the AFS version of **login** should issue **pagsh** before issuing this command, so that the tokens get stored in a credential structure that is identified by PAG rather than UNIX UID.

A potential security problem exist when a token is associated with a credential structure identified by a UNIX UID because anyone logged in as "root" can assume any other identity by issuing the **su** command. And when "root" assumes another UNIX UID it can use use any tokens associated with that UID. Use of a PAG as an identifier eliminates this problem.

If the issuer entered the current session by issuing the AFS **login** command, his or her credential structure is already identified by a PAG. Issuing **klog** during the same session creates a new token associated with the existing PAG.

ARGUMENTS

-x appears only for backwards compatibility. Its former function is now the default behavior of this command, as mentioned in the DESCRIPTION section.

-principal is the user name under which the issuer wishes to authenticate. By default, the Authentication Server attempts to authenticate the user logged into the local machine's UNIX file system. This argument allows the issuer to specify a different user name.

-password specifies the issuer's password (or that of *user name* if **principal** is provided). Use of this argument is strongly discouraged, as it makes the password visible on the command line. If the issuer omits this argument, **klog** prompts for the password and does not echo it visibly:

```
Password: <user's password>
```

-tmp indicates that a copy of the Kerberos tickets should be placed in a file on the local machine's */tmp* directory in a file specified by the environment variable KRBTKFILE. If KRBTKFILE is undefined, a file named */tmp/tktU* is used (where *U* is the user's UNIX UID (For example for user with UNIX UID 1000, the file would be named */tmp/tkt1000*).

Putting the Kerberos tickets into a file in */tmp* allows standard Kerberos applications to access and use them for authentication purposes. This flag should not be used with the **klog.krb** version because it automatically writes the Kerberos tickets to the file specified by KRBTKFILE.

-cell specifies the cell in which the issuer wishes to authenticate, by directing the command to that cell's Authentication Servers. During a single login on a given machine, a user may be authenticated in multiple cells simultaneously, but can have only one token at a time for each of them (that is, can only authenticate under one identity per cell per machine). The issuer may abbreviate *cell name* to the shortest form that distinguishes it from the other cells listed in */usr/vice/etc/CellServDB* on the client machine on which the command is issued.

If this argument is omitted, the command is executed in the local cell, as defined

- First, by the value of the environment variable AFSCELL.
- Second, in */usr/vice/etc/ThisCell* on the client machine on which the command is issued.

-servers causes the command interpreter to establish a connection with the Authentication Server running on each specified database server machine. It then chooses one of these at random to execute the command. The command accepts shortened machine names, but exactly which abbreviations are acceptable depends on the state of the cell's name server at the time the command is issued. This option is useful for testing specific servers if problems are encountered.

If this argument is omitted, the command interpreter establishes a connection with each machine listed for the indicated cell in the local workstation's copy of */usr/vice/etc/CellServDB*, and then chooses one of those at random for command execution.

-pipe indicates that the command should run without printing anything on the screen, including prompts or error messages. The **klog** command interpreter Server expects to receive the password from standard input (stdin). The issuer is discouraged from using this argument; it is for use by application programs rather than human users.

-lifetime indicates the lifetime that the issuer wishes the token to have. The value provided is considered in the lifetime calculation described in the DESCRIPTION section, along with the maximum ticket lifetimes mentioned there.

-help prints the online help for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

Most often, this command is issued without arguments. The appropriate password is for the person currently logged into the local UNIX file system. The ticket's lifetime is calculated as described in the DESCRIPTION section above (if no defaults have been changed, it is 25 hours for a user whose Authentication Database entry was created by the AFS 3.1 or later version of the Authentication Server, 100 hours for a user whose Authentication Database entry was created with the AFS 3.0 version).

```
% klog
Password:
```

The following allows the issuer working on a machine in the Transarc cell to authenticate as *admin* in the Transarc test cell, even though he or she is logged into the Transarc machine under a different name.

```
% klog admin -c test.transarc.com
Password: <admin's password>
```

In the following, the issuer requests a ticket lifetime of 104 hours 30 minutes (4 days 8 hours 30 minutes). Presuming that this lifetime is allowed by the maximum ticket lifetimes and other factors described in the DESCRIPTION section, the token will have an actual lifetime of 110:44:28, which the next largest possible value.

```
% klog -life 104:30
Password:
```

PRIVILEGE REQUIRED

None. An entry for the issuer must exist in the Authentication Database, and the issuer must supply the correct password.

MORE INFORMATION

login

kas unlock

pagsh

tokens

knfs — enable authenticated access to AFS from non-supported NFS client using NFS/AFS Translator.

knfs -host <*host name*> **[-id** <*user ID (decimal)*>**]**
[-sysname <*host's '@sys' value*>**] [-unlog] [-tokens] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

knfs -ho <*host name*> **[-i** <*user ID (decimal)*>**] [-s** <*host's '@sys' value*>**] [-u] [-t] [-he]**

DESCRIPTION

Allows a user to authenticate with AFS and work in the AFS file space while working on an NFS client machine – even when using a system type for which AFS binaries are not available. If AFS defines a system type for the corresponding NFS client, the authenticated user(s) on that machine will be able to access the AFS file tree and issue AFS commands (by using the **-sysname** argument). It AFS does not define a system type for the NFS client, the authenticated user(s) on that machine will be able to access the AFS file tree, but will not be able to issue AFS commands.

To properly execute this command, establish a connection to the machine that will act as the NFS/AFS translator machine for the NFS client machine (*host name*). Next, issue the **klog** command to obtain AFS tokens on the translator machine for every cell that will be contacted from the NFS client machine. Finally, issue the **knfs** command specifying the NFS client machine with the **-host** argument, and optionally the specific user who will use the NFS client to access AFS with the **-id** argument.

Issue this command only over a connection (perhaps established at the console or via **telnet**) to the AFS client machine serving as the NFS/AFS translator machine for *host name*, never on the NFS client machine itself. The Cache Manager on the translator machine then uses the credential structure identified by the new PAG when accessing AFS on behalf of the issuer working on the NFS client machine. (If the issuer does not use the **-id** argument, he or she may actually be sharing the credential structure with other users on the NFS client machine; see the WARNINGS section for details.)

When issued on an NFS/AFS Translator machine, this command creates a PAG associated with the indicated NFS client machine (*host name*), and optionally, the user ID (*user ID*) that the issuer is using on *host name*. It associates the issuer's existing AFS token(s) with a new credential structure identified by the new PAG.

The **-unlog** flag discards the tokens in the credential structure identified by the PAG, but does not destroy the credential structure itself. The Cache Manager on the translator machine retains the credential structure until the next reboot, and uses it each time the issuer accesses AFS through the translator machine. The credential structure only has tokens in it if the issuer re-issues **knfs** on the translator machine each time he or she logs into the NFS client machine.

Users working on NFS client machines of system types for which AFS binaries are available (and for which the cell has purchased a license) should use **klog** rather than the **knfs** command.

WARNINGS

Although the **-id** argument is optional, not using it can create potential security problems and confusion about which users are using which tokens. The security problems arise because if the issuer does not use **-id**, then his or her tokens are associated with a "generic" credential structure identified by a PAG that is associated only with *host name*, not with the specific issuer. Every user working on *host name* who also does not use the **-id** argument shares the same generic credential structure and so uses the issuer's tokens. This includes users on *host name* who do not issue **knfs** at all. If another user subsequently issues **knfs** without **-id**, his or her tokens replace the previous issuer's tokens in the credential structure, and everyone shares the new tokens instead. Similarly, when an issuer discards tokens with the **-unlog** flag and does not use **-id**, then everyone using the generic credential structure becomes unauthenticated at the same time.

The sharing of tokens that result from not using **-id** is acceptable from a security standpoint only if

- The *host name* is used by only one person.
- It is acceptable that all of *host name*'s users who do not have their own credential structure have the same access to AFS files. They should be aware that they are subject to token sharing.

The DESCRIPTION section mentioned that using the **-unlog** flag does not destroy the credential structure, but only discards the tokens associated with it. The Cache Manager on the translator machine retains the credential structure until the next reboot and uses it whenever the issuer accesses AFS through the translator machine. One implication is that once the issuer issues **knfs** using the **-id** argument, he or she cannot use the generic credential structure until the machine is rebooted.

This command does not make it possible for users working on non-supported machine types to issue AFS commands. This is possible only on NFS clients of a system type for which AFS is available (and for which the cell has purchased AFS).

ARGUMENTS

-host	names the NFS client machine the issuer is working on. A full name is safest, but abbreviated forms are acceptable depending on the state of the cell's name server at the time the command is issued.
-id	specifies the issuer's user ID on the NFS client (a UNIX UID or equivalent), which NFS passes to the translator machine to identify the user.
-sysname	specifies the value of the *@sys* variable if the NFS client machine is running a defined AFS operating system (that is, one for which AFS binaries are available). This allows users on NFS client machines using defined system types to issue AFS commands.
-unlog	discards the tokens in the credential structure identified by the PAG associated with **-host** and, optionally, **-id**.
-tokens	allows users who do not have access to AFS binaries to obtain information about their AFS tokens.

-help prints the online help for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

The following shows a typical use of this command. The issuer *smith* is working on *nfs-client1.transarc.com* and has user ID 1020 on that machine. He is accessing AFS through the translator machine called *translator4.transarc.com*. He telnets to *translator4* and uses AFS **login** as *smith* in the Transarc cell. He uses **klog** to obtain tokens in the Transarc test cell (*test.transarc.com*) as *admin*. The **knfs** command instructs the Cache Manager on *translator4* to associate both tokens with the PAG identified by *nfs-client1* and user ID *1020*. He exits the telnet connection and works normally on *nfs-client1*.

```
% telnet translator4.transarc.com
. . .
login: smith
Password:
AFS 3.2 (R) login
% klog admin -c test.transarc.com
Password:
% knfs nfs-client1.transarc.com 1020
% exit
```

The following shows *smith* telnetting again to *translator4* and discarding his tokens.

```
% telnet translator4.transarc.com
. . .
login: smith
Password:
AFS 3.2 (R) login
% knfs nfs-client1.transarc.com 1020 -unlog
% exit
```

PRIVILEGE REQUIRED

None, but the issuer must provide correct password(s) when obtaining tokens on the NFS/AFS translator machine.

MORE INFORMATION

klog

login (AFS version)

pagsh

kpasswd — change password in Authentication Database.

kpasswd [-x] [-principal <*user name*>**] [-password** <*user's password*>**]**
[-newpassword <*user's new password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of servers*>**+] [-pipe] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

kpasswd [-x] [-pr <*user name*>**] [-pa** <*user's password*>**] [-n** <*user's new password*>**]**
[-c <*cell name*>**] [-s** <*explicit list of servers*>**+] [-pi] [-h]**

DESCRIPTION

Changes password for the indicated user in the Authentication Database. By default, the change occurs in the local cell's Authentication Database for the user logged into the local machine's UNIX file system.

The issuer (or user indicated with **-principal**) does not have to appear in the local password file (*/etc/passwd* or equivalent) to issue this command; in previous versions of this command, users had to add the **-x** flag if they did not appear in that file.

During execution, this command will make the following checks:

- If the program *kpwvalid* exists in the same directory as **kpasswd**, it will be executed using both the old and the new password as input. The *kpwvalid* program can either be the example program provided by Transarc Corporation, or a program customized for your cell. Custom *kpwvalid* programs can be written in any compatible programming language or *kpwvalid* can be a shell script.

  To protect against the risk of Trojan horses, *kpwvalid* must reside in the same directory as **kpasswd**. The directory must be in AFS and must permit write access to *system:administrators* only.

  A custom *kpwvalid* program or shell script must follow the general behavior of the example program provided by Transarc (shown below); however, the exact checks made by a custom version of *kpwvalid* are dependent only on the needs and conventions of your cell. A customized program or shell script must write a zero to standard output (stdout) followed by a newline if the new password is valid; otherwise it will write a non-zero decimal number to stdout. Any error messages generated by *kpwvalid* must be directed to standard error (stderr) only; do not direct error messages to stdout.

  Below is the content of the example *kpwvalid* program provided by Transarc. This program checks to make sure that a new password contains eight or more characters:

```
#include <stdio.h>

/* returns 0 if the password is long enough, otherwise non-zero */
main()
{
char oldpassword[512];
char password[512];
int rc;

if (fgets(oldpassword, 512, stdin))
   while (fgets(password, 512, stdin)) {
      if (strlen(password) > 8) { /* password includes a newline */
         rc = 0;
         fputs("0\n",stdout);
         fflush(stdout);
      }
      else {
         rc = 1;
         fputs("Passwords must contain at least 8 characters.\n",
               stderr);
         fputs("1\n",stdout);
         fflush(stdout);
      }

exit(rc);
}
```

- If the user is prohibited from reusing old passwords, **kpasswd** will check to make sure that the new password does not match any of the last twenty passwords. The **kas setfields** command has an argument that allows you to prohibit the reuse of the last twenty old passwords.

  If a user attempts to create a password that is the same as or similar to a recently used password, the following error message will be displayed:

  ```
  kpasswd:  Password was not changed because it seems like a reused
  password
  ```

  Note: To prevent users from getting around this restriction by simply changing their passwords twenty times (or running a script which automatically does this), you can set a minimum time between password changes when **kaserver** is started. If a user attempts to change a password before the minimum time between password changes has passed, the following error message will be displayed:

  ```
  kpasswd:  Password was not changed because you changed it too
  recently; see your systems administrator
  ```

ARGUMENTS

-x	appears only for backwards compatibility. Its former function is now the default behavior of this command, as mentioned in the DESCRIPTION section.
-principal	names the Authentication Database entry in which to change the password. If this argument is omitted, the change is in the entry of the person logged into the local machine's UNIX file system.

-password specifies the current password. It is not recommended that the issuer provide this argument on the command line. If it is omitted, the new password is prompted for and does not echo visibly:

```
Old password: <user's password>
```

-newpassword specifies the new password, which the **kpasswd** command interpreter converts into an encryption key (string of octal numbers) before sending it to the Authentication Server for storage in the user's Authentication Database entry. Unlike the UNIX **passwd** command, **kpasswd** does not restrict passwords to 8 characters; it accepts passwords of virtually any length. All AFS commands that require passwords (**klog**, the **kas** suite, **kpasswd**, and the AFS version of **login**) can handle passwords longer than 8 characters, but some non-AFS programs cannot. This is a consideration if non-AFS programs handle AFS passwords in the local environment.

It is not recommended that the issuer provide this argument on the command line. If it is omitted, it is prompted for and does not echo visibly:

```
New password (RETURN to abort): <new password>
Retype new password: <new password>
```

-cell specifies the cell in which to change the password, by directing the command to that cell's Authentication Servers. The issuer may abbreviate *cell name* to the shortest form that distinguishes it from the other cells listed in */usr/vice/etc/CellServDB* on the client machine on which the command is issued.

By default, the command is executed in the local cell, as defined

- First, by the value of the environment variable AFSCELL.
- Second, in */usr/vice/etc/ThisCell* on the client machine on which the command is issued.

-servers causes the command interpreter to establish a connection with the Authentication Server running on each specified database server machine. It then chooses one of these at random to execute the command. The issuer may abbreviate the machine name to the extent the cell's name server will accept.

By default, the command interpreter establishes a connection with each machine listed for the indicated cell in the local workstation's copy of */usr/vice/etc/CellServDB*, and then chooses one of those at random for command execution.

This option is useful for testing specific servers if problems are encountered.

-pipe indicates that the command should run without printing anything on the screen, including prompts or error messages. The **kpasswd** command interpreter expects to receive all necessary arguments, each on a separate line, from standard input (stdin). The issuer is discouraged from including this argument; it is for use by application programs rather than human users.

-help prints the online help for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

The following shows the typical use of this command, when the issuer wishes to change his or her own password.

```
% kpasswd
Changing password for 'user' in cell 'cellname'.
Old password:
New password (RETURN to abort):
Verifying, please re-enter new_password:
```

PRIVILEGE REQUIRED

Issuer must provide correct current password.

MORE INFORMATION

klog

kas setfields

kas setpassword

kpwvalid

package_test — test the validity of a package configuration file.

package_test <*config file*>

ACCEPTABLE ABBREVIATIONS/ALIASES

This command must be typed in full.

DESCRIPTION

Tests the validity of a package configuration file created when a prototype file is compiled. When the command encounters an error, a message is printed to standard output.

ARGUMENTS

config file specifies the package configuration file that you want to validate.

EXAMPLES

The following tests the validity of the **package** configuration file *staff.sun4c_411*.

% package_test staff.sun4c_411

PRIVILEGE REQUIRED

None.

MORE INFORMATION

package

pagsh — create new process authentication group (PAG).

pagsh

DESCRIPTION

Creates a new command shell (owned by the issuer of the command) and associates a new "process authentication group" (PAG) with the shell and the user. A PAG is a number guaranteed to identify the issuer of commands in the new shell uniquely to the local Cache Manager. The PAG is used, instead of the issuer's UNIX UID, to identify him or her in the credential structure that the Cache Manager creates to track each user.

Any additional tokens issued to the user (presumably for other cells) become associated with the PAG, rather than with the user's UNIX UID. This method for distinguishing users has two advantages.

- It means that processes spawned by the user inherit the PAG and so share the token; thus they gain access to AFS as the authenticated user. In many environments, for example, printer and other daemons run under identities (such as "root") that the AFS server processes recognize only as *anonymous*. Unless PAGs are used, such daemons cannot access files protected against *system:anyuser*.
- It closes a potential security loophole: UNIX allows anyone already logged in as "root" on a machine to assume any other identity by issuing **su**. If the credential structure were identified by a UNIX UID rather than a PAG, then a root user could assume a UNIX UID and use any tokens associated with that UID. Use of a PAG as an identifier eliminates that possibility.

Note: The **pagsh.krb** version of this command is intended for use by sites that employ standard Kerberos authentication for their clients. The **pagsh.krb** command provides all of the functionality of **pagsh**. In addition, it defines the environment variable KRBTKFILE (which specifies the storage location of Kerberos tickets) to be */tmp/tktpX* (where *X* is the number of the user's PAG). The functionality of this command supports the placement of Kerberos tickets by **login.krb** and **klog.krb** in the file specified by KRBTKFILE.

WARNINGS AND REQUIREMENTS

Each PAG created uses two of the memory slots that the kernel uses to record the UNIX groups associated with a user. If none of these slots are available, the **pagsh** command fails. This is not a problem with most operating systems, which make at least 16 slots available per user.

Users in cells not using the AFS version of **login** should issue this command before issuing **klog**. If they do not, then the Cache Manager stores the token in a credential structure identified by UNIX UID rather than PAG. This creates the potential security loophole described in the DESCRIPTION section.

If the command is to be issued on NFS client machines for which AFS is available, the **pagsh** binary accessed by the NFS client must be owned by, and grant setuid privilege to, "root." The complete set of mode bits should be `-rwsr-xr-x`. This is not a requirement when the command is issued on AFS client machines.

EXAMPLES

The following shows the only possible use of the command.

```
% pagsh
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

klog

login (AFS version)

tokens

ptserver — initialize Protection Server.

/usr/afs/bin/ptserver [-database <*db path*>**] [-rebuild]**

ACCEPTABLE ABBREVIATIONS/ALIASES

This command does not use the syntax conventions of the AFS command suites, so the command and its switches and flags must be typed in full.

DESCRIPTION

Initializes the Protection Server, which runs on database server machines. The Protection Server is responsible for

- Maintaining the Protection Database which contains entries for every user and group in the cell.
- Allocating AFS UIDs for new users and groups and mapping each UID to its corresponding user/group name.
- When requested by the File Server, identifying whether or not a user's identity or group membership allows that user access to a file or directory in the requested manner.

Note: This command is not normally issued at the command shell prompt, but rather placed into a database server machine's */usr/afs/local/BosConfig* with the **bos create** command. If it is ever issued at the command shell prompt, the issuer must be working on a database server machine.

ARGUMENTS

-database	specifies a database for the Protection Server to use rather than the default (typically, */usr/afs/db/prdb*).
-rebuild	directs the Protection Server to rebuild the Protection Database prior to assuming its normal responsibilities. This is an internal Transarc Corporation debugging tool and should not be used.

EXAMPLES

The following **bos create** command creates a **ptserver** process on *fs3.transarc.com*. Type the command on one line:

```
% bos create  fs3.transarc.com  ptserver  simple  /usr/afs/bin/ptserver
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

Issuer must be logged into the machine’s UNIX file system as "root" to run this command from the command line.

MORE INFORMATION

bos create

bos getlog

runntp — initialize Network Time Protocol Daemon.

/usr/afs/bin/runntp [-localclock] [-precision <*small negative integer*>**]**
[-logfile <*pathname*>**] [**<*machine name*>**+]**

DESCRIPTION

Initializes the Network Time Protocol Daemon and related programs on the local machine. It also constructs the *ntp.conf* configuration file.

This command is intended as a convenient interface to the standard **ntpd** program, to be used on AFS file server machines.

WARNING

A cell in which the NTPD was running before the introduction of AFS does not need to use **runntp**. It is an error to run two NTPDs on one machine.

This command is not normally issued at the command shell prompt, but rather placed into a file server machine's */usr/afs/local/BosConfig* with the **bos create** command.

This command does not use the syntax conventions of the AFS command suites, so the command and switches must be typed in full.

ARGUMENTS

-localclock tells NTPD to use the local machine's internal clock as a possible source of the correct time in case a network partition separates the machine from the specified *machine name*(s). Cells connected to the Internet should not normally use this flag. In cells that experience frequent separations from the network (voluntary or otherwise), it should be used only on the system control machine.

-precision specifies the precision of the local clock. **This argument is not normally provided.** As **ntpd** initializes, it determines the precision of the local clock on its own. If the argument is provided, it should be a small integer preceded by a hyphen to show that it is negative. The value is used as an exponent on the number 2, and the result interpreted as the frequency, in fractions of a second, at which the local clock ticks (advances).

For example, a value of *-6*, which translates to 2^{-6} or 1/64, means that the local clock ticks once every 1/64th of a second, or has a precision of about 60 ticks per second. A value of *-7* translates to about 100 ticks per second. A value of *-10* translates to about 1000 ticks per second (a millisecond clock).

