AFS Administration Reference

Version 3.6

AFS Administration Reference: Version 3.6

This edition applies to:
IBM AFS for AIX, Version 3.6
IBM AFS for Digital Unix, Version 3.6
IBM AFS for HP-UX, Version 3.6
IBM AFS for Linux, Version 3.6
IBM AFS for SGI IRIX, Version 3.6
IBM AFS for Solaris, Version 3.6

and to all subsequent releases and modifications until otherwise indicated in new editions.This softcopy version is based on the printed edition of this book. Some formatting amendments have been made to make this information more suitable for softcopy.

Revision History

Revision 3.6 April 2000
First Edition

Table of Contents

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(R) 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 *IBM AFS Administration Guide*.

Organization

This document presents AFS files and commands in separate sections, with the files or commands in alphabetical order.

The following sections of each reference page provide the indicated type of information:

- Purpose briefly describes the command's function.
- Synopsis displays the complete syntax statement for a command, which specifies the required order for all options, using the same notation as the AFS online help. If abbreviating the command name a nd option names is acceptable, as it is for most commands, a second statement specifies the shortest acceptable abbreviation of each name. If the command has an alias, it also appears in this section.
- Description describes the file or command's function in detail.
- Cautions describes restrictions, requirements, and potential complications in use of the command. It appears only when necessary.
- Options describes the function and required form of each argument and flag.
- Output describes any output the command writes to the standard output stream. This section does not appear if the command does not produce output or if the only output is a message confirming the command's success.
- Examples provides one or more sample commands and resulting output.
- Privilege Required lists each privilege required to perform the command.
- Related Information lists related commands and files, if any.

How to Use This Document

Refer to this document when you need detailed information about a specific command. For a description of all the steps in a procedure, refer to the *IBM AFS Administration Guide*.

Related Documents

The following documents are included in the AFS documentation set.

IBM AFS Administration Guide

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

The first chapters of the *IBM AFS Administration Guide* present basic concepts and guidelines. Understanding them is crucial to successful administration of an AFS cell. The remaining chapters in the guide provide step-by-step instructions for specific administrative tasks, along with discussions of the concepts important to that particular task.

IBM AFS Quick Beginnings

This guide provides instructions for installing AFS server and client machines. It is assumed that the installer is an experienced UNIX(R) system administrator.

For predictable performance, machines must be installed and configured in accordance with the instructions in this guide.

IBM AFS Release Notes

This document provides information specific to each release of AFS, such as a list of new features and commands, a list of requirements and limitations, and instructions for upgrading server and client machines.

IBM AFS User 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 authenticating, changing a password, protecting AFS data, creating groups, and troubleshooting. It provides illustrative examples for each function and describes some of the differences between the UNIX file system and AFS.

Typographical Conventions

This document uses the following typographical conventions:

- Command and option names appear in **bold type** in syntax definitions, examples, and running text. Names of directories, files, machines, partitions, volumes, and users also appear in **bold type**.
- 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. New terms also appear in *italic type*.
- Examples of screen output and file contents appear in `monospace type`.

In addition, the following symbols appear in command syntax definitions, both in the documentation and in AFS online help statements. When issuing a command, do not type these symbols.

- Square brackets [ ] surround optional items.
- Angle brackets < > surround user-supplied values in AFS commands.
- A superscripted plus sign + follows an argument that accepts more than one value.
- The percent sign % represents the regular command shell prompt. Some operating systems possibly use a different character for this prompt.
- The number sign # represents the command shell prompt for the local superuser root. Some operating systems possibly use a different character for this prompt.
- The pipe symbol | in a command syntax statement separates mutually exclusive values for an argument.

I. AFS Commands

afs

Name

`afs` — Introduction to AFS commands

Description

AFS provides many commands that enable users and system administrators to use and customize its features. Many of the commands belong to the following categories, called *command suites*.

backup

Interface for configuring and operating the AFS Backup System.

bos

Interface to the Basic Overseer (BOS) Server for administering server processes and configuration files.

fs

Interface for administering access control lists (ACLs), the Cache Manager, and other miscellaneous file system functions.

fstrace

Interface for tracing Cache Manager operations when debugging problems.

kas

Interface to the Authentication Server for administering security and authentication information.

pts

Interface to the Protection Server for administering AFS ID and group membership information.

uss

Interface for automated administration of user accounts.

vos

Interface to the Volume Server and Volume Location (VL) Server for administering volumes.

In addition, there are several commands that do not belong to suites.

AFS Command Syntax

AFS commands that belong to suites have the following structure:

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

Command Names

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, and indicates which command interpreter and server process perform the command. AFS has several command suites, including bos, fs, kas, package, pts, uss and vos. Some of these suites have an interactive mode in which the issuer omits the *operation_code* portion of the command name.

The *operation_code* tells the command interpreter and server process which action to perform. Most command suites include several operation codes. The man pages for each command name describe each operation code in detail, and the *IBM AFS Administration Guide* describes how to use them in the context of performing administrative tasks.