-logfile indicates the pathname of the file in which to store log output from NTPD. Indicate a location on the file server machine's local disk, not in AFS; a suitable location might be */usr/afs/logs/ntp.log*. The log records information about which machines are serving as time sources and peers, what adjustments have been made to reduce drift, and so on. Use **ntpd**'s debugging mechanism to control the amount of information produced. If this argument is omitted, the information is discarded.

machine name is the complete Internet-style host name of each machine the local machine should consult for the correct time. Preferably the machines are outside the cell.

In general, this argument is necessary only on the system control machine. If the issuer omits it, then the local machine consults its */usr/afs/etc/CellServDB* file and uses the machines listed there as time sources.

See the *AFS Installation Guide* or consult AFS Product Support at Transarc for advice on selecting appropriate time sources.

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

MORE INFORMATION

bos create

UNIX manual page for **ntp**

UNIX manual page for **ntpd**

UNIX manual page for **ntpdc**

salvager — initialize Salvager in *BosConfig* or manually.

/usr/afs/bin/salvager [-initcmd] [-partition <*Name of partition to salvage*>**]**
[-volumeid <*Volume Id to salvage*>**] [-debug]**
[-nowrite] [-inodes] [-force] [-oktozap]
[-rootinodes] [-salvagedirs] [-blockreads]
[-parallel <*# of max parallel partition salvaging*>**]**
[-tmpdir <*directory to place tmp files*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

This command does not use the syntax conventions of the AFS command suites, so the command and its switches and flags must be typed in full.

DESCRIPTION

Initializes the Salvager component of the *fs* process, which salvages (restores internal consistency in) corrupt ReadWrite volumes on the local file server machine where possible.

Note: Unlike most other AFS file server processes, this process can also be started from the command line. There is no limit to the number of volumes that the Salvager can salvage.

If the Salvager detects corruption in ReadOnly or Backup volumes, it removes them rather than attempting to correct the corruption; it records the removal in *SalvageLog*. If desired, the issuer can recreate the ReadOnly or Backup version by issuing **vos release** or **vos backup**, respectively.

By default, up to four Salvager processes execute to salvage four or more partitions; if fewer than four partitions are being salvaged, only that number of Salvager processes execute. To increase or decrease the number of Salvager processes that execute, use the **-parallel** argument to specify an integer value between *1* and *32*. Specify a value of *1* to run the Salvager processes sequentially (that is, one partition at a time).

The Salvager normally salvages only those ReadWrite volumes that are marked as having been active when a crash occurred. To have it salvage all relevant ReadWrite volumes, add the **-force** flag.

The number of volumes the Salvager attempts to salvage depends on the combination of arguments and flags used. To salvage

- All volumes on the file server machine, provide no arguments. A file system outage results because the BOS Server stops the File Server and Volume Server components while the Salvager is running.
- All the volumes on one partition of the file server machine, provide **-partition**. A file system outage results because the BOS Server stops the File Server and Volume Server components while the Salvager is running.
- Only one volume, combine **-partition** and **-volumeid**. Only the one volume goes off-line during the salvage. The BOS Server does not need to shutdown the File Server and Volume Server during this type of salvage, so there is no outage.

The Salvager creates *tmp* files as it goes about its task. The Salvager writes these files to the partition it is currently salvaging. Use the **-tmpdir** argument to redirect the destination of these files.

If the Salvager runs out of space to write its *tmp* files, its stops salvaging the corresponding partition. If a partition cannot be salvaged due to a lack of space, re-issue the **salvager** command using the **-tmpdir** argument to direct the files to a directory on another partition; the Salvager will automatically remove its *tmp* files from the full partition.

When executed, the Salvager creates an activity log file named */usr/afs/logs/SalvageLog*. This log file can be examined using the **bos getlog** command.

ARGUMENTS

initcmd is an optional string that accommodates the command's use of the AFS command parser. It can be omitted and ignored.

-partition specifies the name of the partition the Salvager is to inspect for corruption. Specify the partition name as */vicep***x**. Omit this option to inspect every partition on the file server machine.

-volumeid specifies the volume ID of a specific ReadWrite volume to be salvaged. If you specify a volume ID with this argument, you must also specify a partition name with **-partition**.

-debug allows for the effective use of the Salvager with a debugger. Use this flag to prevent parallel salvaging when running the Salvager in a debugger; it ensures that only one Salvager process is executing at a time.

-nowrite directs the Salvager to bring up all undamaged volumes, but refrain from salvaging the damaged volumes. This flag is useful if you need to salvage a full partition because during the salvage of damaged volumes, the Salvager creates new inodes; if the partition is full, the Salvager will be unable to write those inodes and the salvage will fail. By using this flag, you can bring up undamaged volumes (a task which does not require the creation of new inodes) and then move some of them to other partitions. This frees up space on the partition so that the Salvager can salvage the damaged volumes.

-inodes causes the Salvager to write to the */usr/afs/logs/SalvageLog* file a list of all AFS inodes that it had to modify in some way.

-force causes the Salvager to inspect all volumes for corruption, not just those that are marked as having been active when a crash occurred.

-oktozap enables the Salvager to remove volumes that are too damaged to be removed in the normal manner (using **vos zap** with **-force**). Use this argument only when directed by a member of the AFS Product Support staff. If this flag is provided, it should be combined with *-partition* and *-volumeid*.

-rootinodes causes the Salvager to write to the */usr/afs/logs/SalvageLog* file a list of all AFS inodes owned by root.

-salvagedirs forces the Salvager to salvage entire directory structures, regardless of whether it believes they are damaged. By default, the Salvager salvages a directory only if that directory is flagged as being corrupt.

-blockreads forces that Salvager to read a partition one disk block (512 bytes) at a time and to skip any blocks that are too badly damaged to be salvaged. This allows it to salvage as many volumes as possible. By default, the Salvager reads large disk blocks, which can cause it to exit prematurely if it encounters disk errors. Use this flag if the partition to be salvaged has disk errors.

-parallel specifies the maximum number of Salvager processes to be run in parallel. By default, four Salvager processes run if four or more partitions require salvaging; if fewer than four partitions require salvaging, a Salvager process will be run for each partition. Use this argument to increase or decrease this default number to a value between *1* and *32*. Specifying *1* runs the Salvager process sequentially; specifying a number greater than the number of partitions requiring salvage is unnecessary, because the Salvager runs only the number of processes necessary to salvage the specified number of partitions.

-tmpdir redirects the write destination of the *tmp* files created by the command to the specified *directory to place tmp files* on the machine being salvaged. If this argument is not used, the *tmp* files are placed on the partition being salvaged. If the *directory to place tmp files* is incorrectly specified or is not on the machine being salvaged, the *tmp* files are placed on the partition being salvaged. You may want to use this argument to indicate a specific directory large enough to accommodate the Salvager's *tmp* files if a partition to be salvaged has insufficient space to store the *tmp* files.

-help prints the online help for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

The following command instructs the Salvager to attempt to salvage the volume with volume ID 258347486 on */vicepg* on the local machine.

```
% /usr/afs/bin/salvager /vicepg 258347486
```

PRIVILEGE REQUIRED

Issuer must be logged into the local UNIX file system as "root" to issue this command to the shell.

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

MORE INFORMATION

bos create

bos getlog

bos salvage

scout — monitor File Server process on one or more file server machines.

scout [initcmd] -server <*FileServer name(s) to monitor*>+
[-basename <*base server name*>**]**
[-frequency <*poll frequency, in seconds*>**] [-host]**
[-attention <*specify attention (highlighting) level*>+**]**
[-debug <*turn debugging output on to the named file*>**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

scout [-initcmd] -s <*FileServer name(s) to monitor*>+ **[-b** <*base server name*>**]f**
[-f <*poll frequency, in seconds*>**] [-ho]**
[-a <*specify attention (highlighting) level*>+**]**
[-d <*turn debugging output on to the named file*>**] [-he]**

DESCRIPTION

Displays statistics gathered from the File Server process running on each machine specified with **-server**. The OUTPUT section explains the meaning of the statistics and describes how they appear on the screen.

REQUIREMENTS

In addition to the **scout** binary, the machine must be able to access the "**curses**" graphics package, since **scout**'s display abilities rely on it. Most UNIX distributions include **curses** as a standard utility.

Both "dumb" terminals and windowing systems that emulate terminals can display **scout**'s statistics. The **scout** display makes use of reverse video and cursor addressing, so for the display to look its best, the display environment should support those features (most windowing systems do, most dumb terminals do not). To let **scout** know what type of environment is being used, the user should set the environment variable `TERM` to the correct terminal type, or one with characteristics similar to the actual ones. For machines running AIX, the recommended setting for `TERM` is `vt100`, as long as the terminal is similar to that. For other operating systems, the wider range of acceptable values includes `xterm`, `xterms`, `vt100`, `vt200`, and `wyse85`.

ARGUMENTS

-server names each File Server process to monitor, by identifying the file server machine it is running on. Complete Internet-style host names are best, unless the **-basename** argument is used. In that case, specify only the unique initial part of each machine name, omitting the domain/cell name suffix (or "basename") common to all the names. Shortened unambiguous names are legal when the **-basename** argument is not used, but in that case their correct resolution depends on the state of the cell's naming service at the time the command is issued.

-basename specifies the basename (domain/cell name) suffix common to all of the file server machine names specified with **-server**, and is automatically appended to them. This

argument is normally the name of the cell to which the machines belong. Do not include the period that separates this suffix from the distinguishing part of each file server machine name, but do include any periods that occur within the suffix itself. (For example, in the Transarc Corporation cell, the proper value would be "*transarc.com*", not "*.transarc.com*".)

-frequency indicates how often **scout** should probe the File Server processes. Specify a number of seconds greater than *0*. The default is 60 seconds.

-host causes the banner line to display the name of the machine that is running **scout**. The default is not to display the machine name in the banner.

-attention defines a list of one or more entries, each of which pairs a type of statistic and a threshold value. When the value of the statistic exceeds the indicated threshold value, **scout** highlights the value. The pairs may appear in any order. Legal values for the pairs are

- **conn** <*connections*>. Indicates the maximum number of connections that the File Server may have open to client machines before the value is highlighted. Highlighting goes away when the value goes back below the threshold. By default, no threshold is set.

  Example of a legal value: *conn 300*

- **fetch** <*number of fetch RPCs*>. Indicates the maximum number of fetch RPCs that clients can make to the File Server before the value is highlighted. Highlighting goes away when the value goes back below the threshold, but this is possible only if the File Server is restarted, at which time the value returns to *0*. By default, no threshold is set.

  Example of a legal value: *fetch 6000000*

- **store** <*number of store RPCs*>. Indicates the maximum number of store RPCs that clients can make to the File Server before the value is highlighted. Highlighting goes away when the value goes back below the threshold, but this is possible only if the File Server is restarted, at which time the value returns to *0*. By default, no threshold is set.

  Example of a legal value: *store 200000*

- **ws** <*number of active client machines*>. Indicates the maximum number of "active" client machines the File Server can be serving before the value is highlighted. "Active" machines are defined as those that have communicated with the File Server in the last 15 minutes. Highlighting goes away when the value goes back below the threshold. By default, no threshold is set.

  Example of a legal value: *ws 65*

- **disk** <*percent full*>*%*. Indicates the maximum percentage of the disk that can be full before the value is highlighted; in that case legal values are the integers between *0* and *99*.

  Examples of legal values: *disk 90%* and *disk 72%*.

  or

  disk <*minimum blocks free*>. Indicates the minimum number of kilobyte blocks that must be available on the partition if the value is not to be highlighted.

Examples of legal values: *disk 1500* and *disk 5000*.

Specify one type of value or the other. Highlighting goes away when the value goes back below the percent threshold or above the blocks-free threshold. By default, the threshold is 95%.

-debug enables debugging output and directs it into the specified file. Providing a complete pathname is best; otherwise it is interpreted relative to the current working directory. By default, no debugging output is produced.

-help prints the online help for this command. Do not provide any other arguments or flags with this one.

OUTPUT

Scout can display statistics either in a dedicated window or on a plain screen if a windowing environment is not available. For best results, the window or screen should have the ability to print in reverse video.

The **scout** screen has three main parts: the banner line, the statistics display region and the message/probe line.

The Banner Line

By default, the word "`Scout`" appears in the banner line at the top of the window or screen, to indicate that **scout** is running. Two additional types of information may appear in the banner line if you include the appropriate flag or argument on the command line:

- The **-host** flag causes **scout** to display this machine name in the banner line. As mentioned previously, this is useful when you are running **scout** on several machines but displaying the results on a single machine.

  For example, if you run **scout** on the machine *client1.transarc.com* and use the **-host** flag, the banner line reads:

  ```
  [client1.transarc.com] Scout
  ```

- The **-basename** argument causes **scout** to display the indicated basename on the banner line. For example, if you specify a value of *transarc.com* for **-basename**, the banner line reads:

  ```
  Scout for transarc.com
  ```

The Statistics Display Region

In this region, which occupies the majority of the window, **scout** displays the statistics gathered for each File Server process. Each process appears on its own line.

The region is divided into six columns, labeled as indicated and displaying the following information:

- `Conn`: The first column displays the number of RPC connections open between the File Server process and client machines. This number should equal or exceed the number in the `Ws` column (see the fourth entry below). It can exceed the `Ws` number because each user on the machine can have several separate connections open at once, and one client machine can handle several users.

- `Fetch`: The second column displays the number of fetch type RPCs (fetch data, fetch access list, and fetch status) that client machines have made to the File Server process since the latter started. This number is reset to zero each time the File Server process restarts.
- `Store`: The third column displays the number of store type RPCs (store data, store access list, and store status) that client machines have made to the File Server process since the latter started. This number is reset to zero each time the File Server process restarts.
- `Ws`: The fourth column displays the number of client machines (workstations) that have communicated with the File Server process within the last 15 minutes (such machines are defined as "active"). This number is likely to be smaller than the number in the first (`Conn`) column because a single client machine can have several connections open to one File Server.
- The fifth, unlabeled, column displays the name of the file server machine on which the File Server process is running. Names of 12 characters or less are displayed in full; longer names are truncated and an asterisk (`*`) appears as the last character in the name. Using the **-basename** argument is a good way to avoid truncation, but only if all machine names end in a common string.
- `Disk attn`: The sixth column displays the number of available kilobyte blocks on each AFS disk partition on the file server machine. The display for each partition has the form

  `x:`*free_blocks*

  where `x` indicates the partition name. For example, `a:8949` specifies that partition */vicepa* has 8,949 1Kbyte blocks free. Available space may be displayed for up to 26 partitions. If the window is not wide enough for all partition entries to appear on a single line, **scout** automatically creates multiple lines, "stacking" the partition entries into sub-columns within the sixth column.

  The label on the `Disk attn` column indicates the threshold value at which entries in the column become highlighted. By default, the label is

  ```
  Disk attn: > 95% used
  ```

  because by default **scout** highlights the entry for any partition that is over 95% full.

For all columns except the fifth (file server machine name), you can use the **-attention** argument to set the value at which entries in the column should be highlighted to notify you that a certain value has been exceeded. Only the fifth and `Disk attn` columns ever become highlighted by default.

The Message/Probe Line

The bottom line of the **scout** screen indicates how many times **scout** has "probed" the File Server processes for statistics. The statistics gathered in the latest probe appear in the statistics display region. By default, **scout** probes the File Servers every 60 seconds, but you can use the **-frequency** argument to specify a different probe frequency in seconds.

EXAMPLES

See the chapter on **scout** in the *AFS System Administrator's Guide*, which illustrates the on-screen displays that result from different combinations of arguments and flags.

PRIVILEGE REQUIRED

None.

tokens — display all tokens.

tokens

DESCRIPTION

Instructs the Cache Manager on the local machine to display all tokens (tickets) it has for the issuer. AFS server processes require that their clients present a token as evidence that they have authenticated in the server's local cell.

Note: The **tokens.krb** version of this command is intended for use by sites that employ standard Kerberos authentication for their clients. The **tokens.krb** command provides all of the functionality of **tokens**. In addition, it provides information on the Kerberos tickets stored in the file specified by the KRBTKFILE environment variable (*/tmp/tktX* where *X* is the number of the user's PAG).

OUTPUT

The output lists one token for each cell in which the user is authenticated. The output indicates the

- User's AFS UID, if it is available for display.
- Server for which the token is valid (normally, "*afs*"). This includes a cell specification.
- Day and time the token expires.

Note: If you are using **tokens.krb**, the output will also contain information on the Kerberos ticket-granting ticket including: the ticket owner, the Kerberos ticket-granting service that issued the ticket (e.g., *krbtgt.TRANSARC.COM*), and the expiration date of the ticket.

An `--End of list--` message appears at the end of the output. If the user is not authenticated in any cell, this message is all that appears.

EXAMPLES

The following shows the output when the issuer is not authenticated in any cell.

```
% tokens
Tokens held by the Cache Manager:

    [  1]   --End of list--
```

The following shows the output when the issuer is authenticated in Transarc Corporation cell, where he or she has AFS UID 1000.

```
% tokens
Tokens held by the Cache Manager:

    [  1]User's (AFS ID 1000) tokens for afs@transarc.com
                                        [Expires Jan 2 10:00]
    [  2]    --End of list--
```

The following shows the output when the issuer is authenticated in Transarc Corporation cell, Andrew cell at Carnegie Mellon University and Athena cell at MIT. The user has different AFS UIDs in the three cells. Tokens for last cell are expired:

```
% tokens
Tokens held by the Cache Manager:

    [  1]User's (AFS ID 1000) tokens for afs@transarc.com
                      [Expires Jan 3 10:00]
    [  2]User's (AFS ID 4286) tokens for afs@andrew.cmu.edu
                      [Expires Jan 3 1:34]
    [  3]User's (AFS ID 22) tokens for afs@athena.mit.edu
                      [>>Expired<<]
    [  4]    --End of list--
```

The following shows the output when the issuer uses **tokens.krb** after authenticating in the Transarc Corporation cell using **klog.krb**.

```
% tokens.krb
Tokens held by the Cache Manager:

User's (AFS ID 1000) tokens for afs@transarc.com [Expires Jan 31 00:09]
User smiths tokens for krbtgt.TRANSARC.COM@transarc.com
[Expires Jan 31 00:09]
   --End of list--
```

PRIVILEGE REQUIRED

None.

MORE INFORMATION

klog

unlog

translate_et — translates numbered error codes into textual messages.

translate_et <*error number*>+

ACCEPTABLE ABBREVIATIONS/ALIASES

This command must be typed in full.

DESCRIPTION

Translates each specified *error number* into a textual message. This command may be useful if you need to decode numbered error messages.

ARGUMENTS

error number specifies each error number that you want to translate into a textual message.

EXAMPLES

The following translates the error numbers *1* and *4* into their corresponding messages:

```
% translate_et 1 4
1 ().1 = Not owner
4 ().4 = Interrupted system call
```

PRIVILEGE REQUIRED

None.

unlog — discard all tokens.

unlog [-cell <*cell name*>$^{+}$**] [-help]**

ACCEPTABLE ABBREVIATIONS/ALIASES

There is no abbreviation or alias for the **unlog** command; however, you can abbreviate the switch and flag normally.

DESCRIPTION

Instructs the Cache Manager on the local machine to discard the specified token(s) currently held for the issuer. If no cell names are provided, the Cache Manager discards the token for the local cell and all tokens for foreign cells; if cell names are provided, using the **-cell** argument, only the tokens for those cells are discarded.

Note: Since tokens are granted on a per-machine basis, destroying your tokens on one machine has no effect on tokens on another machine.

WARNINGS

Specifying one or more *cell names* may cause brief "authentication outages," during which the issuer has no valid tokens in any cell. This is because the command actually discards all tokens and then restores those that the issuer did not specify with *cell name* (and so presumably wishes to retain). The authentication outage may interrupt the operation of jobs that require authentication.

ARGUMENTS

-cell	specifies each cell for which the Cache Manager should discard the token. If omitted, all tokens are discarded. Abbreviated cell names are acceptable, but which abbreviations are legal depends on the naming service available in the cell at the time the command is issued.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

The following discards all tokens.

% unlog

The following discards only the tokens for the *transarc.com* and *athena.mit.edu* cells.

% unlog transarc.com athena

PRIVILEGE REQUIRED

None.

MORE INFORMATION

klog

tokens

up — recursively copy the contents of a source directory to a destination directory.

up [-v] [-1] [-f] [-r] [-x] <*source directory*> <*destination directory*>

ACCEPTABLE ABBREVIATIONS/ALIASES

This command does not use the syntax conventions of the AFS command suites.

DESCRIPTION

Recursively copies the contents (both files and subdirectories) of the *source directory* into the *destination directory*. By default, the following changes are made to the *destination directory* and the files and subdirectories within the *destination directory* (as appropriate):

- The ACL associated with the *source directory* is copied over the ACL of the *destination directory* and the subdirectories of the *destination directory*.
- If the command issuer runs the command as root and as a member of the group *system:administrators*, the file and directory ownerships of the *destination directory* are set to those of the *source directory*. If the command issuer runs the command as a regular user, the file and directory ownerships of the *destination directory* are set to the issuer's user name.
- Files and directories that exist in both the *source directory* and the *destination directory* are overwritten in the *destination directory* by their counterparts in the *source directory*, though overwriting fails if a directory, subdirectory, or file has the UNIX write mode bit turned off.
- The date and time of each file is set to the date and time associated with the source file. The command always sets the date and time of a newly created directory to the current date and time, while the date and time of existing directories remain unchanged.

Note: If the **up** command is interrupted by a network or server problem, you can reissue the command and it will takeover from where it left off. This saves time and reduces network traffic in comparison to the UNIX commands that provide similar functionality.

ARGUMENTS

-v	prints a detailed diagnostic message to standard output as the command executes.
-1	directs the command to copy only the files in the top level *source directory* to the *destination directory*, though the ACL of the *source directory* is copied over the ACL of the *destination directory*.
-f	directs the command to overwrite existing directories, subdirectories, and files whether or not those entities are protected against writing with the UNIX write user mode bit.
-r	directs the command to create a "backup" of all files that it overwrites in the *destination directory* and in the *destination directory*'s sub-directories. A *.old* extension is attached to the "backup" files.
-x	directs the command to set the date and time of each file to the current date and time.

source directory specifies the directory from which the recursive copy will be made.

destination directory
specifies the directory to which the recursive copy will be made.

EXAMPLES

The following copies the contents of the directory *dir1* to directory *dir2*:

% up dir1 dir2

PRIVILEGE REQUIRED

The issuer must have administer writes in the *source directory* and the *destination directory*.

upclient — initialize client portion of Update Server.

/usr/afs/bin/upclient <*hostname*> **[-level** <*clear*|*crypt*>**] [-clear] [-crypt]**
[-t <*retry time*>**]** <*dir*>+

ACCEPTABLE ABBREVIATIONS/ALIASES

This command does not use the syntax conventions of the AFS command suites, so the command and its switches and flags must be typed in full.

DESCRIPTION

Initializes the client portion of the Update Server. The *upclient* process periodically checks to make sure that the files in each specified directory (*dir*) match the files in the corresponding directory on *hostname*. If a file does not match, *upclient* calls upon the *upserver* process on *hostname* to transfer that file so that *upclient* can overwrite the local file with the transferred file.

The **-clear** or **-crypt** flag or **-level** argument is used to specify whether or not *upclient* requests file transfers to be encrypted or unencrypted. In the United States edition of AFS, an *upclient* process can be set up to transfer data in either an encrypted or unencrypted form; however, in the international edition, an *upclient* process can be set up to transfer data in unencrypted form only. (The **-crypt** flag is not available in the international edition of this command.) By default, the *upclient* in the United States edition of AFS requests encryption.

The *upclient* and *upserver* always mutually authenticate, whether or not the data they pass is encrypted. The *upclient* process uses the local *KeyFile* when generating a key for mutually authenticating with the *upserver* process.

Note: This command is not normally issued at the command shell prompt; rather, it is placed into a database server machine's */usr/afs/local/BosConfig* with the **bos create** command. If it is ever issued at the command shell prompt, the issuer must be working on a file server machine.

WARNING

Cells using the international edition of AFS should not use the Update Server to distribute the contents of */usr/afs/etc*. The contents of this directory should be closely protected and the international edition of AFS does not provide any facility for encrypting files before transfer across the network.

ARGUMENTS

hostname names either the cell's system control machine (if *dir* is */usr/afs/etc*) or the binary distribution machine for the local machine's CPU/operating system type (if *dir* is */usr/afs/bin*).

-level specifies whether or not *upclient* should request encrypted (*crypt*) or unencrypted (*clear*) data transfer from *upserver*. You should use this argument or the **-clear** or **-crypt** flag when specifying the transfer method; if the transfer method is not specified, it will default to *crypt* for the US version of AFS and *clear* for the international version.

-clear indicates that *upclient* requests *upserver* to transfer information in unencrypted form. Use this flag to change the default for the United States edition of AFS. You can use this flag or the **-crypt** flag or change the default using the **-level** argument.

-crypt indicates that *upclient* requests *upserver* to transfer information in encrypted form. With the United States edition of AFS, use this flag to set the default explicitly. You can use this flag or the **-crypt** flag or change the default using the **-level** argument. This flag is not available with the international edition of AFS.

-t specifies how often the *upclient* process should check *dir*, in number of seconds. If omitted, this value defaults to *300* (5 minutes). This number effectively determines how long it will take for changes made on *hostname* to propagate to this machine.

dir names each directory to check for modified files. The usual choices are:

- */usr/afs/bin*, in which case the recommended name for the process (assigned with the **-instance** argument on **bos create**) is *upclientbin*. The *hostname* should be the binary distribution machine for the local machine's CPU/operating system type. It is recommended that the **-clear** flag be used for */usr/afs/bin*, since binaries are not particularly sensitive and transferring them in encrypted form can take a long time.
- */usr/afs/etc*, in which case the recommended name for the process (assigned with the **-instance** argument on **bos create**) is *upclientetc*. The *hostname* should be the system control machine. It is recommended that the **-crypt** flag be used for */usr/afs/etc*, since this directory contains the *KeyFile* which is vital to system security

  Note: Since the international version of *upclient* cannot request encrypted data transfer, you should not transfer the contents of the */usr/afs/etc* directory using *upclient* and *upserver*. See the WARNING above.

EXAMPLES

The following **bos create** command creates an *upclientbin* process on *fs4.transarc.com* that refers to *fs1.transarc.com* as the source for */usr/afs/bin* (thus *fs1.transarc.com* is the binary distribution machine for machines of *fs4.transarc.com*'s type). The files in */usr/afs/bin* every 120 seconds. The command requests transfer in unencrypted form.

```
% bos create fs4.transarc.com upclientbin simple  "/usr/afs/bin/upclient
    fs1.transarc.com -clear -t 120 /usr/afs/bin"
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

Issuer must be logged into the machine's UNIX file system as "root" to run this command from the command line.

MORE INFORMATION

bos create

upserver

upserver — initialize server portion of Update Server.

/usr/afs/bin/upserver [<*directory*>$^{+}$**] [-crypt** <*directory*>$^{+}$**] [-clear** <*directory*>$^{+}$**]**

ACCEPTABLE ABBREVIATIONS/ALIASES

This command does not use the syntax conventions of the AFS command suites, so the command and its switches and flags must be typed in full.

DESCRIPTION

Initializes the server portion of the Update Server (*upserver* process), specifying which of its local disk directories the process is willing to distribute in response to requests from the client portion of the Update Server (*upclient* process) running on other machines. If no directories are specified, the server portion can distribute the contents of any directory on its local disk. While multiple instances of the *upclient* process can run on a single file server, only one instance of the *upserver* process can run on a single file server.

The *upserver* process will distribute a directory's contents in encrypted or unencrypted form. By default, the encryption level is **-clear**, meaning that *upserver* transfers data in unencrypted form unless an *upclient* requests encryption. When **-crypt** is specified, the *upserver* will refuse an *upclient*'s request for unencrypted transfer.

With the United States edition of AFS, using **-crypt** guarantees that *upserver* transfers a directory's contents only in encrypted form. With the international edition, using **-crypt** prevents *upserver* from ever transferring data, because *upclient* has no way to comply with the requirement that it request the contents in encrypted form (because the **-crypt** flag is not available on the international version of *upclient*).

The *upclient* and *upserver* processes always mutually authenticate, whether or not the data they pass is encrypted. The *upclient* process uses the local *KeyFile* when generating a key for mutual authentication with the *upserver* process.

Note: This command is not normally issued at the command shell prompt, but rather placed into a database server machine's */usr/afs/local/BosConfig* with the **bos create** command. If it is ever issued at the command shell prompt, the issuer must be working on a file server machine.

WARNING

Cells using the international edition of AFS should not use the Update Server to distribute the contents of */usr/afs/etc*. The contents of this directory are extremely crucial to system security, and the international edition of AFS has no way of encrypting them before transfer across the network. One way to prevent this distribution is to mark */usr/afs/etc* with **-crypt**.

ARGUMENTS

directory	names each directory to be distributed in unencrypted form (because they appear before the first **-crypt** or **-clear** argument). If omitted, all directories on the machine's local disk are eligible for distribution.
-crypt	precedes a list of one or more directories which *upserver* will distribute only in encrypted form.
-clear	precedes a list of one or more directories which *upserver* will distribute in unencrypted form (but if *upclient* requests them in encrypted form, then *upserver* encrypts them). This argument is necessary only if the issuer has previously used **-crypt** and wants to switch back to unencrypted form.

EXAMPLES

In the example below, the **bos create** command is used to initiate an *upserver* process on host *fs1.transarc.com*. The last parameter (enclosed in quotes) instructs *upserver* to distribute */usr/afs/bin* in unencrypted form and */usr/afs/etc* in encrypted form.

```
% bos create  fs1.transarc.com  upserver  simple
                "/usr/afs/bin/upserver /usr/afs/bin -crypt /usr/afs/etc"
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

Issuer must be logged into the machine's UNIX file system as "root" to run this command from the command line.

MORE INFORMATION

bos create

upclient

vldb_convert — convert Volume Location Database from AFS 3.1 to AFS 3.2.

vldb_convert [initcmd] [-to <*goal version*>**] [-from** <*current version*>**] [-path** <*pathname*>**] [-showversion] [-help]**

ACCEPTABLE ABBREVIATIONS

The name of this command must be typed in full. However, its switches can be omitted (if appropriate) or abbreviated as follows:

vldb_convert [-t <*goal version*>**] [-f** <*current version*>**] [-p** <*pathname*>**] [-s] [-h]**

DESCRIPTION

Converts a Volume Location Database (VLDB) from AFS 3.1 format to AFS 3.2 format or from an AFS 3.2 format back to an AFS 3.1 format. It can also be used to display the version number of the VLDB in use.

Note: Due to a limitation in the VLDB in AFS 3.1 and earlier versions of AFS, a cell could have no more than 31 file server machines. With AFS 3.2 and later versions of the VLDB, a cell can have as many as 255 file server machines.

When converting from AFS 3.1 to AFS 3.2 VLDB, **vldb_convert** reads the */usr/afs/db/vldb.DB0* database file for the old VLDB and writes a new, converted version of the database into a temporary file in the */usr/afs/db* directory. If successful, it then removes the old *vldb.DB0* file from the directory and replaces it with the new version from the temporary file. After the conversion, the database functions as it did before, with the exception that it can accommodate the entries for the additional file server machines. The binary file for the **vldb_convert** command resides in */usr/afsws/etc*.

The command requires the file system in which the VLDB resides to have free space at least equivalent to the size of the old VLDB plus 9 kilobytes. Because the command removes the old VLDB and replaces it with the new version of the database, the additional space is required only while the command executes. (The new version of the database requires a small amount of additional space, but the difference is inconsequential.)

Begin the conversion of the VLDB by shutting down the Volume Location (VL) Servers on all three database server machines. (The instructions assume the cell has three database server machines; adjust the steps accordingly if it has fewer.) This causes a service outage in the cell, but the outage for the entire conversion procedure should last no more than 15 to 20 minutes.

Replace the AFS 3.1 version of the */usr/afs/bin/vlserver* file on all three database server machines with the AFS 3.2 version of the file. The new VL Server understands the format of the new version of the VLDB. The old version of the VL Server cannot be run with the new version of the VLDB, and vice versa. Make sure you replace the *vlserver* binary files on each binary distribution machine; otherwise, the Update Servers overwrite the version stored on the database server machines by automatically propagating the old version of the binary file to the machines.

Determine which database server machine has the lowest IP address; Ubik will use this machine as the synchronization site when the VL Servers are restarted. Remove the VLDB from the other two database server machines by deleting the */usr/afs/db/vldb.DB0* database and */usr/afs/db/vldb.DBSYS1* log files from the other machines. Remove only the */usr/afs/db/vldb.DBSYS1* log file from the database server machine with the lowest IP address. Do not remove the */usr/afs/db/vldb.DB0* database file from the machine with the lowest IP address.

Copy the */usr/afs/db/vldb.DB0* file on the database server machine with the lowest IP address to a different directory. Then execute the **vldb_convert** command on that machine. The command takes no more than a few minutes to complete. However, it displays no messages while it executes.

When the command is finished, restart the VL Server on the machine with the lowest IP address. Then restart the VL Servers on the remaining two database server machines. Ubik elects the database server machine with the lowest IP address as the synchronization site and distributes the copy of the new VLDB from that machine to the other two database server machines. The VLDB is then available.

The **vldb_convert** command can be also used to convert the VLDB from AFS 3.2 format back to AFS 3.1 format. The command's **-to** and **-from** arguments are used to specify the direction of the conversion. The **-to** argument specifies the version to which the database is to be converted; the **-from** argument indicates the version from which the database is to be converted.

Both arguments accept a *1*, to indicate 3.1, or a *2*, to indicate 3.2. When converting from 3.1 to 3.2, specify **-to** as *2* and **-from** as *1*. Should the need to revert to 3.1 arise, specify **-to** as *1* and **-from** as *2*. When converting from 3.1 to 3.2, you can omit both options; the command automatically converts the database to the next higher version.

Finally, the command can be used to display the version of the current VLDB. The **-showversion** flag directs the command to display the version number (3.1 or 3.2) of the VLDB at the time the command is issued. Do not specify the **-to** or **-from** argument if the **-showversion** flag is used.

NOTE

The **vldb_convert** command is also useful as a means of removing from the VLDB entries for server machines that house no volumes. When a file server machine name or its IP address is changed, the VLDB still contains an entry for the previous name or address. This becomes a problem only when the administrators in a large cell must change the names or IP addresses of a significant number of machines, in which case it is conceivable (though highly unlikely) that all 255 possible file server machine entries will be taken.

If this problem occurs in a cell, the administrators can execute **vldb_convert** with a value of *2* for both its **-to** and **-from** arguments. The command removes the entries for all unreferenced servers from the VLDB. The steps described previously for executing the command (shutting down the servers, removing the copies of the VLDB) must be executed as they are for initially upgrading the database.

WARNING

Copy the */usr/afs/db/vldb.DB0* file to a different directory after the VL Servers are shut down but before the **vldb_convert** command is run. The command should perform the conversion without problems. Also, it does not remove the old database file until it has successfully completed the conversion. However, if anything happens to disrupt the conversion or if the operation fails for any reason, copying the database beforehand allows the VLDB to be restored from the copy.

Leave no VL Servers running while the conversion is in progress. If a number of VL Servers sufficient for Ubik to attain a quorum are available while the conversion is in progress, changes made while the conversion is underway are lost when the new version of the VLDB is distributed.

Ensure that all VL Servers are referencing the same version of the VLDB at all times. The 3.1 version of the VL Server can reference only the 3.1 version of the VLDB, and the 3.2 version of the VL Server can use only the 3.2 version of the VLDB. The 3.1 version of the VL Server is not compatible with the 3.2 version of the VLDB, and vice versa. For this reason, the **vldb_convert** command must be used to convert the existing VLDB to the new format before the new version of the VL Server distributed from the cell's binary distribution machines via the Update Server can be used.

Specify only valid, correct values for the **-to** and **-from** arguments. Specifying invalid or incorrect values for these arguments can result in damage to the VLDB. This is another reason to copy the *vldb.DB0* database file to a different directory before using the **vldb_convert** command.

Finally, the entire procedure takes no more than a few minutes to complete. However, because volumes are inaccessible while the procedure is underway, it is best to perform the operation over a weekend or overnight to disrupt the fewest users.

ARGUMENTS

initcmd is an optional string that accommodates the command's use of the AFS command parser. It can be omitted and ignored.

-to indicates the version of the VLDB to which the current version of the database is to be converted. Valid values are *1*, indicating that the existing version of the database is to be converted to AFS 3.1 format, or *2*, indicating that the existing version of the database is to be converted to AFS 3.2 format. See the DESCRIPTION section for information about the possible combinations of the **-to** and **-from** arguments and the effects of omitting the arguments.

-from indicates the version of the current VLDB. Valid values are *1*, indicating that the existing version of the database is in AFS 3.1 format, or *2*, indicating that the existing version of the database is in AFS 3.2 format. See the DESCRIPTION section for information about the possible combinations of the **-to** and **-from** arguments and the effects of omitting the arguments.

-path provides the complete pathname of the *vldb.DB0* database file if the file is kept in an alternate directory (or has a different name). Always include the name of the file, even if it is *vldb.DB0*. By default, the command converts the *vldb.DB0* database file in the */usr/afs/db* directory. Use this argument only if the database is stored in a different directory (or has a different name).

-showversion displays the version (3.1 or 3.2) of the current VLDB. Include the **-path** argument if the VLDB is stored in a directory other than */usr/afs/db*. If you use this flag, the **-to** and **-from** arguments are ignored.

-help prints the online help for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

The following, issued on the database server machine with the lowest IP address, executes the **vldb_convert** command to convert the VLDB from AFS 3.1 format to AFS 3.2 format:

```
% vldb_convert
```

PRIVILEGE REQUIRED

The issuer must be able to read, delete, and write to the */usr/afs/db/vldb.DB0* and */usr/afs/db/vldb.DBSYS1* files on all database server machines. The issuer must also be able to insert, write to, and delete files in the */usr/afs/db* directories on the machines.

In addition, to install the new version of the VL Server, the issuer must be able to delete the */usr/afs/bin/vlserver* file on all database server machines (or on all binary distribution machines). The issuer must also be able to insert files in the */usr/afs/bin* directories on the machines.

An issuer who is logged into the UNIX file systems of the database server machines as "root" has the necessary rights to perform the entire conversion procedure.

vlserver — initialize Volume Location Server.

/usr/afs/bin/vlserver [-p <*lwp processes*>**]**

ACCEPTABLE ABBREVIATIONS

This command does not use the syntax conventions of the AFS command suites, so the command and its switch must be typed in full.

DESCRIPTION

Initializes the Volume Location (VL) Server. The **-p** argument is used to set the number of lightweight processes which defaults to 4.

The VL Server maintains a complete list of volume locations in the ***Volume Location Database (VLDB)***. When the Cache Manager begins to fill a file request from an application program, it first contacts the VL Server in order to learn which file server machine currently houses the volume containing the file. The Cache Manager then requests the file from the File Server process running on that file server machine.

The VLDB and VL Server make it possible for AFS to take advantage of the increased system availability gained by using multiple file server machines, because the Cache Manager knows where to find a particular file. Indeed, in a certain sense the VL Server is the keystone of the entire file system—when the information in the VLDB is inaccessible, the Cache Manager cannot retrieve files, even if the File Server processes are working properly.

Note: This command is not normally issued at the command shell prompt, but rather placed into a database server machine's */usr/afs/local/BosConfig* with the **bos create** command. If it is ever issued at the command shell prompt, the issuer must be working on a file server machine.

ARGUMENTS

-p	sets the number of server lightweight processes (LWPs) to run. Provide a positive integer. The default is 4.

EXAMPLES

The following **bos create** command creates a *vlserver* process on *fs2.transarc.com* that uses six lightweight processes. Type the command on a single line:

```
% bos create  fs2.transarc.com  vlserver  simple  "/usr/afs/bin/vlserver  -p  6"
```

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

Issuer must be logged into the machine’s UNIX file system as "root" to run this command from the command line.

MORE INFORMATION

bos create

bos getlog

volserver — initialize Volume Server component of *fs* process.

/usr/afs/bin/volserver [-verbose <*debuglevel*>**] [-log] [-help]**

ACCEPTABLE ABBREVIATIONS

This command does not use the syntax conventions of the AFS command suites, so the command and its switches and flags must be typed in full.

DESCRIPTION

Initializes the Volume Server component of the *fs* process. The Volume Server creates, removes, moves, dumps, and restores volumes in response to **vos** commands.

The Volume Server provides the interface through which you create, delete, move, and replicate volumes, as well as prepare them for archiving to tape or other media (backing up). Creating and deleting volumes are necessary when adding and removing users from the system; volume moves are done for load balancing; and replication allows volume placement on multiple file server machines.

Note: This command is not normally issued at the command shell prompt, but rather placed into a file server machine's */usr/afs/local/BosConfig* with the **bos create** command. If it is ever issued at the command shell prompt, the issuer must be working on a file server machine.

When executed, the Volume Server creates an activity log file named */usr/afs/logs/VolserLog* if the file does not already exist. This log file can be examined using the **bos getlog** command. It will contain information on when the Volume Server is started, when the salvage of a volume is complete, and when volume creation, movement, and deletion took place. When a new version of the *VolserLog* is created, the current version is renamed *VolserLog.old*.

ARGUMENTS

-verbose	causes the Volume Server to write status information to */usr/afs/logs/VolserLog* as it goes about its tasks. This flag is mainly used for debugging purposes.
-log	causes the Volume Server to record the names of all users who successfully initiate a **vos** commands in the */usr/afs/logs/VolserLog* file.
-help	prints the online help for this command. Do not provide any other arguments or flags with this one.

EXAMPLES

The following **bos create** command creates a *volserver* process on *fs2.transarc.com*:

% bos create fs2.transarc.com volserver simple /usr/afs/bin/volserver

PRIVILEGE REQUIRED

Issuer must be listed in */usr/afs/etc/UserList* to place this command in */usr/afs/local/BosConfig*, because that is the privilege required to issue **bos create**.

Issuer must be logged into the machine's UNIX file system as "root" to run this command from the command line.

MORE INFORMATION

bos create

bos getlog

10. Modified UNIX Commands

This chapter describes standard UNIX commands that have been modified to work more appropriately in the AFS environment. Most of the modifications are to make the commands able to handle AFS authentication information (tokens) properly. The commands are

ftpd
inetd
login
rcp
rlogind
rsh

The privilege required to issue the commands described in this chapter is the same as for the standard UNIX versions. The process for installing the command binaries and the appropriate settings for UNIX owner and mode bits are similarly unchanged.

Unless otherwise noted, the AFS version of all commands in this chapter is based on the BSD 4.3 Tahoe distribution.

ftpd (AFS version) — initialize Internet File Transfer Protocol server.

/usr/etc/ftpd [-d] [-l] [-t <*timeout*>] [-v] [-T <*MaxTimeOut*>] [-u] [-s]

DESCRIPTION

Functions like the standard UNIX **ftpd**, except that it also authenticates the issuer of the **ftp** command (who is presumably working on a remote machine) with the Authentication Server in the local cell (the home cell of the machine where **ftpd** is running). The authentication is based on the user name and password provided at the **ftp** prompts on the remote machine. The Cache Manager on the machine running **ftpd** stores the newly created token, identifying it by PAG rather than by the user's UNIX UID.

The issuer of **ftp** may be working in a foreign cell, as long as the user name and password provided are valid in the cell where **ftpd** is running. If the user name under which **ftp** is issued does not exist in the Authentication Database for the cell where **ftpd** is running, or the issuer provides the wrong password, then **ftpd** logs the user into the UNIX file system of the machine where **ftpd** is running. The success of this local login depends on the user name appearing in the local password file and on the user providing the correct local password. In the case of a local login, AFS server processes consider the issuer of **ftp** to be *anonymous*.

In the configuration recommended by Transarc Corporation, the AFS version of **ftpd** is substituted for the standard version (only one of the versions can run at a time). The administrator then has two choices:

- Name the binary for the AFS version something like **ftpd.afs**, leaving the standard version as **ftpd**. Change *inetd.conf* to refer to **ftpd.afs**; the standard version is not referenced.
- Name the binary for the AFS version **ftpd** and rename the binary for the standard version to something like **ftpd.old**. No change to *inetd.conf* is necessary, but it is not as obvious that the standard version of **ftpd** is no longer in use.

WARNING

If you intend to use a shell other than */bin/csh* to use the **ftp** program to access data from a machine, this command requires that the */etc/shells* file on the local disk of the machine from which data is to be accessed list the shell that you are running. Otherwise, the **ftp** program will not function properly. For example, users on Hewlett-Packard system types commonly use */bin/ksh*; the */etc/shells* file on a machine that such users contact via **ftp** must include a line for */bin/ksh*.