Several AFS commands do not belong to a suite and so their names do not have a *command_suite* portion. Their structure is otherwise similar to the commands in the suites.

Options

The term *option* refers to both arguments and flags, which are described in the following sections.

Arguments

One or more arguments can follow the command name. Arguments specify the entities on which to act while performing the command (for example, which server machine, server process, or file). To minimize the potential for error, provide a command's arguments in the order prescribed in its syntax definition.

Each argument has two parts, which appear in the indicated 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 server machine. Switches can often be omitted, subject to the rules outlined in Conditions for Omitting Switches.
- The *value* names a particular entity of the type specified by the preceding switch. For example, the proper value for a -server switch is a server machine name like `fs3.abc.com`. Unlike switches (which have a required form), values vary depending on what the issuer wants to accomplish. Values appear surrounded by angle brackets (<>) in command descriptions and the online help to show that they are user-supplied variable information.

Some arguments accept multiple values, as indicated by trailing plus sign (+) in the command descriptions and online help. How many of a command's arguments take multiple values, and their ordering with respect to other arguments, determine when it is acceptable to omit switches. See Conditions for Omitting Switches.

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

Flags

Some commands have one or more flags, which specify the manner in which the command interpreter and server process perform the command, or what kind of output it produces. Flags are preceded by hyphens like switches, but they take no values. Although the command descriptions and online help generally list a command's flags after its arguments, there is no prescribed order for flags. They can appear anywhere on the command line following the operation code, except in between the parts of an argument. Flags are always optional.

An Example Command

The following example illustrates the different parts of a command that belongs to an AFS command suite.

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

where

- 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 server machine (in this case `fs1.abc.com`) to report the modification dates of binary files in the local */usr/afs/bin* directory.
- `-server fs1.abc.com` is one argument, with -server as the switch and `fs1.abc.com` as the value. This argument specifies the server machine on which BOS Server is to collect and report binary dates.
- `-file ptserver kaserver` is an argument that takes multiple values. The switch is -file and the values are `ptserver` and `kaserver`. This argument tells the BOS Server to report the modification dates on the files */usr/afs/bin/kaserver* and */usr/afs/bin/ptserver*.

Rules for Entering AFS Commands

Enter each AFS command on a single line (press <Return> only at the end of the command). Some commands in this document appear broken across multiple lines, but that is for legibility only.

Use a space to separate each element on a command line from its neighbors. Spaces rather than commas also separate multiple values of an argument.

In many cases, the issuer of a command can reduce the amount of typing necessary by using one or both of the following methods:

- Omitting switches.
- Using accepted abbreviations for operation codes, switches (if they are included at all), and some types of values.

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

Conditions for Omitting Switches

It is always acceptable to type the switch part of an argument, but in many cases it is not necessary. Specifically, switches can be omitted if the following conditions are met.

- All of the command's required arguments appear in the order prescribed by the syntax statement.
- No switch is provided for any argument.
- There is only one value for each argument (but note the important exception discussed in the following paragraph).

Omitting switches is possible only because there is a prescribed order for each command's arguments. When the issuer does not include switches, 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, the next element is the command's second argument, and so on. The important exception is when a command's final required argument accepts multiple values. In this case, the command interpreter assumes that the issuer has correctly provided one value for each argument up through the final one, so any additional values at the end belong to the final argument.

The following list describes the rules for omitting switches from the opposite perspective: an argument's switch must be provided when any of the following conditions apply.

- The command's arguments do not appear in the prescribed order.
- An optional argument is omitted but a subsequent optional argument is provided.
- A switch is provided for a preceding argument.
- More than one value is supplied for a preceding argument (which must take multiple values, of course); without a switch on the current argument, the command interpreter assumes that the current argument is another value for the preceding argument.

An Example of Omitting Switches

Consider again the example command from An Example Command.

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

This command has two required arguments: the server machine name (identified by the -server switch) and binary file name (identified by the -file switch). The second argument accepts multiple values. By complying with all three conditions, the issuer can omit the switches:

```
% bos getdate fs1.abc.com ptserver kaserver
```

Because there are no switches, the bos command interpreter relies on the order of arguments. It assumes that the first element following the operation code, `fs1.abc.com`, is the server machine name, and that the next argument, `ptserver`, is a binary file name. Then, because the command's second (and last) argument accepts multiple values, the command interpreter correctly interprets `kaserver` as an additional value for it.