ARGUMENTS

-d	directs debugging information to the system log daemon.
-l	directs each FTP session to be logged to the system log daemon.
-t	specifies a timeout period. By default, the FTP server will timeout an inactive session after 15 minutes.

-v	same as **-d**.
-T	specifies a timeout period in seconds. By default, the FTP server will timeout after 2 hours (7200 seconds). This argument is only valid on machines using the AIX 3.2 operating system.
-s	turns on socket level debugging. This flag is only valid on machines using the AIX 3.2 operating system.
-u	specifies the default unix mode bit file mask that will be used. This is flag only valid on machines using the AIX 3.2 operating system.

PRIVILEGE REQUIRED

See the UNIX manual page for **ftpd**.

MORE INFORMATION

UNIX manual page for **ftp**

UNIX manual page for **ftpd**

inetd (AFS version) — initialize Internet service daemon.

/etc/inetd [-d] [<*configfile*>]

DESCRIPTION

Functions like the standard UNIX **inetd** in invoking an Internet service daemon (itself called "inetd") that handles remotely-issued commands. The AFS **inetd** enables users of the remote services it supports to access those services as authenticated AFS users, provided that the supported services are also AFS versions capable of passing AFS tokens (authentication information). Examples of supported services are **rcp** and **rsh**.

AFS **inetd** can service the standard UNIX versions of the remote services, but it is instead recommended that the standard UNIX version of **inetd** be run in parallel with the AFS version. Name the AFS version something like **inetd.afs** and use it to service requests from AFS-modified programs; use standard **inetd** to service requests from standard UNIX programs. This separation requires using two different *inetd.conf* files, as described in the REQUIREMENTS/RESTRICTIONS section.

REQUIREMENTS/RESTRICTIONS

Several configuration changes are necessary for token passing to work correctly with the AFS version of **inetd**. There may be other UNIX-based requirements/restrictions not mentioned here; consult the UNIX manual page. (One important restriction is that there can be no blank lines in the configuration file other than at the end.)

The requirements/restrictions include the following. They assume that **inetd.afs** is running in parallel with standard **inetd**.

- For token passing to work, the request must come from the AFS version of the remote service (such as AFS **rcp** or AFS **rsh**). If the remote service is the standard UNIX version, it will not pass tokens. The issuer of the remote command is authenticated only in the local UNIX file system, not with AFS, so the AFS server processes in the local cell consider the issuer to be *anonymous*.
- The machine's reboot configuration file (*/etc/rc* or equivalent) should be altered so that it starts both standard **inetd** and **inetd.afs**.
- An AFS-specific *inetd.conf* file, perhaps called *inetd.conf.afs*, should exist alongside the standard one. When initializing **inetd.afs**, specify this configuration file rather than the standard one.

Each line in the *inetd.conf.afs* file must include an additional field, fifth from the left, to specify the identity under which the program is to run. The normal choice is "root." The following sample shows the only lines that should appear in this file:

```
ta-rauth stream tcp nowait root internal             ta-rauth
shell    stream tcp nowait root /usr/etc/rshd        rshd
login    stream tcp nowait root /usr/etc/rlogind.afs rlogind.afs
```

Substitute appropriate values for the binary locations and names on the `shell` and `login` lines. Include the `login` line only if the AFS version of **login** is also in use in the cell; otherwise, refer to `login` in the standard *inetd.conf* instead. In addition, if the *inetd.cond.afs* file is used on a machine with a Sun system type, change `rshd` to `in.rshd` and change `rlogind.afs` to `in.rlogind.afs` on the `shell` and `login` lines, respectively.

- The standard *inetd.conf* (referred to by the standard **inetd**) should be altered: comment out the `shell` line and, if the AFS version of **login** is in use in the cell, the `login` line. References to these programs appear in *inetd.conf.afs* instead. Retain the `login` line if AFS **login** is not being used. Alter the `ftp` line to refer to the AFS version of **ftpd**, if it was substituted for the standard version. Do not insert the extra fifth column into standard *inetd.conf* if it does not already appear there. See the EXAMPLES section below for an illustration.
- The following two lines must appear in the */etc/services* file on the machine running **inetd** (as well as on the machine running modified **rcp** or **rsh**). On NeXT machines, this information must appear in the NetInfo database rather than in */etc/services*.

```
auth 113/tcp authentication
ta-rauth 601/tcp rauth
```

ARGUMENTS

See the UNIX manual page for **inetd**.

EXAMPLES

The following are sample *inetd.conf.afs* and *inetd.conf* files, appropriate for use when **inetd.afs** is running in parallel with standard **inetd** and AFS **login** is being used in the cell. Changes to standard *inetd.conf* include referencing the AFS version of the **ftpd** binary and commenting out `shell` and `login`. The example *inetd.conf* does not include the extra fifth column. **Do not use these examples without modifying them appropriately for the local machine type or cell.**

```
# AFS version of Internet server configuration database
#(EXAMPLE ONLY)
#
ta-rauth stream tcp nowait root internal             ta-rauth
shell    stream tcp nowait root /usr/etc/rshd        rshd
login    stream tcp nowait root /usr/etc/rlogind.afs rlogind.afs
```

```
# Standard version of Internet server configuration database
#(EXAMPLE ONLY)
#
ftp     stream  tcp     nowait  /etc/ftpd.afs   ftpd.afs
telnet  stream  tcp     nowait  /etc/telnetd    telnetd
#shell  stream  tcp     nowait  /etc/rshd       rshd
#login  stream  tcp     nowait  /etc/rlogind    rlogind
finger  stream  tcp     nowait  /usr/etc/fingd  fingd
uucp    stream  tcp     nowait  /etc/uucpd      uucpd
exec    stream  tcp     nowait  /etc/rexecd     rexecd
comsat  dgram   udp     wait    /etc/comsat     comsat
talk    dgram   udp     wait    /etc/talkd      talkd
ntalk   dgram   udp     wait    /usr/etc/ntalkd talkd
time    dgram   udp     wait    /etc/miscd      timed
```

PRIVILEGE REQUIRED

See the UNIX manual page for **inetd**.

MORE INFORMATION

rcp (AFS version)

rsh (AFS version)

UNIX manual page for **inetd**

login (AFS version) — login to AFS and UNIX file systems.

login [<*user name*>]

DESCRIPTION

Authenticates the issuer with the AFS Authentication Server in the local cell and logs him or her into the local machine's UNIX file system. More precisely, AFS **login**:

- Creates a new "process authentication group" (PAG) associated with the issuer and the command shell where the command is issued. A PAG is a number guaranteed to identify the issuer uniquely to the local Cache Manager. The Cache Manager uses the PAG, instead of the issuer's UNIX UID, to identify him or her in the credential structure that it creates to track each user.
- Authenticates the user with the AFS Authentication Server and obtains a "token"; the other AFS server processes in the cell accept the token as partial proof that the user is a legitimate AFS user. The Cache Manager stores the token in the credential structure along with the PAG, and presents it to AFS server processes as necessary.
- Logs the user into the local UNIX file system.
- Sets the global environment variable (PASSWORD_EXPIRES) to the number of days until the password is scheduled to expire. Zero means that the password expires today and will not be valid tomorrow. If a user's password never expires, PASSWORD_EXPIRES will not be set. This environment variable permits system administrators to customize user login scripts to
  - Display the password expiration date during user login (after authentication).
  - Provide the user the option to change his or her password whenever the expiration date is close at hand.

If a user receives the message `Unable to authenticate to AFS because ID is locked - see your system admin` it indicates that someone has attempted to authenticate with the user's account and has exceeded the authentication failure limit. Use the **kas unlock** command to unlock the user account.

The AFS **login** command can only be used on machines that have run **afsd**. It does not work on NFS client machines, even if they are of a supported system type and can see the **login** binary in AFS via an NFS/AFS translator machine.

Note: There are three versions of the **login** command; the binaries of the versions are *login*, *login.krb*, and *login.noafs*.

- The *login* version should be used by sites using regular AFS authentication. It places a copy of the ticket-granting ticket associated with the user's authenticated AFS login in the kernel of the machine.
- The **login.krb** version of this command is intended for use by sites that employ standard Kerberos authentication for their clients. It provides all of the functionality of the **login** command. In addition, it places a copy of the Kerberos tickets associated with the user's authenticated AFS login in the file specified by the KRBTKFILE environment variable. KRBTKFILE is defined by pagsh.krb to be */tmp/tktX* (where *X* is the number of the user's PAG).
- The *login.noafs* version is intended for use on the AIX 3.2 operating system.

The lifetime of the token resulting from this command is the smallest of the following:

- The "maximum ticket lifetime" recorded in the "*afs*" entry in the Authentication Database. The default is 100 hours. Administrators can inspect this value using **kas examine**, and change it using **kas setfields**.
- The "maximum ticket lifetime" recorded in the issuer's Authentication Database entry. The default is 25 hours for user entries created by the AFS 3.1 or later version of the Authentication Server, and 100 hours for user entries created by the AFS 3.0 version of the Authentication Server. Administrators and the user himself/herself can inspect this value using **kas examine**, and administrators can change it using **kas setfields**.
- The "maximum ticket lifetime" recorded in the "*krbtgt.**CELLNAME***" entry in the Authentication Database; this entry corresponds to the ticket-granting ticket used internally in generating the token. The default is 720 hours (30 days).

If none of these defaults have been changed, then the standard token lifetime is 25 hours for users whose Authentication Database entries were created by the AFS 3.1 or later version of the Authentication Server, and 100 hours for users whose Authentication Database entries were created by the AFS 3.0 version of the Authentication Server. The user can issue **klog** to request a token with a different lifetime.

Using a PAG instead of the UNIX UID to distinguish users has two advantages.

- It means that processes spawned by the user inherit the PAG and so share the token; thus they gain access to AFS as the authenticated user. This is important in many environments where, for example, printer and other daemons run under identities (such as "root") that the AFS server processes recognize only as *anonymous*. Unless PAGs are used, such daemons cannot access information in directories protected against *system:anyuser*.
- It closes a potential security loophole: UNIX allows anyone already logged in as "root" on a machine to assume any other identity by issuing **su**. If the credential structure were identified by a UNIX UID rather than a PAG, then assuming the same UID would mean being able to use the token, too. Use of a PAG as an identifier eliminates that possibility.

The process of authenticating with the AFS Authentication Server is as follows:

1. The **login** program checks the user's entry in the local */etc/passwd* file.

   If no entry exists, or if an asterisk (*) appears in the password field, the login attempt fails.

   If the entry exists, the attempt proceeds to step 2.

2. The **login** program invokes the command that creates a PAG.

3. The **login** program converts the password provided by the user into an encryption key and encrypts a packet of data with the key. It sends the packet to an AFS Authentication Server. The Authentication Server decrypts the packet and, depending on the success of the decryption, judges the password to be correct or incorrect. (Consult the *AFS System Administrator's Guide* for more information on the "mutual authentication" procedure used to verify the password in this step.)

   If the Authentication Server judges the password incorrect, the user does not receive an AFS token. The attempt proceeds to step 4.

   If the Authentication Server judges the password correct, it issues a token to the user as proof of AFS authentication. The **login** program also logs the user into the local UNIX file system. Step 4 is skipped.

4. If no AFS token was granted in step 3, the **login** program attempts to log the user into the local UNIX file system, by comparing the password provided to the local password file (*/etc/passwd*, for instance).

   If the provided password is incorrect, or the password field in the local password file contains anything other than a 13-character UNIX-encrypted password string, then the login attempt fails.

   If the password is correct, the user is logged into the local UNIX file system only. (The success of this attempt and the failure of the AFS authentication implies that the AFS and local passwords are different and that the issuer provided the latter.)

WARNINGS

Each PAG uses two of the memory slots that the kernel uses to record the UNIX groups associated with a user. If none of these slots are available, the PAG creation fails. The use of two group slots per PAG does not present a problem with most operating systems, which make at least 16 slots available.

The AFS version of **login** is based on the Berkeley 4.3 Distribution and does not include the modified features included in some proprietary versions of **login**.

The AFS version of **login** works only on machines that have run **afsd**.

ARGUMENTS

See the UNIX manual page for **login**. The AFS version does not impose any stronger restrictions on acceptable user names than does the UNIX file system.

OUTPUT

To distinguish the AFS version of **login** from other versions, one of the following banners appears on standard out (stdout) when the command executes successfully.

```
AFS 3.0 Login
```

or

```
AFS 3.3 (R) Login
```

Various error messages appear if the login attempt is not successful. The "`login:`" prompt normally returns following the error messages, giving the user another chance to type the password correctly.

PRIVILEGE REQUIRED

None.

MORE INFORMATION

afsd

kas unlock

klog

pagsh

rlogind (AFS version)

UNIX manual page for **login**

rcp (AFS version) — copy file on remote machine.

rcp [-p] <*file1*> <*file2*>
rcp [-r] [-p] <*file*>+ <*directory*>

DESCRIPTION

Functions like the standard UNIX **rcp**, except it allows the issuer to use the remote machine's Cache Manager to access AFS files as an authenticated AFS user. The command passes a copy of the AFS token which the issuer obtained on the local machine to the remote machine's Cache Manager. The remote Cache Manager can use the token to gain authenticated access to AFS.

Token passing is most effective if both the remote machine and local machine belong to the same cell, because **rcp** can pass only one token even if the user has several tokens—it passes the token that is marked `[1]` in the output from the **tokens** command. If the remote and local machine are not in the same cell, the possibilities are:

- The token passed is for the cell to which the remote machine belongs. The issuer accesses the remote cell's AFS file tree as an authenticated AFS user, but is considered *anonymous* in the local cell. This means that the issuer can only exercise the access rights granted to *system:anyuser* in the local AFS file tree. For instance, a file being copied into the local cell can only be copied into a UNIX directory or an AFS directory where *system:anyuser* has the INSERT right.
- The token passed is for the cell to which the local machine belongs. The issuer accesses the remote cell's AFS file tree as *anonymous*, and so can only exercise the access rights granted to *system:anyuser*.

In addition to running the AFS version of the **rcp** binary on the machine where the **rcp** command is issued, other configuration changes are necessary for token passing to work properly. See the REQUIREMENTS/RESTRICTIONS section for a list.

The AFS version of **rcp** is compatible with the standard **inetd**, but token passing works only if the AFS versions of both programs are being used. If only one of them is modified, the issuer will access AFS files through the remote machine as *anonymous*.

WARNINGS

AFS **rcp** does not allow "third party copies", in which neither source nor target file is on the current machine. Standard UNIX **rcp** claims to provide this functionality.

The protections required on the *.rhosts* file in order for token passing to work with this command (see the REQUIREMENTS/RESTRICTIONS section) are the opposite of those necessary for token creation to work with the AFS version of **rlogind**.

For security's sake, use the AFS version of **rcp** only in conjunction with PAGs, either by using the AFS version of **login** or always issuing **pagsh** before obtaining tokens.

REQUIREMENTS/RESTRICTIONS]

Several configuration requirements and restrictions are necessary for token passing to work correctly with the AFS version of **rcp**. Some of these are also necessary with the standard UNIX version, but are included here because the issuer accustomed to AFS protections may not consider them. There may be other UNIX-based requirements/restrictions not mentioned here; consult the UNIX manual page. (One important one is that no `stty` commands may appear in the issuer's shell initialization file, such as *.cshrc*.)

The requirements/restrictions for token passing include the following.

- The local machine must be running the AFS version of **rcp**, with the **rcp** binary file locally installed to grant setuid privilege to the owner, "root".
- The remote machine must be running the AFS version of **inetd**.
- The following two lines must appear in the */etc/services* file on the local machine (as well as on the remote machine running modified **inetd**). On NeXT machines, this information must appear in the NetInfo database rather than in */etc/services*.

```
auth 113/tcp authentication
ta-rauth 601/tcp rauth
```

- If **rcp** is to consult an *.rhosts* file on the remote machine, the file must have UNIX protections no more liberal than `-rw-r--r--`. If *.rhosts* resides in a user home directory in AFS, the home directory must also grant the LOOKUP and READ rights to *system:anyuser*.

ARGUMENTS

Consult the UNIX manual page for **rcp**.

PRIVILEGE REQUIRED

None.

MORE INFORMATION

inetd (AFS version)

UNIX manual page for **rcp**

rlogind (AFS version)

tokens

rlogind (AFS version) — initialize remote login server.

/etc/rlogind [-l] [-n] [-s]

DESCRIPTION

Functions like the standard UNIX **rlogind**, except that it is appropriate only for cells using the AFS version of the **login** program supplied by Transarc Corporation. The AFS modifications to **rlogind** are strictly internal and are necessary so that remote AFS authentication is possible with the **rlogin** command.

When a user issues **rlogin**, the AFS version of **rlogind** running on the remote machine invokes the AFS version of **login**. The user therefore obtains AFS tokens on the remote machine.

Note: When a user logs into a local machine as "root." issues **rlogind**, and then issues **klog**, the tokens created are associated with the PAG of the user's AFS identity rather than with the UID for "root."

In the configuration recommended by Transarc Corporation, the AFS version of **rlogind** is substituted for the standard version—but only if the AFS version of **login** is also being used in the cell. (Only one version of **rlogind** can run on a machine at a time.) The administrator then has two choices:

- Name the binary for the AFS version something like **rlogind.afs**, leaving the standard version as **rlogind**. Refer to **rlogind.afs** in *inetd.conf.afs* (the AFS version of *inetd.conf*), and comment out the reference to standard **rlogind** in standard *inetd.conf*.
- Name the binary for the AFS version **rlogind** and rename the binary for the standard version to something like **rlogind.old**. Refer to **rlogind** in *inetd.conf.afs* (the AFS version of *inetd.conf*), and comment out the reference to **rlogind** in standard *inetd.conf*.

WARNINGS

The AFS version of **rlogind** is not available for the AIX 2.2.1 operating system. (AIX 2.2.1 does not include the standard **rlogind** either.)

Do not install the AFS version of **rlogind** if the AFS version of **login** is not being used in the cell.

Remote AFS authentication is possible with **rlogin** only if

either no *.rhosts* file exists on the machine where **rlogind** is running

or *.rhosts* exists on the machine where **rlogind** is running, but has mode bits more liberal than `-rw-r--r--`. If *.rhosts* has mode bits as restrictive as `-rw-r--r--`, then **rlogind** logs the issuer of the remote **rlogin** command into the local UNIX file system without prompting for a password. The issuer does not obtain tokens (authenticate with AFS), because that requires providing a password. The user can, however, obtain tokens by issuing **pagsh** and **klog** after establishing the connection.

The protection required on *.rhosts* for token creation to work properly are exactly opposite those necessary for the AFS versions of **rcp** and **rsh** to handle tokens properly. The recommended solution is to configure *.rhosts* so that **rcp** and **rsh** work properly and to use **telnet** instead of **rlogin**.

No modifications to **rlogin** itself are necessary for AFS authentication to work.

ARGUMENTS

-l	directs the command to check the *.rhosts* file for the UNIX UID of the command issuer.
-n	directs the command to use the socket option to keep the connection alive.
-s	directs the command to turn on tracing for debugging purposes. This flag can only be used on machines using the AIX 3.2 operating system.

PRIVILEGE REQUIRED

See the UNIX manual page for **rlogind**.

MORE INFORMATION

login (AFS version)

rcp (AFS version)

UNIX manual page for **rlogind**

rsh (AFS version)

rsh (AFS version) — open shell on remote machine.

rsh host [-l <*username*>**] [-n]** <*command*>
host [-l <*username*>**] [-n]** <*command*>

DESCRIPTION

Functions like the standard UNIX **rsh**, except that it allows the issuer to execute commands on the remote machine as an authenticated AFS user. The command passes a copy of the AFS token that the issuer has obtained on the local machine to the remote machine's Cache Manager. The remote Cache Manager can use the token to have the issuer recognized as an authenticated AFS user.

Token passing is most effective if both the remote machine and local machine belong to the same cell, because **rsh** can pass only one token even if the user has several—it passes the token that is marked `[1]` in the output from the **tokens** command. If the remote and local machine are not in the same cell, the token should be for the cell to which the remote machine belongs, so that the remote cell's server processes will recognize the issuer as authenticated.

In addition to running the AFS version of the **rsh** binary on the machine where the **rsh** command is issued, other configuration changes are necessary for token passing to work properly. See the REQUIREMENTS/RESTRICTIONS section for a list.

The AFS version of **rsh** is compatible with the standard **inetd**, but token passing only works if the AFS versions of both programs are being used. If only one of them is modified, the issuer will access AFS files through the remote machine as *anonymous*.

WARNINGS

The protections required on the *.rhosts* file for token passing to work correctly with this command (see the REQUIREMENTS/RESTRICTIONS section) are the opposite of those necessary for token creation to work correctly with the AFS version of **rlogind**.

For security's sake, use the AFS version of **rsh** only in conjunction with PAGs, either by using the AFS version of **login** or always issuing **pagsh** before obtaining tokens.

REQUIREMENTS/RESTRICTIONS

Several configuration requirements and restrictions are necessary for token passing to work correctly with the AFS version of **rsh**. Some of these are also necessary with the standard UNIX version, but are included here because the issuer used to AFS protections may not be inclined to think of them. There may be other UNIX-based requirements/restrictions not mentioned here; consult the UNIX manual page. (One important one is that no `stty` commands may appear in the issuer's shell initialization file, such as *.cshrc*.)

The requirements/restrictions for token passing include the following.

- The local machine must be running the AFS version of **rsh**, with the binary file locally installed to grant setuid privilege to the owner, "root".
- The remote machine must be running the AFS version of **inetd**.
- The following two lines must appear in the */etc/services* file on the local machine (as well as on the remote machine running modified **inetd**). On NeXT machines, this information must appear in the NetInfo database rather than in */etc/services*.

```
auth 113/tcp authentication
ta-rauth 601/tcp rauth
```

- If **rsh** is to consult an *.rhosts* file on the remote machine, the file must have UNIX protections no more liberal than `-rw-r--r--`. If *.rhosts* resides in a user home directory in AFS, the home directory must also grant the LOOKUP and READ rights to *system:anyuser*.

ARGUMENTS

Consult the UNIX manual page for **rsh**.

MORE INFORMATION

inetd (AFS version)

UNIX manual page for **rsh**

rlogind (AFS version)

tokens

11. Command Summary

This chapter lists the syntax for all AFS commands. Commands are organized by command suite. The meaning of symbols used in this summary (brackets, etc.) is described in the *About This Manual* section, page x.

11.1. Summary of backup Commands

backup adddump -dump <*dump level name*>+ **[-expires** <*expiration date*>+**] [-help]**

backup addhost -tapehost <*tape machine name*> **[-portoffset** <*TC port offset*>**]**
[-help]

backup addvolentry -name <*volume set name*> **-server** <*machine name*>
-partition <*partition name*> **-volumes** <*volume name (regular expression)*>
[-help]

backup addvolset -name <*volume set name*> **[-help]**

backup apropos -topic <*help string*> **[-help]**

backup dbverify [-detail] [-help]

backup deldump -dump <*dump level name*> **[-help]**

backup deletedump -dumpid <*dump id*>+ **[-from** <*date time*>**]**
[-to <*date time*>**] [-help]**

backup delhost -tapehost <*tape machine name*> **[-portoffset** <*TC port offset*>**]**
[-help]

backup delvolentry -name <*volume set name*> **-entry** <*volume set index*> **[-help]**

backup delvolset -name <*volume set name*>+ **[-help]**

backup diskrestore -server <*machine to restore*> **-partition** <*partition to restore*>
[-portoffset <*TC port offset*>**] [-newserver** <*destination machine*>**]**
[-newpartition <*destination partition*>**] [-n] [-help]**

backup dump -volumeset <*volume set name*> **-dump** <*dump level name*>
[-portoffset <*TC port offset*>**] [-at** <*Date/time to start dump*>**]**
[-append] [-n] [-file <*load file*>**] [-help]**

backup dumpinfo [-ndumps <*no. of dumps*>**] [-id** <*dump id*>**] [-verbose] [-help]**

backup help [-topic <*help string*>+**] [-help]**

(backup) jobs [-help]

(backup) kill -id <*job ID or dump set name*> **[-help]**

backup labeltape [-name <*tape name, defaults to NULL*>**]**
[-size <*tape size in Kbytes, defaults to size in tapeconfig*>**]**
[-portoffset <*TC port offset*>**] [-help]**

backup listdumps [-help]

backup listhosts [-help]

backup listvolsets [-name <*volume set name*>**] [-help]**

(backup) quit [-help]

backup readlabel [-portoffset <*TC port offset*>**] [-help]**

backup restoredb [-portoffset <*TC port offset*>**] [-help]**

backup savedb [-portoffset <*TC port offset*>**] [-archive** <*date time*>**]**
[-help]

backup scantape [-dbadd] [-portoffset <*TC port offset*>**] [-help]**

backup setexp -dump <*dump level name*>$^{+}$ **[-expires** <*expiration date*>$^{+}$**] [-help]**

backup status [-portoffset <*TC port offset*>**] [-help]**

backup volinfo -volume <*volume name*> **[-help]**

backup volrestore -server <*destination machine*>
-partition <*destination partition*> **-volume** <*volume(s) to restore*>$^{+}$
[-extension <*new volume name extension*>**] [-date** <*date from which to restore*>**]**
[-portoffset <*TC port offsets*>$^{+}$**] [-n] [-help]**

11.2. Summary of bos Commands

bos addhost **-server** <*machine name*> **-host** <*host name*>+ **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

bos addkey **-server** <*machine name*> **-key** <*key*> **-kvno** <*key version number*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos adduser **-server** <*machine name*> **-user** <*user names*>+ **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

bos apropos **-topic** <*help string*> **[-help]**

bos create **-server** <*machine name*> **-instance** <*server process name*>
-type <*server type*> **-cmd** <*command lines*>+ **[-notifier** <*Notifier program*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos delete **-server** <*machine name*> **-instance** <*server process name*>+
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos exec **-server** <*machine name*> **-cmd** <*command to execute*> **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

bos getdate **-server** <*machine name*> **-file** <*files to check*>+ **[-dir** <*destination dir*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos getlog **-server** <*machine name*> **-file** <*log file to examine*> **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

bos getrestart **-server** <*machine name*> **[-cell** <*cell name*>**] [-noauth]**
[-localauth] [-help]

bos help **[-topic** <*help string*>+**] [-help]**

bos install **-server** <*machine name*> **-file** <*files to install*>+ **[-dir** <*destination dir*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos listhosts **-server** <*machine name*> **[-cell** <*cell name*>**] [-noauth]**
[-localauth] [-help]
alias: **bos getcell**

bos listkeys **-server** <*machine name*> **[-cell** <*cell name*>**] [-noauth]**
[-localauth] [-help]

bos listusers **-server** <*machine name*> **[-cell** <*cell name*>**] [-noauth]**
[-localauth] [-help]

bos prune **-server** <*machine name*> **[-bak] [-old] [-core] [-all] [-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

bos removehost **-server** <*machine name*> **-host** <*host name*>+ **[-cell** <*cell name*>**]**

[-noauth] [-localauth] [-help]

bos removekey -server <*machine name*> **-kvno** <*key version number*>+
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos removeuser -server <*machine name*> **-user** <*user names*>+ **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

bos restart -server <*machine name*> **[-instance** <*instances*>+**] [-bosserver] [-all]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos salvage -server <*machine name*> **[-partition** <*salvage partition*>**]**
[-volume <*salvage volume number or volume name*>**] [-file** <*salvage log output file*>**]**
[-all] [-showlog] [-parallel <*# of max parallel partition salvaging*>**]**
[-tmpdir <*directory to place tmp files*>**] [-cell** <*cell name*>**] [-noauth]**
[-localauth] [-help]

bos setauth -server <*machine name*>
-authrequired <*on or off: authentication required for admin requests*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos setcellname -server <*machine name*> **-name** <*cell name*> **[-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

bos setrestart -server <*machine name*> **-time** <*time to restart server*>
[-general] [-newbinary] [-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos shutdown -server <*machine name*> **[-instance** <*instances*>+**] [-wait]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos start -server <*machine name*> **-instance** <*server process name*>+
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos startup -server <*machine name*> **[-instance** <*instances*>+**] [-cell** <*cell name*>**]**
[-noauth] [-localauth] [-help]

bos status -server <*machine name*> **[-instance** <*server process name*>+**] [-long]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos stop -server <*machine name*> **-instance** <*server process name*>+ **[-wait]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

bos uninstall -server <*machine name*> **-file** <*files to uninstall*>+ **[-dir** <*destination dir*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-help]**

11.3. Summary of fs Commands

fs apropos -topic <*help string*> **[-help]**

fs checkservers [-cell <*cell to check*>**] [-all] [-fast] [-interval** <*seconds between probes*>**]**
[-help]

fs checkvolumes [-help]

fs cleanacl [-path <*dir/file path*>+**] [-help]**

fs copyacl -fromdir <*source directory*> **-todir** <*destination directory*>+ **[-clear] [-help]**

fs debug -debug <*'on' or 'off'*> **[-dafs** <*afs debug level*>**] [-dnet** <*network debug level*>**]**
[-syslog] [-help]

fs diskfree [-path <*dir/file path*>+**] [-help]**
alias: **fs df**

fs examine [-path <*dir/file path*>+**] [-help]**
aliases: **fs lv** and **fs listvol**

fs exportafs -type <*exporter name*> **[-state** <*'on' or 'off'*>**] [-noconvert] [-help]**

fs flush [-path <*dir/file path*>+**] [-help]**

fs flushvolume [-path <*dir/file path*>+**] [-help]**

fs getcacheparms [-help]

fs getcellstatus -cell <*cell name*>+ **[-help]**

fs getserverprefs [-file <*dir/file path*>**] [-numeric] [-help]**
alias: **fs gp**

fs help [-topic <*help string*>+**] [-help]**

fs listacl [-path <*dir/file path*>+**] [-help]**
alias: **fs la**

fs listcells [-help]

fs listquota [-path <*dir/file path*>+**] [-help]**
alias: **fs lq**

fs lsmount -dir <*directory*>+ **[-help]**

fs messages [-show <*user*|*console*|*all*|*none*>**] [-log** <*user*|*console*|*all*|*none*>**] [-help]**

fs mkmount -dir <*directory*> **-vol** <*volume name*> **[-cell** <*cell name*>**] [-rw] [-fast]**
[-root] [-help]

fs monitor [-server <*host name or 'off'*>**] [-help]**

fs newcell -name <*cell name*> **-servers** <*primary servers*>+ **[-help]**

fs quota [-path <*dir/file path*>+**] [-help]**

fs rmmount -dir <*directory*>+ **[-help]**

fs setacl -dir <*directory*>+ **-acl** <*access list entries*>+ **[-clear] [-negative] [-help]**
alias: **fs sa**

fs setcachesize [-blocks <*size in 1K byte blocks (0 => reset*>**] [-reset] [-help]**
alias: **fs cachesize**

fs setcell -cell <*cell name*>+ **[-suid] [-nosuid] [-help]**

fs setquota [-path <*dir/file path*>**] -max** <*max quota in kbytes*> **[-help]**
alias: **fs sq**

fs setserverprefs [-servers <*machine name and rank*>+**] [-file** <*dir/file path*>**]**
[-stdin] [-help]
alias: **fs sp**

fs setvol [-path <*dir/file path*>+ **[-max** <*disk space quota in 1K units*>**]**
[-motd <*message of the day*>**] [-offlinemsg** <*offline message*>**] [-help]**
alias: **fs sv**

fs sysname [-newsys <*new sysname*>**] [-help]**

fs whereis [-path <*dir/file path*>+**] [-help]**

fs whichcell [-path <*dir/file path*>+**] [-help]**

fs wscell [-help]

11.4. Summary of kas Commands

kas apropos -topic <*help string*> **[-help]**

kas create -name <*name of user*> **[-initial_password** <*initial password*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth] [-help]**

kas debuginfo [-hostname <*authentication server host name*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth] [-help]**

kas delete -name <*name of user*>
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth] [-help]**
alias: **kas rm**

kas examine -name <*name of user*>
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth] [-help]**
alias: **kas ls**

kas forgetticket [-name <*name of server*>**] [-all] [-help]**

kas getpassword -name <*name of user*> **[-help]**

kas getrandomkey [-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth] [-help]**

kas getticket -name <*name of server*> **[-lifetime** <*ticket lifetime*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth]**
[-help]

kas help [-topic <*help string*>+**] [-help]**

kas interactive [-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth] [-help]**

kas list [-long] [-showadmin]
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>+**] [-noauth] [-help]**

kas listtickets [-name <*name of server*>**] [-long] [-help]**

kas noauthentication [-help]

kas quit [-help]

kas setfields -name <*name of user*> **[-flags** <*hex flag value or flag name expression*>**]**
[-expiration <*date of account expiration*>**] [-lifetime** <*maximum ticket lifetime*>**]**
[-pwexpires <*number days password is valid ([0..254])*>**]**
[-reuse <*permit password reuse (yes/no)*>**]**
[-attempts <*maximum successive failed login tries ([0..254])*>**]**
[-locktime <*failure penalty [hh:mm or minutes]*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**] [-noauth] [-help]**
alias: **kas sf**

kas setkey -name <*name of user*> **-new_key** <*eight byte new key*>
[-kvno <*key version number*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**] [-noauth] [-help]**

kas setpassword -name <*name of user*> **[-new_password** <*new password*>**]**
[-kvno <*key version number*>**]**
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**] [-noauth] [-help]**
alias: **kas sp**

kas statistics [-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**] [-noauth] [-help]**

kas stringtokey -string <*password string*> **[-cell** <*cell name*>**] [-help]**

kas unlock -name <*authentication ID*>
[-admin_username <*admin principal to use for authentication*>**]**
[-password_for_admin <*admin password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of authentication servers*>$^{+}$**] [-noauth] [-help]**

11.5. Summary of package Command and Template Lines

package [initcmd] [-config <*base name of configuration file*>**]**
[-fullconfig <*full name of configuration file, or stdin for standard input*>**]**
[-overwrite] [-noaction] [-verbose] [-silent] [-rebootfiles] [-debug]
[-help]

package apropos [-topic <*help string*>+**] [-help]**

package help [-topic <*help string*>+**] [-help]**

B <*device name*> <*major device number*> <*minor device number*> <*owner name*>
<*group name*> <*mode bits*>

C <*device name*> <*major device number*> <*minor device number*> <*owner name*>
<*group name*> <*mode bits*>

D[<*update code*>] <*directory*> <*owner name*> <*group name*> <*mode bits*>

F[<*update code*>] <*file*> <*source file*> [<*owner name*> <*group name*> <*mode bits*>]

L[<*update code*>] <*link*> <*actual file*> [<*owner name*> <*group name*> <*mode bits*>]

S <*socket name*> [<*owner name*> <*group name*> <*mode bits*>]

%define <*variable*> <*value*>
%define <*declaration*>

%ifdef <*declaration*>
<*action*>+
[%else [<*declaration*>]
<*alternate action*>+]
%endif <*declaration*>

%ifndef <*declaration*>
<*action*>+
[%else [<*declaration*>]
<*alternate action*>+]
%endif <*declaration*>

%include <*file pathname*>

%undef <*declaration*>

11.6. Summary of pts Commands

pts adduser -user <*user name*>+ **-group** <*group name*>+ **[-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

pts apropos -topic <*help string*> **[-help]**

pts chown -name <*group name*> **-owner** <*new owner*> **[-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

pts creategroup -name <*group name*>+ **[-owner** <*owner of the group*>**]**
[-id <*id (negated) for the group*>+**] [-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]
alias: **pts cg**

pts createuser -name <*user name*>+ **[-id** <*user id*>+**] [-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]
alias: **pts cu**

pts delete -nameorid <*user or group name or id*>+ **[-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

pts examine -nameorid <*user or group name or id*>+ **[-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

pts help [-topic <*help string*>+**] [-help]**

pts listmax [-cell <*cell name*>**] [-noauth] [-test] [-force] [-help]**

pts listowned -nameorid <*user or group name or id*>+ **[-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

pts membership -nameorid <*user or group name or id*>+ **[-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]
alias: **pts groups**

pts removeuser -user <*user name*>+ **-group** <*group name*>+ **[-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

pts rename -oldname <*old name*> **-newname** <*new name*> **[-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

pts setfields -nameorid <*user or group name or id*>+ **[-access** <*set privacy flags*>**]**
[-groupquota <*set limit on group creation*>**] [-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

pts setmax [-group <*group max*>**] [-user** <*user max*>**] [-cell** <*cell name*>**] [-noauth]**
[-test] [-force] [-help]

11.7. Summary of uss Commands and Template Lines

uss add -user <*login name*> **[-realname** <*full name in quotes*>**]**
[-pass <*initial passwd*>**] [-server** <*FileServer for home volume*>**]**
[-partition <*FileServer's disk partition for home volume*>**]**
[-mount <*home directory mount point*>**] [-uid** <*uid to assign the user*>**]**
[-template <*pathname of template file*>**] [-verbose]**
[-var <*auxiliary argument pairs (Num val)*>+**] [-cell** <*cell name*>**]**
[-admin <*administrator to authenticate*>**] [-dryrun] [-skipauth] [-overwrite]**
[-pwexpires <*password expires in [0..254] days (0 => never)*>**] [-help]**

uss apropos -topic <*help string*> **[-help]**

uss bulk -file <*bulk input file*> **[-template** <*pathname of template file*>**]**
[-verbose] [-cell <*cell name*>**] [-admin** <*administrator to authenticate*>**]**
[-dryrun] [-skipauth] [-overwrite] [-help]

uss delete -user <*login name*> **-mountpoint** <*mountpoint for user's volume*>
[-savevolume] [-verbose] [-cell <*cell name*>**]**
[-admin <*administrator to authenticate*>**] [-dryrun] [-skipauth] [-help]**

uss help [-topic <*help string*>+**] [-help]**

A <*username*> <*password lifetime*> <*reuse/noreuse*> <*failures*> <*locktime*>

D <*pathname*> <*mode*> <*owner*> <*access list*>

E <*pathname*> <*mode*> <*owner*> "<*contents*>"

F <*pathname*> <*mode*> <*owner*> <*prototype*>

G <*directory*>

L <*existing-file*> <*link*>

S <*existing-file*> <*link*>

V <*volume name*> <*server*> <*partition*> <*quota*> <*mount point*> <*owner*> <*access list*>

X "<*command*>"

11.8. Summary of vos Commands

vos addsite -server <*machine name for new site*>
-partition <*partition name for new site*> **-id** <*volume name or ID*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

vos apropos -topic <*help string*> **[-help]**

vos backup -id <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos backupsys [-prefix <*common prefix on volume(s)*>**] [-server** <*machine name*>**]**
[-partition <*partition name*>**] [-exclude] [-cell** <*cell name*>**] [-noauth]**
[-localauth] [-verbose] [-help]

vos create -server <*machine name*> **-partition** <*partition name*>
-name <*volume name*> **[-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos delentry [-id <*volume name or ID*>+**]**
[-prefix <*prefix of the volume whose VLDB entry is to be deleted*>**]**
[-server <*machine name*>**] [-partition** <*partition name*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

vos dump -id <*volume name or ID*> **-time** <*dump from time*> **[-file** <*arg*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

vos examine -id <*volume name or ID*> **[-extended] [-cell** <*cell name*>**] [-noauth]**
[-localauth] [-verbose] [-help]

vos help [-topic <*help string*>+**] [-help]**

vos listpart -server <*machine name*> **[-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos listvldb [-name <*volume name or ID*>**] [-server** <*machine name*>**]**
[-partition <*partition name*>**] [-locked] [-quiet] [-cell** <*cell name*>**]**
[-noauth] [-localauth] [-verbose] [-help]

vos listvol -server <*machine name*> **-partition** <*partition name*> **[-fast]**
[-long] [-quiet] [-extended] [-cell <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos lock -id <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos move -id <*volume name or ID*> **-fromserver** <*machine name on source*>
-frompartition <*partition name on source*> **-toserver** <*machine name on destination*>
-topartition <*machine name on destination*> **[-cell** <*cell name*>**] [-noauth]**
[-localauth] [-verbose] [-help]

vos partinfo -server <*machine name*> **[-partition** <*partition name*>**] [-cell** <*cell name*>**]**

[-noauth] [-localauth] [-verbose] [-help]

vos release -id <*volume name or ID*> **[-f] [-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos remove -server <*machine name*> **-partition** <*partition name*>
-id <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos remsite -server <*machine name*> **-partition** <*partition name*>
-id <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth]**
[-localauth] [-verbose] [-help]

vos rename -oldname <*old volume name*> **-newname** <*new volume name*>
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

vos restore -server <*machine name*> **-partition** <*partition name*>
-name <*name of volume to be restored*> **[-file** <*dump file*>**]**
[-id <*volume ID*>**] [-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos status -server <*machine name*> **[-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos syncserv -server <*machine name*> **[-partition** <*partition name*>**] [-force]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

vos syncvldb -server <*machine name*> **[-partition** <*partition name*>**] [-force]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

vos unlock -id <*volume name or ID*> **[-cell** <*cell name*>**] [-noauth] [-localauth]**
[-verbose] [-help]

vos unlockvldb [-server <*machine name*>**] [-partition** <*partition name*>**]**
[-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

vos zap -server <*machine name*> **-partition** <*partition name*> **-id** <*volume ID*>
[-force] [-backup] [-cell <*cell name*>**] [-noauth] [-localauth] [-verbose] [-help]**

11.9. Summary of AFS Miscellaneous Commands

afsd [-blocks <*1024 byte blocks in cache*>**] [-files** <*files in cache*>**]**
[-rootvol <*name of AFS root volume*>**] [-stat** <*number of stat entries*>**]**
[-memcache] [-cachedir <*cache directory*>**] [-mountdir** <*mount location*>**]**
[-daemons <*number of daemons to use*>**] [-nosettime] [-verbose] [-rmtsys]**
[-debug] [-chunksize <*log(2) of chunk size*>**] [-dcache** <*number of dcache entries*>**]**
[-volumes <*number of volume entries*>**]**
[-biods <*number of bkg I/O daemons (aix vm)*>**]**
[-prealloc <*number of 'small' preallocated blocks*>**]**
[-confdir <*configuration directory*> **[-logfile** <*Place to keep the CM log*>**]**
[-waitclose] [-help]

afsmonitor [initcmd] [-config <*configuration file*>**]**
[-frequency <*poll frequency, in seconds*>**] [-output** <*storage file name*>**]**
[-detailed] [-debug <*turn debugging output on to the named file*>**]**
[-fshosts <*list of file servers to monitor*>$^{+}$**]**
[-cmhosts <*list of cache managers to monitor*>$^{+}$**]**
[-buffers <*number of buffer slots*>**] [-help]**

/usr/afs/bin/bosserver [-noauth] [-log]

budb_convert

/usr/afs/bin/buserver [-database <*database directory*>**]**
[-cellservdb <*cell configuration directory*>**] [-resetdb] [-noauth] [-smallht]**
[-servers <*list of ubik database servers*>$^{+}$**] [-help]**

butc [-port <*port offset*>**] [-debuglevel** <*debug level*>**] [-cell** <*cell name*>**]**
[-aixscsi] [-noautoquery] [-help]

dkload [-readOnly] [-quiet] [-verbose] [-syscallresult <*number*>**]**
[-path <*object path*>**] [-ld_cmd** <*loader path*>**] [-as_cmd** <*assembler path*>**]**
[-nm_cmd <*nm path*>**] [-libcommon** <*dkload libcommon.a path*>**]**
[-kernel_alloc <*dkload kernel allocator path*>**] {-name** <*mod*> | <*object files*>$^{+}$**}**
[-dir <*scratch dir for dkload*>**] [-deletedir]**

/usr/afs/bin/fileserver [-d <*debug level*>**] [-p** <*number of processes*>**]**
[-spare <*number of spare blocks*>**] [-pctspare** <*percentage spare*>**] [-b** <*buffers*>**]**
[-l <*large vnodes*>**] [-s** <*small vnodes*>**] [-vc** <*volume cachesize*>**]**
[-w <*call back wait interval*>**] [-cb** <*number of call backs*>**]**
[-banner <*print banner every 10 minutes*>**]**
[-hr <*number of hours between refreshing the host cps*>**]**
[-m <*min percentage spare in partition*>**] [-L** <*large server conf*>**]**
[-S <*Small server conf*>**] [-k** <*stack size*>**] [-c** <*check interval*>**] [-help]**

fms -tape <*tape special file*> **[-help]**

/usr/afs/bin/kaserver [-noAuth] [-fastKeys] [-database *dbpath***] [-localfiles** *lclpath***]**
[-minhours <*n*>**] [-servers** <*serverlist*>**]**

kdb [-dbmfile <*dbmfile to use (default /usr/afs/logs/AuthLog)*>**]**
[-key <*extract entries that match specified key*>**] [-help]**

klog [-x] [-principal <*user name*>**] [-password** <*user's password*>**] [-tmp]**
[-cell <*cell name*>**] [-servers** <*explicit list of servers*>$^{+}$**] [-pipe]**
[-lifetime <*ticket lifetime in hh[:mm[:ss]]*>**] [-help]**

knfs -host <*host name*> **[-id** <*user ID (decimal)*>**]**
[-sysname <*host's '@sys' value*>**] [-unlog] [-tokens] [-help]**

kpasswd [-x] [-principal <*user name*>**] [-password** <*user's password*>**]**
[-newpassword <*user's new password*>**] [-cell** <*cell name*>**]**
[-servers <*explicit list of servers*>$^{+}$**] [-pipe] [-help]**

package_test <*config file*>

pagsh

/usr/afs/bin/ptserver [-database <*db path*>**] [-rebuild]**

/usr/afs/bin/runntp [-localclock] [-precision <*small negative integer*>**]**
[-logfile <*pathname*>**] [**<*machine name*>$^{+}$**]**

/usr/afs/bin/salvager [initcmd] [-partition <*Name of partition to salvage*>**]**
[-volumeid <*Volume Id to salvage*>**] [-debug] [-nowrite] [-inodes] [-force]**
[-oktozap] [-rootinodes] [-salvagedirs] [-blockreads]
[-parallel <*# of max parallel partition salvaging*>**]**
[-tmpdir <*directory to place tmp files*>**] [-help]**

scout [initcmd] -server <*FileServer name(s) to monitor*>$^{+}$
[-basename <*base server name*>**] [-frequency** <*poll frequency, in seconds*>**]**
[-host] [-attention <*specify attention (highlighting) level*>$^{+}$**]**
[-debug <*turn debugging output on to the named file*>**] [-help]**

translate_et <*error number*>$^{+}$

tokens

unlog [-cell <*cell name*>$^{+}$**] [-help]**

up [-d] [-v] [-s] [-1] [-n] [-o] [-f] [-r] [-x] <*source directory*> <*destination directory*>

/usr/afs/bin/upclient <*hostname*> **[-level** <*clear*|*crypt*>**] [-clear] [-crypt]**
[-t <*retry time*>**]** <*dir*>$^{+}$

/usr/afs/bin/upserver [<*directory*>$^{+}$**] [crypt** <*directory*>$^{+}$**] [clear** <*directory*>$^{+}$**]**

vldb_convert [initcmd] [-to <*goal version*>**] [-from** <*current version*>**]**
[-path <*pathname*>**] [-showversion] [-help]**

/usr/afs/bin/vlserver [-p <*lwp processes*>**]**

/usr/afs/bin/volserver [-verbose <*debuglevel*>**] [-log] [-help]**

11.10. Summary of Modified UNIX Commands

/usr/etc/ftpd [-d] [-l] [-t <*timeout*>**] [-v] [-T** <*MaxTimeOut*>**] [-u] [-s]**

/etc/inetd [-d] [<*configfile*>**]**

login [<*user name*>**]**

rcp [-p] <*file1*> <*file2*>
rcp [-r] [-p] <*file*>$^{+}$ <*directory*>

/etc/rlogind [-l] [-n] [-s]

rsh host [-l <*username*>**] [-n]** <*command*>
host [-l <*username*>**] [-n]** <*command*>

Index

ContentsEnv tc0

.BAK version of binary file
creation by bos install command 2-32
listing time stamp on 2-24
use by bos uninstall command 2-74
.OLD version of binary file
creation by bos install command 2-32
listing time stamp on 2-24
use by bos uninstall command 2-74
.backup extension on volume name
added by vos backup command 8-11
added by vos backupsys command 8-12
.readonly extension on volume name
added by vos release command 8-44

/etc/package 5-1
/usr/afs/bin
checking time stamps 2-24
removing .BAK files from 2-41
removing .OLD files from 2-41
/usr/afs/etc/CellServDB
See also CellServDB file (server)
/usr/afs/etc/KeyFile
See also KeyFile file
/usr/afs/etc/ThisCell
See also ThisCell file (server)
/usr/afs/etc/UserList
See also UserList file
/usr/afs/local/BosConfig
See also BosConfig file
/usr/afs/local/NoAuth
See also NoAuth file
/usr/afs/logs
examining log files in 2-26
removing core files from 2-41
/usr/vice/etc/ThisCell
See also ThisCell file (client)
/usr/vice/etc/cacheinfo
See also Cacheinfo file

@sys variable in pathnames 3-65

ACL
adding entries 3-51
cleaning 3-8
clearing 3-51
copying 3-10
default rights assigned when volume created 6-3
default set by vos create 8-15
defined 6-1
listing 3-30
removing entries 3-51
removing obsolete AFS UIDs 3-8
setting 3-51
setting with uss 7-21, 7-31
system groups on 6-3
ADMIN flag in Authentication Database entry 4-6, 4-38
AFS 2.0
volume location method 3-23
AFS Backup System
See also Backup System
AFS UID
assigning for group 6-12
assigning in uss 7-4
assigning to user 6-15
defined 6-2
learning, given user/group name 6-20
removing obsolete from ACL 3-8
AFS server process
See also Server process
AFSMonitor
setting terminal type 9-11
Access control list
See also ACL, copying
Access rights
shorthand for 3-52
See also ACL
Accesses
of volume, counter of 8-4
Account
See also User account
Active clients statistic in scout 9-69
Adding
ReadOnly site definition in VLDB 8-8
access rights entry to ACL 3-51
database server machine to CellServDB file (server) 2-9
machine to group 6-7
privileged user to UserList file 2-13
server encryption key to KeyFile file 2-11
user to group 6-7
ADMINISTER access right 3-31, 3-52
"all" grouping of access rights 3-52
All-or-nothing release of ReadOnly volumes 8-44
Apropos commands xix
Argument
defined xi
list-taking xi
See also Common arguments
Assigning
AFS UID to group 6-12
AFS UID to user 6-15
@sys variable in pathnames 3-65
Authenticated identity
acquiring on NFS client of non-supported type 9-49
acquiring with klog 9-43
discarding while in kas interactive mode 4-35
Authentication Database
changing/setting password in 9-52
creating entry 4-8
creating entry with uss 7-4
deleting entry 4-12
deleting entry with uss 7-15
information in 4-3, 4-4
listing all entries 4-30

listing entry 4-14
listing key from 4-14
setting flags and expiration dates 4-37
setting key field directly in octal form 4-42
setting key field using password 4-45
setting password in 4-45
Authentication Server 4-1
debugging trace 4-10
examining statistics from 4-49
listed in CellServDB file 2-5
reading files associated with 9-41
starting 9-39
unlocking locked user account 4-53
See also kaserver process
Authorization
See also Privilege required
Authorization checking
setting requirements on file server machine 2-55
Availability of data
interrupted by dumping 8-20

BAK version of binary file
See also .BAK (at beginning of Index)
BOS Server 2-1
memory state 2-4
restart times, examining 2-28
restart times, setting 2-59
restarting 2-49
starting 9-17
Backup Database Server
starting 9-22
Backup
See also Backup System
Backup Database 1-1
adding dump level to dump hierarchy 1-5
adding volume entry to volume set 1-10
checking 1-15
converting from 3.1 to 3.2 9-19
creating Tape Coordinator entry 1-8
creating volume set 1-13
defining port offset number for Tape Coordinator 1-8
information in 1-1
listing Tape Coordinators 1-48
listing dump hierarchy 1-46
listing dump history for volume 1-67
listing used port offset numbers 1-48
listing volume entries in volume set 1-50
removing Tape Coordinator entry 1-20
removing dump level from dump hierarchy 1-17
removing volume entry from volume set 1-22
removing volume set 1-23
restoring 1-55
saving 1-57
setting expiration date on existing dump level 1-63
setting expiration date on new dump level 1-5
verifying 1-15
Backup System
aborting operation in interactive mode 1-41
checking Backup Database 1-15
converting 3.1 Backup Database to 3.2 9-19
database (see entry: *Backup Database*) 1-1
defining capacity of tape 1-43
determining end-of-file mark size for tape drive 9-36
determining tape capacity 9-36
job ID number, listing 1-39
job ID number, using to kill operation 1-41
labeling tape 1-43
listing capacity of tape 1-53
reading label on tape 1-53
regular expressions in 1-10
restoring Backup Database 1-55
saving Backup Database 1-57
starting Backup Database Server 9-22
starting Tape Coordinator 9-25
the Backup Database 1-1
verifying Backup Database 1-15
Backup commands
common arguments 1-2
privilege requirements 1-3, 1-4
Backup extension on volume name
See also .backup extension on volume name
Backup volume
ID number in VLDB 8-2
ID number in volume header 8-3
changing name of 8-51
creating 8-11
creating many at once 8-12
dumping 8-20
moving 8-39
removed by ReadWrite move 8-39
removed by ReadWrite removal 8-47
removing 8-47
Basic OverSeer Server
See also BOS Server
Binary distribution machine 2-32, 2-74
Binary file
installing 2-32
listing time stamp on 2-24
uninstalling 2-74
Block special device
defining with package 5-7
BosConfig file
controlling process status via 2-5
deleting server process from 2-21
examining server process entry from 2-68
function defined 2-2
how BOS Server uses 2-4
status flag in 2-3
Bulk mode in uss 7-10

Cache
See also Data cache
Cache Manager
changing database server machines known to 3-46
configuration scripts 9-5
configuring with afsd 9-2
configuring with fs commands 3-1
debugging trace 3-12
disabling messages 3-38
displaying file server machine preferences 3-25
flushing directory/file from data cache 3-20
flushing entire volume from data cache 3-21
initializing with afsd 9-2
listing amount of cache used 3-22
listing cache size 3-22
listing database server machines known to 3-33
listing down file server machines 3-4
logging messages 3-38
monitoring activity 3-44
monitoring status with afsmonitor 9-9

setting cache size 3-54
setting file server machine preferences 3-59
setting the interval between file server probes 3-4
volume name to ID mapping, forcing update of 3-7
Cacheinfo file 3-54
Cell
client's kernel-resident list of, changing 3-46
client's kernel-resident list of, listing 3-33
name of, setting 2-57
setuid status, listing 3-23
setuid status, setting 3-56
use of AFS 2.0 volume location method 3-23
CellServDB file (client)
transferred into kernel 3-33
See also Client machine
CellServDB file (server)
adding database server machine 2-9
creating initial version of 2-57
function defined 2-5
listing contents of 2-34
removing database server machine 2-43
setting cellname in 2-57
Changing
data cache size 3-54
database server machines listed in client kernel 3-46
name of Protection Database entry 6-34
owner of Protection Database entry 6-10
password in Authentication Database 9-52
volume name 8-51
volume quota 3-58, 3-63
Character special device
defining with package 5-8
Character string
converting to octal key form 4-51
Checking
AFS client as exporter non-AFS file system 3-18
file server machine status 3-4
See also Examining, Listing
Cleaning
ACL 3-8
Clearing
ACL 3-51
Client machine
as exporter of non-AFS file system 3-18
changing database server machines known to 3-46
changing size of data cache 3-54
configuration scripts 9-5
configuring local disk with package 5-2
listing database server machines known to 3-33
listing home cell of 3-69
system type 3-65
Client machines statistic in scout 9-69
Client portion of Update Server
See also Upclient process
Clone 8-44
forcing creation of new 8-44
Cloning
for backup 8-11, 8-12
for replication 8-44
Command suite
defined vii, xi
Commands
afsd 9-2
afsmonitor 9-9
argument defined xi
backup adddump 1-5
backup addhost 1-8
backup addvolentry 1-10
backup addvolset 1-13
backup apropos 1-14
backup dbverify 1-15
backup deldump 1-17
backup deletedump 1-18
backup delhost 1-20
backup delvolentry 1-22
backup delvolset 1-23
backup diskrestore 1-24
backup dump 1-27
backup dumpinfo 1-33
backup help 1-37
backup jobs 1-39
backup kill 1-41
backup labeltape 1-43
backup listdumps 1-46
backup listhosts 1-48
backup listvolsets 1-50
backup quit 1-52
backup readlabel 1-53
backup restoredb 1-55
backup savedb 1-57
backup scantape 1-59
backup setexp 1-63
backup status 1-65
backup volinfo 1-67
backup volrestore 1-69
bos addhost 2-9
bos addkey 2-11
bos adduser 2-13
bos apropos 2-15
bos create 2-16
bos delete 2-21
bos exec 2-23
bos getdate 2-24
bos getlog 2-26
bos getrestart 2-28
bos help 2-30
bos install 2-32
bos listhosts 2-34
bos listkeys 2-36
bos listusers 2-39
bos prune 2-41
bos removehost 2-43
bos removekey 2-45
bos removeuser 2-47
bos restart 2-49
bos salvage 2-51
bos setauth 2-55
bos setcellname 2-57
bos setrestart 2-59
bos shutdown 2-62
bos start 2-64
bos startup 2-66
bos status 2-68
bos stop 2-72
bos uninstall 2-74
bosserver 9-17
budb_convert 9-19
buserver 9-22
butc 9-25
dkload 9-28

executing from uss template file 7-33
executing remotely 2-23
fileserver 9-31
fms 9-36
fs apropos 3-3
fs checkservers 3-4
fs checkvolumes 3-7
fs cleanacl 3-8
fs copyacl 3-10
fs debug 3-12
fs diskfree 3-14
fs examine 3-16
fs exportafs 3-18
fs flush 3-20
fs flushvolume 3-21
fs getcacheparms 3-22
fs getcellstatus 3-23
fs getserverprefs 3-25
fs help 3-28
fs listacl 3-30
fs listcells 3-33
fs listquota 3-34
fs lsmount 3-36
fs messages 3-38
fs mkmount 3-40
fs monitor 3-44
fs newcell 3-46
fs quota 3-48
fs rmmount 3-50
fs setacl 3-51
fs setcachesize 3-54
fs setcell 3-56
fs setquota 3-58
fs setserverprefs 3-59
fs setvol 3-63
fs sysname 3-65
fs whereis 3-67
fs whichcell 3-68
fs wscell 3-69
ftpd (AFS version) 10-2
help (online) for xviii
inetd (AFS version) 10-4
instance defined xi
kas apropos 4-7
kas create 4-8
kas debuginfo 4-10
kas delete 4-12
kas examine 4-14
kas forgetticket 4-18
kas getpassword 4-19
kas getrandomkey 4-21
kas getticket 4-23
kas help 4-25
kas interactive 4-27
kas list 4-30
kas listtickets 4-33
kas noauthentication 4-35
kas quit 4-36
kas setfields 4-37
kas setkey 4-42
kas setpassword 4-45
kas statistics 4-49
kas stringtokey 4-51
kas unlock 4-53
kaserver 9-39
kdb 9-41
keywords for looking up xix
klog 9-43
knfs 9-49
kpasswd 9-52
login (AFS version) 10-7
man xx
operation code defined xi
package 5-2
package apropos 5-4
package help 5-5
package_test 9-56
pagsh 9-57
pts adduser 6-7
pts apropos 6-9
pts chown 6-10
pts creategroup 6-12
pts createuser 6-15
pts delete 6-18
pts examine 6-20
pts help 6-23
pts listmax 6-25
pts listowned 6-27
pts membership 6-29
pts removeuser 6-32
pts rename 6-34
pts setfields 6-36
pts setmax 6-39
ptserver 9-59
rcp (AFS version) 10-11
rlogind (AFS version) 10-13
rsh (AFS version) 10-15
runntp 9-61
salvager 9-63
scout 9-66
suite defined vii, xi
switch defined xi
tokens 9-71
translate_et 9-73
unlog 9-74
up 9-76
upclient 9-78
upserver 9-81
uss add 7-4
uss apropos 7-9
uss bulk 7-10
uss delete 7-15
uss help 7-17
vldb_convert 9-83
vlserver 9-87
volserver 9-89
vos addsite 8-8
vos apropos 8-10
vos backup 8-11
vos backupsys 8-12
vos create 8-15
vos delentry 8-17
vos dump 8-20
vos examine 8-22
vos help 8-26
vos listpart 8-28
vos listvldb 8-30
vos listvol 8-33
vos lock 8-37
vos move 8-39

vos partinfo 8-42
vos release 8-44
vos remove 8-47
vos remsite 8-49
vos rename 8-51
vos restore 8-53
vos status 8-55
vos syncserv 8-57
vos syncvldb 8-59
vos unlock 8-61
vos unlockvldb 8-63
vos zap 8-65
Common arguments
on backup commands 1-2
on bos commands 2-7
on fs commands 3-2
on kas commands 4-4
on pts commands 6-4
on uss commands 7-2
on vos commands 8-5
Compiling
package prototype file 5-1
Configuration file in package 5-1
%define in 5-16
%ifdef in 5-17
%ifndef in 5-18
%include in 5-19
%undef in 5-20
B line in 5-7
C line in 5-8
D line in 5-9
F line in 5-10
L line in 5-13
S line in 5-15
defining block special device in 5-7
defining character special device in 5-8
defining directory in 5-9
defining file in 5-10
defining socket in 5-15
defining symbolic link in 5-13
Configuring
Cache Manager with afsd 9-2
Cache Manager with fs commands 3-1
local disk of client with package 5-2
Connections statistic in scout 9-68
Contacting
File Server with fs commands 3-1
Controlling
Cache Manager with fs commands 3-1
process status with BosConfig entries 2-5
Converting
Backup Database from 3.1 to 3.2 9-19
VLDB from 3.1 to 3.2 9-83
character string to octal key form 4-51
Copying
access control list 3-10
file remotely with rcp command (AFS version) 10-11
files and directories 9-76
Core files
removing from /usr/afs/logs 2-41
Creating
Authentication Database entry 4-8
Authentication Database entry with uss 7-4
Backup volume 8-11
Backup volumes, many at once 8-12
PAG with AFS login 10-7
PAG with pagsh command 9-57
Protection Database entry with uss 7-4
ReadOnly volume 8-44
ReadWrite volume 8-15
directory with uss 7-21, 7-27
file with uss 7-23, 7-25
group in Protection Database 6-12
hard link with uss 7-29
mount point 3-40
mount point with uss 7-31
privileged user in UserList file 2-13
random encryption key 4-21
server encryption key 2-11
server process in BosConfig file 2-16
symbolic link with uss 7-30
ticket (tokens) for server process 9-43
ticket for server process 4-23
user account with uss 7-4
user accounts in bulk 7-10
user in Protection Database 6-15
volume (see type--ReadWrite, etc.) 8-15
volume with uss 7-4, 7-31
See also Defining
Creation date
recorded in volume header 8-4
Creator
of Protection Database entry 6-2
of Protection Database entry, listing 6-20
Cron-type process 2-4
Curses graphics utility
and AFSMonitor 9-11

Daily restart for new process binaries
examining time for 2-28
setting time for 2-59
Data
availability interrupted by dumping 8-20
Data cache
changing size of 3-54
flushing directory/file 3-20
flushing entire volume 3-21
listing amount used 3-22
listing size 3-22
resetting to default size 3-54
setting location with afsd 9-2
setting size with afsd 9-2
Database server machine 2-5
adding to CellServDB file (server) 2-9
changing client kernel list of 3-46
listing from CellServDB file (server) 2-34
listing from client kernel 3-33
removing from CellServDB file (server) 2-43
Date
See also Time stamp
Date-specific restore 1-69
Debugging trace
Authentication Server 4-10
Cache Manager 3-12
Default
ACL 8-15
volume quota 8-15
%define in package conf. file 5-16
Defining
ReadOnly site in VLDB 8-8

block special device with package 5-7
capacity of tape for use in Backup System 1-43
character special device with package 5-8
directory with package 5-9
file with package 5-10
privileged user in UserList file 2-13
server encryption key 2-11
socket with package 5-15
symbolic link with package 5-13
system type of client machine 3-65
See also Creating, Setting
DELETE access right 3-31, 3-52
Deleting
Authentication Database entry 4-12
Authentication Database entry with uss 7-15
Protection Database entry 6-18
Protection Database entry with uss 7-15
VLDB entry (but not volume header) 8-17
server process from BosConfig file 2-21
user account with uss 7-15
volume with uss 7-15
See also Removing
Desynchronization of VLDB/volume headers
fixing 8-57, 8-59
Determining
end-of-file mark size for tape drive 9-36
success of replication 8-44
tape capacity for Backup System 9-36
Device
See also Block special device, Character special device
Directories
/usr/afs/bin 2-24, 2-41
/usr/afs/logs 2-26, 2-41
Directory
creating with uss 7-21, 7-27
defining with package 5-9
flushing from cache 3-20
listing ACL 3-30
listing home cell 3-68
overwriting with uss 7-4
setting ACL 3-51
Directory/file name
translating to volume ID number 3-16
translating to volume name 3-14, 3-16, 3-34
Disabling
Cache Manager debugging trace 3-12
export of non-AFS file system 3-18
Discarding
authenticated identity in kas interactive mode 4-35
ticket/tokens 9-74
tickets 4-18
Disk partition
listing blocks available 3-14, 3-16, 8-42
listing blocks used 3-14
listing for file server machine 8-28
listing percent used 3-14
listing size 3-14, 3-16, 8-42
monitoring usage of 9-69
monitoring usage with scout 9-66
moving volumes to reduce overcrowding 8-39
restoring all volumes on (with Backup System) 1-24
Disk, local
See also Local disk
Dumb terminal
use with AFSMonitor 9-11
Dump hierarchy (Backup System)
adding dump level to 1-5
listing 1-46
removing dump level from 1-17
Dump level (Backup System)
adding to dump hierarchy 1-5
listing 1-46
removing from dump hierarchy 1-17
setting expiration date on existing 1-63
setting expiration date on new 1-5
Dumping volumes
using AFS Backup System 1-27
without using Backup System 8-20

Enabling
Authentication Server debugging trace 4-10
Cache Manager debugging trace 3-12
export of non-AFS file system 3-18
Encryption key
in Authentication Database 4-3, 4-31
random generation of 4-21
See also Server encryption key
End-of-file mark
determining size of 9-36
Entering
kas interactive mode 4-27
Entry
in Authentication Database (see entry: *Authentication Database*) 4-3, 4-4
Entry in VLDB
information in 8-2
listing information from 8-30
listing, with volume header 8-22
locking 8-37
unlocking 8-61, 8-63
Error codes
translating numbers to messages 9-73
Examining
Authentication Database entry 4-14
BOS Server's restart times 2-28
Protection Database entry 6-20
VLDB entry 8-30
VLDB entry with volume header 8-22
log files on file server machine 2-26
server process' entry in BosConfig file 2-68
statistics from Authentication Server 4-49
time stamp on binary file 2-24
volume header 8-33
volume header with VLDB entry 8-22
See also Checking, Listing
Examples
apropos operation code xix
help operation code xviii
Executing
command using uss template line 7-33
commands remotely 2-23
Expiration date
of Authentication Database entry, listing 4-14
of Authentication Database entry, setting 4-37
setting on existing dump level 1-63
setting on new dump level 1-5
Export of non-AFS file system
by AFS client 3-18

Failed authentication attempts
listing limit from Authentication Database entry 4-14
-fast flag on vos listvol command 8-33
Fetching statistic in Scout 9-69
File
copying remotely with rcp command (AFS version) 10-11
creating with uss 7-23, 7-25
defining with package 5-10
flushing from cache 3-20
listing ACL on parent directory 3-30
listing home cell 3-68
overwriting with uss 7-4
setting ACL on parent directory 3-51
File Server
as part of fs process 2-4
contacting with fs commands 3-1
monitoring status with afsmonitor 9-9
monitoring status with scout 9-66
starting 9-31
File server machine
as volume location 3-67
checking status of 3-4
examining log files on 2-26
listing AFS partitions on 8-28
listing disk partition size and space available 8-42
maintaining clock with NTPD 9-61
monitoring outages of 9-66
restart time for new binaries, examining 2-28
restart time for new binaries, setting 2-59
restoring all volumes on partition of (with Backup System) 1-24
salvaging volumes on 2-51
setting authorization checking requirements 2-55
weekly restart time, examining 2-28
weekly restart time, setting 2-59
File system
export of non-AFS by AFS client 3-18
monitoring activity 3-44
restoring internal consistency (salvaging) 2-51
File-system salvager
See also Salvager
Files
BosConfig 2-2
Cacheinfo 3-54
CellServDB (server) 2-5
KeyFile 2-6
NoAuth 2-55
ThisCell (server) 2-57
UserList 2-6
Flag xii
See also Privacy flag, Site flags, Status flag, Type flags
Flushing
directory/file from data cache 3-20
entire volume from data cache 3-21
Full dump 8-20
Full restore 1-69

Gathering
statistics from Authentication Server 4-49
Generating
random encryption key 4-21
Getting help xviii
for fs commands 3-28
for kas commands 4-25
for pts commands 6-23
for uss commands 7-17
for vos commands 8-26
Group
adding members 6-7
changing name in Protection Database 6-34
creating in Protection Database 6-12
defined 6-1
deleting from Protection Database 6-18
listing groups owned 6-27
listing members 6-29
listing number of members 6-20
mapping name to AFS UID 6-20
orphaned, listing 6-27
prefix-less 6-12
regular 6-12
removing members 6-32
restrictions on name format 6-12
setting AFS UID of 6-12
setting privacy flags on 6-36
system-defined 6-3
Group-creation quota
defined 6-2
listing 6-20
lowered by pts creategroup 6-12
restored by pts delete 6-18
setting 6-36
GroupMax counter in Protection Database
affected by setting AFS UID 6-12
listing 6-25
setting 6-39

Hard link
creating with uss 7-29
overwriting with uss 7-4
Help
example command xviii
for backup commands 1-37
for fs commands 3-28
for kas commands 4-25
for pts commands 6-23
for uss commands 7-17
for vos commands 8-26
online xviii
See also Keyword
Help commands xviii
Help string
See also Keyword
Home cell
listing for client machine 3-69
listing for directory/file 3-68

Idempotency
of vos commands 8-1
%ifdef in package conf. file 5-17
%ifndef in package conf. file 5-18
%include in package conf. file 5-19
Incorporating
AFS into kernel with rebooting 9-28
Incremental dump 8-20
Initializing
Backup Database Server 9-22
Cache Manager with afsd 9-2
FTP Daemon 10-2

Internet daemon 10-4
NTPD, with runntp 9-61
Protection Server 9-59
Tape Coordinator 9-25
remote login server 10-13
server process 2-16
INSERT access right 3-31, 3-52
Installing
binary file 2-32
Instance xi
Interactive mode
backup, quitting from 1-52
backup, stopping operation in 1-41
kas commands, described 4-2
kas, entering 4-27
kas, quitting from 4-36
Internet daemon
initializing 10-4
Internet file transfer protocol
initializing daemon 10-2

Job ID number (Backup System)
listing 1-39
using to kill operation 1-41

Kas quit command 4-36
Kernel
incorporating AFS into dynamically 9-28
list of database server machines, changing 3-46
list of database server machines, examining 3-33
Kernel dynamic loader 9-28
Key
deriving from character string 4-51
See also Server encryption key
Key field in Authentication Database
setting octal form directly 4-42
setting using password 4-45
Key version number 2-6
listing from Authentication Database 4-14
listing from KeyFile file 2-36
rules for 2-11
setting in KeyFile file 2-11
KeyFile file
adding server encryption key 2-11
function defined 2-6
listing server encryption keys 2-36
removing server encryption key 2-45
Keyword
using to get help on backup commands 1-14
using to get help on bos commands 2-15
using to get help on fs commands 3-3
using to get help on kas commands 4-7
using to get help on package commands 5-4
using to get help on pts commands 6-9
using to get help on uss commands 7-9
using to get help on vos commands 8-10
using to look up command name xviii

Labeling
tape for use in Backup System 1-43
Learning
AFS UID from user/group name 6-20
home cell of directory/file 3-68
user/group name given AFS UID 6-20
volume ID given volume name 8-22
volume ID number given directory/file name 3-16
volume ID number given volume location 8-33
volume location given directory/file name 3-67
volume location given volume name/ID number 8-30
volume name given directory/file name 3-14, 3-16, 3-34
volume quota given directory/file name 3-16
Leaving
kas interactive mode 4-36
Lifetime
See also Ticket lifetime
Link
See also Hard link, Symbolic link
List-taking argument xi
Listing
ACL 3-30
AFS disk partitions on file server machine 8-28
Authentication Database entry 4-14
CellServDB file (server), contents of 2-34
GroupMax counter in Protection Database 6-25
ListMax counter in Protection Database 6-25
VLDB entry with volume header 8-22
Volume Server status 8-55
all entries in Authentication Database 4-30
client machine's home cell 3-69
creator of Protection Database entry 6-20
data cache amount used 3-22
data cache size 3-22
database server machines from client kernel 3-33
database server machines in CellServDB file (server) 2-34
directory/file location 3-67
disk partition blocks available 3-14, 3-16, 8-42
disk partition blocks used 3-14
disk partition percent used 3-14, 3-34
disk partition size 3-14, 3-16, 8-42
file server machine status 3-4
group-creation quota 6-20
groups machine belongs to 6-29
groups owned by user/group 6-27
groups user belongs to 6-29
home cell of directory/file 3-68
information from VLDB 8-30
information from volume header 8-33
key version number from Authentication Database 4-14
key version number from KeyFile file 2-36
members of group 6-29
mount point 3-36
number of group memberships 6-20
owner of Protection Database entry 6-20
password (octal equivalent) 4-19
privacy flags on Protection Database entry 6-20
privileged users from UserList file 2-39
server encryption key in Authentication Database 4-14
server encryption keys in KeyFile file 2-36
server process run status 2-68
server process' entry in BosConfig file 2-68
setuid status 3-23
system type of client machine 3-65
target machine for Cache Manager monitoring 3-44
tickets held by issuer 4-33
time stamp on binary file 2-24
tokens held by issuer 9-71
volume header with VLDB entry 8-22
volume percent used 3-34

volume quota percent used 3-48
volume quota, with volume & partition info. 3-16
volume quota, with volume size 3-34
volume size 3-16, 3-34
See also Checking, Examining
Local disk
configuring on client, using package 5-2
Lock Status
listing limit from Authentication Database entry 4-14
LOCK access right 3-31, 3-52
Locked VLDB entry
listing information from 8-30
unlocking 8-61, 8-63
Locking
VLDB entry 8-37
Locktime
listing limit from Authentication Database entry 4-14
setting in Authentication Database entry 4-37
Log file
examining on file server machine 2-26
Logging in
AFS version 10-7
-long flag on vos listvol command 8-33
LOOKUP access right 3-31, 3-51

Machine
adding to group 6-7
listing groups belonged to 6-29
Maintaining
synchrony of VLDB with volume headers 8-2
Makefile file for package 5-1
Man command xx
Manual pages (UNIX style) xx
Member
adding to group 6-7
removing from group 6-32
Membership
system-defined groups 6-3
Memory state of BOS Server 2-4
Messages
associated with volume, creating 3-63
associated with volume, examining 3-16
Monitoring
Cache Manager activity 3-44
File Server status with scout 9-66
disk usage with scout 9-66, 9-69
outages with scout 9-66
Mount point
ReadWrite 3-40
cellular 3-40
creating 3-40
creating with uss 7-31
listing 3-36
regular 3-40
removing 3-50
Mounting
local volume in local cell 3-40
Moving
volume 8-39
Mutual authentication 2-6

NFS
export of by AFS client 3-18
obtaining authenticated AFS access from non-supported client of 9-49
NFS/AFS Translator
enabling Cache Manager as 9-6
NOCPW flag in Authentication Database entry 4-39
NOSEAL flag in Authentication Database entry 4-39
NOTGS flag in Authentication Database entry 4-38
NTPD
initializing with runntp 9-61
Name
See also Cell name, User name, Volume name
Needs salvage status flag in volume header 8-3
Negative access rights
setting 3-51
Network File System
See also NFS
Network Time Protocol Daemon
See also NTPD
New release flag in VLDB
as indicator of failed replication 8-44
NoAuth file 2-55
"none" grouping of access rights 3-52

OLD version of binary file
See also .OLD (at beginning of Index)
Off-line status flag in volume header 8-3
Offline message
creating 3-63
examining 3-16
Old release flag in VLDB
as indicator of failed replication 8-44
On-line status flag in volume header 8-3
Online help xviii
See also Help, Keyword
Operating system
AFS 3.0 names for 3-65
See also Kernel
Operation code xi
Outages
BOS Server's role in reducing 9-17
monitoring with afsmonitor 9-9
monitoring with scout 9-66
Overwriting
directories/files/links with uss 7-4
Owner
changing for Protection Database entry 6-10
listing for Protection Database entry 6-20
of Protection Database entry defined 6-2
setting for Protection Database entry 6-12

PAG
creating with AFS login 10-7
creating with pagsh command 9-57
Partition
salvaging volumes on 2-51
See also Disk partition
Password
changing/setting in Authentication Database 9-52
checking quality of 4-45, 9-52
defining initial in Authentication Database 4-8
generating octal form of 4-51
in Authentication Database 4-3, 4-31
listing octal equivalent 4-19
setting in Authentication Database 4-45, 4-49
setting in Authentication Database entry in octal key form 4-42
Password lifetime

listing from Authentication Database entry 4-14
setting in Authentication Database entry 4-37
Password reuse
setting restrictions on in Authentication Database entry 4-37
Plus sign on argument xi
Port offset number
assigning to Tape Coordinator 1-8
listing from Backup Database 1-48
Portoffset argument
to backup commands 1-2
Prefix-less group
See also Group, prefix-less
Privacy flags
listing for Protection Database entry 6-20
setting on Protection Database entry 6-36
Privilege requirements
for backup commands 1-3, 1-4
for bos commands 2-8
for fs commands 3-2
for kas commands 4-6
for package command 5-3
for pts commands 6-5
for uss commands 7-3
for vos commands 8-7
Privileged users
adding to UserList file 2-13
adding to system:administrators group 6-7
listing from UserList file 2-39
removing from UserList file 2-47
removing from system:administrators group 6-32
setting ADMIN flag in Authentication Database 4-37
Process
See also Server process
Process binaries
See also Binary file
Program
controlling setuid status 3-56
listing setuid status 3-23
Protection
AFS vs. UNIX 6-1
Protection Database 6-1
adding member to group 6-7
changing name of entry 6-34
changing owner of entry 6-10
creating entry with uss 7-4
creating group entry 6-12
creating user entry 6-15
creator of entry defined 6-2
deleting entry 6-18
deleting entry with uss 7-15
examining entry 6-20
group-creation quota defined 6-2
information in entry 6-2
listing GroupMax counter 6-25
listing UserMax counter 6-25
listing entry creator 6-20
listing entry owner 6-20
listing group membership 6-20, 6-29
listing groups owned 6-27
listing privacy flags 6-20
membership count defined 6-2
owner of entry defined 6-2
removing member from group 6-32
setting GroupMax counter 6-39
setting UserMax counter 6-39
setting group-creation quota 6-36
setting initial owner of entry 6-12
setting privacy flags 6-36
Protection Server 6-1
listed in CellServDB file 2-5
starting 9-59
See also ptserver process
Protection group
See also Group
Prototype file in package 5-1

Quitting
kas interactive mode 4-36
Quota
See also Group-creation quota, Volume quota

Random generation
encryption key 4-21
READ access right 3-31, 3-51
"read" grouping of access rights 3-52
ReadOnly ID number in volume header 8-3
ReadOnly volume
ID number in VLDB 8-2
changing name of 8-51
creating 8-44
defining site in VLDB 8-8
dumping 8-20
forcing Cache Manager to see new release 3-7
moving 8-39
need for all-or-nothing release 8-44
removing 8-47
removing mistakenly defined site 8-49
ReadWrite ID number in volume header 8-3
ReadWrite volume
ID number in VLDB 8-2
changing name of 8-51
cloning for Backup version 8-11
cloning for replication 8-44
cloning multiple for Backup version 8-12
creating 8-15
dumping 8-20
moving 8-39
removing 8-47
Readonly extension on volume name
See also .readonly extension on volume name
Regular expression
use in Backup System 1-10
Regular group
See also Group, regular
Release
flags in VLDB 8-2
ReleaseClone 8-44
ReleaseClone number in volume header 8-4
Releasing ReadOnly volume
forcing Cache Manager to see 3-7
need for all-or-nothing 8-44
Remote
command execution with bos exec 2-23
file copy with rcp command (AFS version) 10-11
login server with rlogind command (AFS version) 10-13
shell with rsh command (AFS version) 10-15
Removing
.BAK version of binary file 2-41

.OLD version of binary file 2-41
ACL entry 3-51
core file from /usr/afs/logs 2-41
database server machine from CellServDB file (server) 2-43
mount point 3-50
obsolete AFS UIDs from ACL 3-8
privileged users from UserList file 2-47
server encryption key from KeyFile file 2-45
user account with uss 7-15
user from group 6-32
volume 8-47
volume from site, without changing VLDB 8-65
volume site mistakenly defined 8-49
See also Deleting
Renaming
volume 8-51
Replacing binary file
See also Installing, Uninstalling
Replication 8-44
determining success of 8-44
forcing creation of new clone 8-44
need for all-or-nothing release 8-44
role of ReleaseClone in 8-44
Restart times
examining 2-28
setting 2-59
Restarting
server process 2-49
Restoring
internal consistency in file system (salvaging) 2-51
synchrony of VLDB and volume headers 8-57, 8-59
volumes, not using Backup System 8-53

Salvager
as part of fs process 2-4
Salvaging volume(s) 2-51, 9-63
Scout
disk usage 9-69
statistics available 9-68
Server
See also File server machine, Server process
Server encryption key 2-6
adding to KeyFile file 2-11
in Authentication Database 4-3, 4-31
listing from Authentication Database 4-14
listing from KeyFile file 2-36
removing from KeyFile file 2-45
Server portion of Update Server
See also Upserver process
Server process 2-1
controlling status via BosConfig file 2-5
creating in BosConfig file 2-16
creating ticket (tokens) for 9-43
creating ticket for 4-23
cron type 2-4
deleting from BosConfig file 2-21
examining BosConfig entry 2-68
examining log file for 2-26
fs type 2-3
installing binary file 2-32
listing run status 2-68
listing time stamp on binary file 2-24
restarting 2-49
simple type 2-3
starting 2-66
starting (changing status flag in BosConfig) 2-64
starting on server 2-16
stopping (changing BosConfig status flag) 2-72
stopping (no change to BosConfig status flag) 2-62
stopping and immediately restarting 2-49
uninstalling binary file 2-74
Server ticket
See also Tickets
Setting
ACL 3-51
ACL with uss 7-21, 7-31
AFS UID for group 6-12
AFS UID for user 6-15
Authentication Database entry flags and expiration dates 4-37
BOS Server's restart times 2-59
GroupMax counter in Protection Database 6-39
ListMax counter in Protection Database 6-39
authorization checking requirements on file server machine 2-55
cell name in CellServDB file 2-57
cell name in ThisCell file 2-57
data cache size 3-54
default ACL at volume creation 8-15
default volume quota at volume creation 8-15
group-creation quota 6-36
initial owner for group 6-12
key field in Authentication Database, directly in octal form 4-42
key field in Authentication Database, using password 4-45
key version number in KeyFile file 2-11
password in Authentication Database 9-52
privacy flags on Protection Database entry 6-36
setuid status 3-56
status flag in BOS Server memory to NotRun 2-72
status flag in BOS Server memory to Run 2-64, 2-66
status flag in BosConfig file 2-16
status flag in BosConfig file to NotRun 2-72
status flag in BosConfig file to Run 2-64
system type of client machine 3-65
target machine for Cache Manager monitoring 3-44
volume quota 3-58, 3-63
volume quota with uss 7-31
See also Defining
Setuid privilege
controlling 3-56
listing 3-23
Shell
opening remotely with rsh command (AFS version) 10-15
"Shorthand" groupings of access rights 3-52
Shutting down
server process 2-62
Simple-type process 2-3
creating 2-17
Site
count in VLDB 8-2
for volume, defined 8-2
Size
of disk partition, listing with space available 8-42
of end-of-file mark for tape drive, determining 9-36
of tape in Backup System, defining 1-43
of tape in Backup System, listing 1-53

See also Data cache
Size quota
See also Volume quota
Socket
defining with package 5-15
Starting
Backup Database Server 9-22
NTPD, with runntp 9-61
Protection Server 9-59
Salvager by hand 9-63
Tape Coordinator 9-25
client portion of Update Server 9-78
process on server (changing status flag in BosConfig) 2-64
process on server (no change to status flag in BosConfig) 2-66
server portion of Update Server 9-81
server process 2-16
Statistics
for Authentication Server 4-49
Statistics available in scout 9-68
Status
listing for Volume Server 8-55
listing for file server machine 3-4
listing for server process 2-68
Status flag (in BOS Server memory) 2-4
Status flag (in BosConfig file) 2-3
setting to Run 2-16
Status flag in volume header 8-3
Stopping
server process (changing status flag in BosConfig) 2-72
server process (no change to BosConfig status flag) 2-62
Storing statistic in Scout 9-69
Switch
defined xi
omitting xiii
Symbolic link
creating with uss 7-30
defining with package 5-13
overwriting with uss 7-4
Synchrony of VLDB and volume headers
maintained by VL and Volume Servers 8-2
restoring 8-57, 8-59
@sys variable in pathnames 3-65
Sysname 3-65
System groups
list of 6-3
on ACLs 6-3
System outages, reducing 9-17
System type
AFS 3.0 names for 3-65
of client machine 3-65

Tape Coordinator
assigning port offset number to 1-8
defining in Backup Database 1-8
determining end-of-file mark size for 9-36
listing port offset number from Backup Database 1-48
listing status 1-65
removing entry from Backup Database 1-20
starting 9-25
Tape in Backup System
defining capacity on label 1-43
determining capacity of 9-36
labeling 1-43
listing capacity of 1-53
listing label on 1-53
Template file in uss
See also Uss template file
Terminal type
setting for AFSMonitor 9-11
Testing
package files 9-56
ThisCell file (client) 3-69
ThisCell file (server) 2-57
Ticket lifetime
listing from Authentication Database 4-14
setting in Authentication Database 4-37
Tickets 2-6
creating for server process 4-23, 9-43
discarding 4-18, 9-74
listing for issuer of command 4-33
listing for user 9-71
Time stamp
listing for binary file 2-24
Tokens
creating for server process 9-43
discarding 9-74
listing for user 9-71
Trace
for Authentication Server debugging 4-10
for Cache Manager debugging 3-12
Tracking
Cache Manager activity 3-44
Translating
AFS UID to user/group name 6-20
directory/file name to volume ID number 3-16
directory/file name to volume location 3-67
directory/file name to volume name 3-14, 3-16, 3-34
directory/file name to volume quota 3-16, 3-34
directory/file name to volume quota percent used 3-48
user/group name to AFS UID 6-20
volume location to volume ID number 8-33
volume name to volume ID number 8-22
volume name/ID number to volume location 8-30
Trouble-shooting
Cache Manager debugging trace 3-12
Troubleshooting
Authentication Server debugging trace 4-10
Type
flags in VLDB 8-2
Types of AFS server processes 2-3

UID
See also AFS UID, UNIX UID
UNIX
manual pages for AFS command suites xx
UNIX UID
functional difference from AFS UID 6-2
UNIX commands
ftpd (AFS version) 10-2
inetd (AFS version) 10-4
login (AFS version) 10-7
rcp (AFS version) 10-11
rlogind (AFS version) 10-13
rsh (AFS version) 10-15
Unauthenticating
while in kas interactive mode 4-35
%undef in package conf. file 5-20

Uninstalling
  binary file 2-74
Unlocking
  VLDB entry 8-61, 8-63
Unmounting volume 3-50
Update Server
  starting client portion 9-78
  starting server portion 9-81
Update date
  recorded in volume header 8-4
Updating
  Cache Manager mapping of volume names to IDs 3-7
User
  account (see entry: *User account*) 7-4
  adding to group 6-7
  changing name in Protection Database 6-34
  creating in Protection Database 6-15
  deleting from Protection Database 6-18
  listing groups belonged to 6-29
  listing groups owned 6-27
  listing number of group memberships 6-20
  mapping name to AFS UID 6-20
  removing from group 6-32
  restrictions on name format in Protection Database 6-15
  setting AFS UID 6-15
  setting group-creation quota 6-36
  setting privacy flags 6-36
User account
  creating accounts for many at once 7-10
  creating with uss 7-4
  deleting with uss 7-15
User name
  assigning in uss 7-4
UserList file
  adding user to 2-13
  function defined 2-6
  listing users in 2-39
  removing users from 2-47
UserMax counter in Protection Database
  affected by setting AFS UID 6-15
  listing 6-25
  setting 6-39

VL Server 8-1
  importance to transparent access 9-87
  listed in CellServDB file 2-5
  See also vlserver process
VLDB 8-1
  AFS 2.0 version of 3-23
  converting from 3.1 to 3.2 9-83
  defining ReadOnly site in 8-8
  deleting entry (but not volume header) 8-17
  entry created by vos create 8-15
  examining entry, incl. locked one 8-30
  examining entry, with volume header 8-22
  information in 8-2
  locking entry 8-37
  release flags in 8-2
  removing mistakenly defined ReadOnly site 8-49
  site count in 8-2
  synchronizing with volume headers 8-57, 8-59
  type flags in 8-2
  unlocking entry 8-61, 8-63
Variables
  @sys in pathnames 3-65
Volume
  Backup (see entry: *Backup volume*) 8-11
  Cache Manager's mapping of names to IDs, updating 3-7
  ID number (see entry: *Volume ID number*) 8-2
  ReadOnly (see entry: *ReadOnly volume*) 8-44
  ReadWrite (see entry: *ReadWrite volume*) 8-15
  creating with uss 7-4, 7-31
  creation date in volume header 8-4
  deleting with uss 7-15
  dumping using AFS Backup System 1-27
  dumping without Backup System 8-20
  flushing from data cache 3-21
  header (see entry: *Volume header*) 8-3
  host partition size, listing 3-16
  information about in VLDB 8-2
  listing dump history of 1-67
  listing mount point 3-36
  messages associated with, creating 3-63
  messages associated with, examining 3-16
  mounting 3-40
  moving 8-39
  name (see entry: *Volume name*) 8-15
  number of accesses in day 8-4
  partition percent use, listing 3-34
  percent use, listing 3-34
  removing 8-47
  removing from site, without changing VLDB 8-65
  removing mistakenly defined ReadOnly site 8-49
  removing mount point for 3-50
  renaming 8-51
  replicating 8-44
  restoring (with Backup System) 1-69
  restoring without Backup System 8-53
  salvaging 2-51
  setting warning message 3-63
  size, listing 3-16, 3-34
  synchronizing VLDB and volume header for 8-57, 8-59
  update date in volume header 8-4
  volume ID number (see entry: *Volume ID number*) 8-2
Volume ID number
  Cache Manager's mapping to volume name, updating 3-7
  allocated by vos create 8-15
  defined 8-2
  learning from volume location 8-33
  learning from volume name 8-22
  learning given directory/file name 3-16
  translating to volume location 8-30
Volume Location Database
  See also VLDB
Volume Location Server
  starting 9-87
  See also VL Server
Volume Server 8-1
  as part of fs process 2-4
  listing status 8-55
  starting 9-89
Volume entry (Backup System)
  adding to volume set 1-10
  listing for volume set 1-50
  removing from volume set 1-22
Volume header 8-3
  examining 8-33

examining, with VLDB entry 8-22
synchronizing with VLDB 8-57, 8-59
Volume location
learning from volume name/ID number 8-30
learning given directory/file name 3-67
translating to volume ID number 8-33
Volume name
Cache Manager forced to see change to 3-7
Cache Manager's mapping to volume ID, updating 3-7
changing 8-51
learning given directory/file name 3-14, 3-16, 3-34
translating to volume ID number 8-22
translating to volume location 8-30
Volume quota
allowing users to exceed 9-32
as recorded in volume header 8-4
default set by vos create 8-15
listing percent used 3-48
listing with volume & partition info. 3-16
listing with volume size 3-34
setting 3-58, 3-63
setting with uss 7-31
Volume set (Backup System)
adding volume entry to 1-10
creating 1-13
listing 1-50
removing 1-23
removing volume entry from 1-22

Weekly restart of BOS Server
examining time for 2-28
setting time for 2-59
Wild card
in volume entry definition (Backup System) 1-10
Workstation
See also Client machine
Workstations statistic in scout 9-69
WRITE access right 3-31, 3-51
"write" grouping of access rights 3-52

admin argument
on all uss commands 7-2
admin password argument
on all kas commands 4-5
admin username argument
on all kas commands 4-4
afsd command 9-2
configuration scripts 9-5
afsmonitor
command syntax for 9-9
outages, monitoring 9-9

backup adddump command 1-5
backup addhost command 1-8
backup addvolentry command 1-10
backup addvolset command 1-13
backup apropos command 1-14
backup dbverify command 1-15
backup deldump command 1-17
backup deletedump command 1-18
backup delhost command 1-20
backup delvolentry command 1-22
backup delvolset command 1-23
backup diskrestore command 1-24
backup dump command 1-27
backup dumpinfo command 1-33
backup help command 1-37
backup jobs command 1-39
backup kill command 1-41
backup labeltape command 1-43
backup listdumps command 1-46
backup listhosts command 1-48
backup listvolsets command 1-50
backup quit command 1-52
backup readlabel command 1-53
backup restoredb command 1-55
backup savedb command 1-57
backup scantape command 1-59
backup setexp command 1-63
backup status command 1-65
backup volinfo command 1-67
backup volrestore command 1-69
bos addhost command 2-9
bos addkey command 2-11
bos adduser command 2-13
bos apropos command 2-15
bos commands
common arguments 2-7
privilege requirements 2-8
bos create command 2-16
bos delete command 2-21
bos exec command 2-23
bos getcell command
See also bos listhosts command
bos getdate command 2-24
bos getlog command 2-26
bos getrestart command 2-28
bos help command 2-30
bos install command 2-32
bos listhosts command 2-34
bos listkeys command 2-36
bos listusers command 2-39
bos prune command 2-41
bos removehost command 2-43
bos removekey command 2-45
bos removeuser command 2-47
bos restart command 2-49
bos salvage command 2-51
bos setauth command 2-55
bos setcellname command 2-57
bos setrestart command 2-59
bos shutdown command 2-62
bos start command 2-64
bos startup command 2-66
bos status command 2-68
bos stop command 2-72
bos uninstall command 2-74
bosserver command 9-17
budb_convert command 9-19
buserver command 9-22
buserver process
creating 2-16
butc command 9-25

cell argument
incompatibility with localauth flag in bos command 2-7
incompatibility with localauth flag in vos command 8-5
on all bos commands 2-7
on all kas commands 4-5
on all pts commands 6-4

on all uss commands 7-2
on all vos commands 8-5
creating
buserver process 2-16
fs process 2-16
kaserver process 2-16
ptserver process 2-16
runntp process 2-16
upclient process 2-16
upserver process 2-16
vlserver process 2-16
cron-type process
creating 2-17

displaying
Cache Manager preferences for file server machines 3-25
dkload command 9-28
dryrun flag
on all uss commands 7-2

file server machine
displaying Cache Manager preferences 3-25
setting Cache Manager preferences 3-59
fileserver command 9-31
fms command 9-36
force flag
on all pts commands 6-5
fs apropos command 3-3
fs checkservers command 3-4
fs checkvolumes command 3-7
fs cleanacl command 3-8
fs commands
common arguments 3-2
privilege requirements 3-2
fs copyacl command 3-10
fs debug command 3-12
fs diskfree command 3-14
fs examine command 3-16
fs exportafs command 3-18
fs flush command 3-20
fs flushvolume command 3-21
fs getcacheparms command 3-22
fs getcellstatus command 3-23
fs getserverprefs command 3-25
fs help command 3-28
fs listacl command 3-30
fs listcells command 3-33
fs listquota command 3-34
fs lsmount command 3-36
fs messages command 3-38
fs mkmount command 3-40
fs monitor command 3-44
fs newcell command 3-46
fs process
creating 2-16
fs quota command 3-48
fs rmmount command 3-50
fs setacl command 3-51
fs setcachesize command 3-54
fs setcell command 3-56
fs setquota command 3-58
fs setserverprefs command 3-59
fs setvol command 3-63
fs sysname command 3-65
fs whereis command 3-67
fs whichcell command 3-68
fs wscell command 3-69
fs-type process 2-3
creating 2-17
ftpd command (AFS version) 10-2

help flag
on all bos commands 2-7
on all fs commands 3-2
on all kas commands 4-6
on all pts commands 6-5
on all uss commands 7-2
on all vos commands 8-6
to backup commands 1-3

inetd command (AFS version) 10-4

kas apropos command 4-7
kas commands
common arguments 4-4
interactive mode 4-2
privilege requirements 4-6
kas create command 4-8
kas debuginfo command 4-10
kas delete command 4-12
kas examine command 4-14
kas forgetticket command 4-18
kas getpassword command 4-19
kas getrandomkey command 4-21
kas getticket command 4-23
kas help command 4-25
kas interactive command 4-27
kas list command 4-30
kas listtickets command 4-33
(kas) noauthentication command 4-35
kas noauthentication command 4-35
(kas) quit command 4-36
kas setfields command 4-37
kas setkey command 4-42
kas setpassword command 4-45
kas statistics command 4-49
kas stringtokey command 4-51
kas unlock command 4-53
kaserver command 9-39
kaserver process
creating 2-16
See also Authentication Server
kdb command 9-41
klog command 9-43
knfs command 9-49
kpasswd command 9-52

localauth flag
incompatibility with cell argument in bos command 2-7
incompatibility with cell argument in vos command 8-5
on all vos commands 8-5
on bos commands 2-7
login command (AFS version) 10-7

noauth flag
on all bos commands 2-7
on all pts commands 6-4
on all vos commands 8-5
on kas commands 4-5

package
%define in configuration file 5-16
%ifdef in configuration file 5-17
%ifndef in configuration file 5-18
%include in configuration file 5-19
%undef in configuration file 5-20
B line in configuration file 5-7
C line in configuration file 5-8
D line in configuration file 5-9
F line in configuration file 5-10
L line in configuration file 5-13
Makefile 5-1
S line in configuration file 5-15
command, syntax defined 5-2
configuration file 5-1
defining block special device 5-7
defining character special device 5-8
defining directory 5-9
defining file 5-10
defining socket 5-15
defining symbolic link 5-13
privilege requirements 5-3
prototype file 5-1
package apropos command 5-4
package help command 5-5
package_test command 9-56
pagsh command 9-57
preferences 3-25, 3-59
displaying 3-25
setting 3-59
pts adduser command 6-7
pts apropos command 6-9
pts chown command 6-10
pts commands
common arguments 6-4
privilege requirements 6-5
pts creategroup command 6-12
pts createuser command 6-15
pts delete command 6-18
pts examine command 6-20
pts help command 6-23
pts listmax command 6-25
pts listowned command 6-27
pts membership command 6-29
pts removeuser command 6-32
pts rename command 6-34
pts setfields command 6-36
pts setmax command 6-39
ptserver command 9-59
ptserver process
creating 2-16
See also Protection Server

rcp command (AFS version) 10-11
rlogind command (AFS version) 10-13
rsh command (AFS version) 10-15
runntp command 9-61
runntp process
creating 2-16

salvager
command for invoking manually 9-63
salvager command 9-63
scout
command syntax for 9-66
outages, monitoring 9-66
servers argument
on all kas commands 4-5
setting
Cache Manager preferences for file server machines 3-59
status flag (in BOS Server memory)
setting to NotRun 2-62, 2-72
setting to Run 2-64, 2-66
status flag (in BosConfig file)
setting to NotRun 2-72
setting to Run 2-64
status flag
in Authentication Database, listing 4-14
in Authentication Database, setting 4-37
system:administrators group 6-3
system:anyuser group 6-3
system:authuser group 6-3

test flag
on all pts commands 6-5
tokens
command 9-71
tokens command 9-71
translate_et command 9-73

unlog command 9-74
up command 9-76
upclient command 9-78
upclient process
creating 2-16
See also upclient command
upserver command 9-81
upserver process
creating 2-16
See also upserver command
uss
creating directory 7-21
creating directory for even distribution 7-27
creating file by echoing 7-23
creating file from prototype 7-25
creating hard link 7-29
creating symbolic link 7-30
creating volume 7-31
executing command from template 7-33
improving password/login security 7-19
mounting volume 7-31
overwriting behavior 7-4
setting ACL 7-21, 7-31
setting volume quota 7-31
template (see entry: *Uss template file*) 7-1
uss add command 7-4
uss apropos command 7-9
uss bulk command 7-10
uss commands
common arguments 7-2
privilege requirements 7-3
uss delete command 7-15
uss help command 7-17
uss template file
A line 7-19
D line 7-21
E line in 7-23
F line in 7-25
G line in 7-27

L line in 7-29
S line in 7-30
V line in 7-31
X line in 7-33
creating directory 7-21
creating directory for even distribution in 7-27
creating file in 7-23, 7-25
creating hard link in 7-29
creating mount point 7-31
creating symbolic link in 7-30
creating volume 7-31
executing command from 7-33
improving password/login security 7-19
setting ACL 7-21, 7-31
setting volume quota 7-31
zero length 7-31

verbose flag
on all vos commands 8-6
vldb_convert command 9-83
vlserver command 9-87
vlserver process
creating 2-16
See also VL Server
volserver command 9-89
vos addsite command 8-8
vos apropos command 8-10
vos backup command 8-11
vos backupsys command 8-12
vos commands
common arguments 8-5
idempotency of 8-1
privilege requirements 8-7
vos create command 8-15
vos delentry command 8-17
vos dump command 8-20
vos examine command 8-22
vos help command 8-26
vos listpart command 8-28
vos listvldb command 8-30
vos listvol command 8-33
vos lock command 8-37
vos move command 8-39
vos partinfo command 8-42
vos release command 8-44
vos remove command 8-47
vos remsite command 8-49
vos rename command 8-51
vos restore command 8-53
vos status command 8-55
vos syncserv command 8-57
vos syncvldb command 8-59
vos unlock command 8-61
vos unlockvldb command 8-63
vos volinfo command
See also vos examine command
vos zap command 8-65

xstat
a requirement for running AFSMonitor 9-11

Table of Contents

About This Manual vii
- Audience and Purpose vii
- Organization vii
- How to Use This Document viii
- Related Documents ix
- Document Conventions x
- Command Syntax xi
- Rules for Typing AFS Commands xiii
- Rules for Using Abbreviations and Aliases xv
- Getting Online Help on AFS Commands xviii
- Ordering, Support, and Training Information ..... -21

1. The backup Commands 1-1
- 1.1. The Information in Backup Database Entries 1-1
  - 1.1.1. The Backup Database 1-1
- 1.2. Common Arguments and Flags 1-2
- 1.3. The Privilege Required for backup Commands 1-3
- 1.4. Running Backup Commands in a Foreign Cell 1-4
- backup adddump 1-5
- backup addhost 1-8
- backup addvolentry 1-10
- backup addvolset 1-13
- backup apropos 1-14
- backup dbverify 1-15
- backup deldump 1-17
- backup deletedump 1-18
- backup delhost 1-20
- backup delvolentry 1-22
- backup delvolset 1-23
- backup diskrestore 1-24
- backup dump 1-27
- backup dumpinfo 1-33
- backup help 1-37
- backup jobs 1-39
- backup kill 1-41
- backup labeltape 1-43
- backup listdumps 1-46
- backup listhosts 1-48
- backup listvolsets 1-50
- backup quit 1-52
- backup readlabel 1-53
- backup restoredb 1-55
- backup savedb 1-57
- backup scantape 1-59
- backup setexp 1-63
- backup status 1-65
- backup volinfo 1-67
- backup volrestore 1-69

2. The bos Commands **2-1**
2.1. Configuration Files Used by the bos Commands **2-2**
2.1.1. The *BosConfig* File **2-2**
2.1.2. The *CellServDB* File **2-5**
2.1.3. The *KeyFile* File **2-6**
2.1.4. The *UserList* File **2-6**
2.2. Common Arguments and Flags **2-7**
2.3. The Privilege Required for bos Commands **2-8**
bos addhost **2-9**
bos addkey **2-11**
bos adduser **2-13**
bos apropos **2-15**
bos create **2-16**
bos delete **2-21**
bos exec **2-23**
bos getdate **2-24**
bos getlog **2-26**
bos getrestart **2-28**
bos help **2-30**
bos install **2-32**
bos listhosts **2-34**
bos listkeys **2-36**
bos listusers **2-39**
bos prune **2-41**
bos removehost **2-43**
bos removekey **2-45**
bos removeuser **2-47**
bos restart **2-49**
bos salvage **2-51**
bos setauth **2-55**
bos setcellname **2-57**
bos setrestart **2-59**
bos shutdown **2-62**
bos start **2-64**
bos startup **2-66**
bos status **2-68**
bos stop **2-72**
bos uninstall **2-74**
3. The fs Commands **3-1**
3.1. Common Arguments and Flags **3-2**
3.2. The Privileges Required for fs Commands **3-2**
fs apropos **3-3**
fs checkservers **3-4**
fs checkvolumes **3-7**
fs cleanacl **3-8**
fs copyacl **3-10**
fs debug **3-12**
fs diskfree **3-14**
fs examine **3-16**
fs exportafs **3-18**
fs flush **3-20**
fs flushvolume **3-21**

fs getcacheparms **3-22**
fs getcellstatus **3-23**
fs getserverprefs **3-25**
fs help **3-28**
fs listacl **3-30**
fs listcells **3-33**
fs listquota **3-34**
fs lsmount **3-36**
fs messages **3-38**
fs mkmount **3-40**
fs monitor **3-44**
fs newcell **3-46**
fs quota **3-48**
fs rmmount **3-50**
fs setacl **3-51**
fs setcachesize **3-54**
fs setcell **3-56**
fs setquota **3-58**
fs setserverprefs **3-59**
fs setvol **3-63**
fs sysname **3-65**
fs whereis **3-67**
fs whichcell **3-68**
fs wscell **3-69**

4. The kas Commands **4-1**

4.1. The kas Interface **4-2**
4.1.1. Interactive Mode **4-2**
4.2. Information in the Authentication Database **4-3**
4.3. The Auxiliary Database **4-4**
4.4. Common Arguments and Flags **4-4**
4.5. The Privilege Required for kas Commands **4-6**
kas apropos **4-7**
kas create **4-8**
kas debuginfo **4-10**
kas delete **4-12**
kas examine **4-14**
kas forgetticket **4-18**
kas getpassword **4-19**
kas getrandomkey **4-21**
kas getticket **4-23**
kas help **4-25**
kas interactive **4-27**
kas list **4-30**
kas listtickets **4-33**
kas noauthentication **4-35**
kas quit **4-36**
kas setfields **4-37**
kas setkey **4-42**
kas setpassword **4-45**
kas statistics **4-49**
kas stringtokey **4-51**
kas unlock **4-53**

5. The package Command and Configuration File Lines 5-1
package 5-2
package apropos 5-4
package help 5-5
Configuration File "B" Line 5-7
Configuration File "C" Line 5-8
Configuration File "D" Line 5-9
Configuration File "F" Line 5-10
Configuration File "L" Line 5-13
Configuration File "S" Line 5-15
Configuration File %define 5-16
Configuration File %ifdef 5-17
Configuration File %ifndef 5-18
Configuration File %include 5-19
Configuration File %undef 5-20
6. The pts Commands 6-1
6.1. The Information in Protection Database Entries 6-2
6.2. The Standard System Groups 6-3
6.3. Common Arguments and Flags 6-4
6.4. Privileges Required for pts Commands 6-5
pts adduser 6-7
pts apropos 6-9
pts chown 6-10
pts creategroup 6-12
pts createuser 6-15
pts delete 6-18
pts examine 6-20
pts help 6-23
pts listmax 6-25
pts listowned 6-27
pts membership 6-29
pts removeuser 6-32
pts rename 6-34
pts setfields 6-36
pts setmax 6-39
7. The uss Commands and Template Lines 7-1
7.1. Common Arguments and Flags 7-2
7.2. The Privilege Required for uss Commands 7-3
uss add 7-4
uss apropos 7-9
uss bulk 7-10
uss delete 7-15
uss help 7-17
Template "A" line 7-19
Template "D" line 7-21
Template "E" line 7-23
Template "F" line 7-25
Template "G" line 7-27
Template "L" line 7-29
Template "S" line 7-30
Template "V" line 7-31

Template "X" line 7-33
8. The vos Commands 8-1
8.1. The Information in VLDB Entries 8-2
8.2. The Information in Volume Headers 8-3
8.3. Common Arguments and Flags 8-5
8.4. The Privilege Required for vos Commands 8-7
vos addsite 8-8
vos apropos 8-10
vos backup 8-11
vos backupsys 8-12
vos create 8-15
vos delentry 8-17
vos dump 8-20
vos examine 8-22
vos help 8-26
vos listpart 8-28
vos listvldb 8-30
vos listvol 8-33
vos lock 8-37
vos move 8-39
vos partinfo 8-42
vos release 8-44
vos remove 8-47
vos remsite 8-49
vos rename 8-51
vos restore 8-53
vos status 8-55
vos syncserv 8-57
vos syncvldb 8-59
vos unlock 8-61
vos unlockvldb 8-63
vos zap 8-65
9. Miscellaneous AFS Commands 9-1
afsd 9-2
afsmonitor 9-9
bosserver 9-17
budb_convert 9-19
buserver 9-22
butc 9-25
dkload 9-28
fileserver 9-31
fms 9-36
kaserver 9-39
kdb . 9-41
klog 9-43
knfs 9-49
kpasswd 9-52
package_test 9-56
pagsh 9-57
ptserver 9-59
runntp 9-61

salvager 9-63
scout 9-66
tokens 9-71
translate_et 9-73
unlog 9-74
up . . 9-76
upclient 9-78
upserver 9-81
vldb_convert 9-83
vlserver 9-87
volserver 9-89

10. Modified UNIX Commands 10-1

ftpd (AFS version) 10-2
inetd (AFS version) 10-4
login (AFS version) 10-7
rcp (AFS version) 10-11
rlogind (AFS version) 10-13
rsh (AFS version) 10-15

11. Command Summary 11-1

11.1. Summary of backup Commands 11-2
11.2. Summary of bos Commands 11-4
11.3. Summary of fs Commands 11-6
11.4. Summary of kas Commands 11-8
11.5. Summary of package Command and Template Lines 11-10
11.6. Summary of pts Commands 11-11
11.7. Summary of uss Commands and Template Lines 11-12
11.8. Summary of vos Commands 11-13
11.9. Summary of AFS Miscellaneous Commands 11-15
11.10. Summary of Modified UNIX Commands 11-18

Index Index-1