On the other hand, the following is not acceptable because it violates the first two conditions in Conditions for Omitting Switches: even though there is only one value 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 fs1.abc.com
```

Rules for Using Abbreviations and Aliases

This section explains how to abbreviate operation codes, option names, server machine names, partition names, and cell names. It is not possible to abbreviate other types of values.

Abbreviating Operation Codes

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

For example, it is acceptable 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 must 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

In addition to 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 know the former name. For example, bos listhosts has the alias bos getcell, its former name. It is acceptable to abbreviate aliases to their shortest unambiguous form (for example, bos getcell to bos getc).

Even if an operation code has an alias, it is still acceptable to use the shortest unambiguous form. Thus, the fs setacl command has three acceptable forms: fs setacl (the full form), fs seta (the shortest abbreviation), and fs sa (the alias).

Abbreviating Switches and Flags

It is acceptable to shorten a switch or flag to the shortest form that distinguishes it from the other switches and flags for its operation code. It is often possible to omit switches entirely, subject to the conditions listed in Conditions for Omitting Switches.

Abbreviating Server Machine Names

AFS server machines must have fully-qualified Internet-style host names (for example, `fs1.abc.com`), but it is not always necessary to type the full name on the command line. AFS commands accept unambiguous shortened forms, but depend on the cell's name service (such as the Domain Name Service) or a local host table to resolve a shortened name to the fully-qualified equivalent when the command is issued.

Most commands also accept the dotted decimal form of the machine's IP address as an identifier.

Abbreviating Partition Names

Partitions that house AFS volumes must have names of the form `/vicepx` or `/vicepxx`, where the variable final portion is one or two lowercase letters. By convention, the first server

partition created on a file server machine is called */vicepa*, the second */vicepb*, and so on. The *IBM AFS Quick Beginnings* 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
```

After /vicepz (for which the index is 25) comes

```
/vicepaa    =     vicepaa     =      aa     =      26
/vicepab    =     vicepab     =      ab     =      27
```

and so on through

```
/vicepiv    =     vicepiv     =      iv     =      255
```

Abbreviating Cell Names

A cell's full name usually matches its Internet domain name (such as stateu.edu for the State University or `abc.com` for ABC Corporation). Some AFS commands accept unambiguous shortened forms, usually with respect to the local */usr/vice/etc/CellServDB file* but sometimes depending on the ability of the local name service to resolve the corresponding domain name.

Displaying Online Help for AFS Commands

To display online help for AFS commands that belong to suites, use the help and apropos operation codes. A -help flag is also available on every almost every AFS command.

The online help entry for a 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 appear on the next line.
- The final line, which begins with the string `Usage:`, lists the command's options in the prescribed order; online help entries use the same typographical symbols (brackets and so on) as this documentation.

If no operation code is specified, the help operation code displays the first line (short description) for every operation code in the suite:

```
% &lt;command_suite&gt; help
```

If the issuer specifies one or more operation codes, the help operation code displays each command's complete online entry (short description, alias if any, and syntax):

```
% &lt;command_suite&gt; help &lt;operation_code&gt;+
```

The -help flag displays a command's syntax but not the short description or alias:

```
% &lt;command_name&gt; -help
```

The apropos operation code displays the short description of any command in a suite whose operation code or short description includes the specified keyword:

```
% &lt;command_suite&gt; apropos "&lt;help string&gt;"
```

The following example command displays the complete online help entry for the fs setacl command:

```
% fs help setacl
fs setacl: set access control list
aliases: sa
Usage: fs setacl -dir &lt;directory&gt;+ -acl &lt;access list entries&gt;+
[-clear] [-negative] [-id] [-if] [-help]
```

To see only the syntax statement, use the -help flag:

```
% fs setacl -help
Usage: fs setacl -dir &lt;directory&gt;+ -acl &lt;access list entries&gt;+
[-clear] [-negative] [-id] [-if] [-help]
```

In the following example, a user wants to display the quota for her home volume. She knows that the relevant command belongs to the fs suite, but cannot remember the operation code. She uses quota as the keyword:

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

The following illustrates the error message that results if no command name or short description contains the keyword:

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

Privilege Required

Many AFS commands require one or more types of administrative privilege. See the reference page for each command.

See Also

"afsd(8)" on page 523, "afsmonitor(1)" on page 25, "backup(8)" on page 531, "bos(8)" on page 661, "bosserver(8)" on page 759, "buserver(8)" on page 763, "butc(8)" on page 767, "dlog(1)" on page 35, "dpass(1)" on page 39, "fileserver(8)" on page 771, "fms(8)" on page 779, "fs(1)" on page 43, "fstrace(8)" on page 783, "kadb_check(8)" on page 803, "kas(8)" on page 805, "kaserver(8)" on page 859, "kdb(8)" on page 863, "klog(1)" on page 153, "knfs(1)" on page 159, "kpasswd(1)" on page 163, "kpwvalid(8)" on page 867, "package(1)" on page , "pagsh(1)" on page 171, "prdb_check(8)" on page 873, "pts(1)" on page 173, "ptserver(8)" on page 875, "rxdebug(1)" on page 227, "salvager(8)" on page 879, "scout(1)" on page 261, "sys(1)" on page 267, "tokens(1)" on page 269, "translate_et(1)" on page 273, "unlog(1)" on page 283, "up(1)" on page 285, "upclient(8)" on page 885, "upserver(8)" on page 889, "uss(8)" on page 893, "vldb_check(8)" on page 913, "vlserver(8)" on page 915, "volinfo(8)" on page 921, "volserver(8)" on page 925, "vos(1)" on page 289, "xfs_size_check(8)" on page 927, "xstat_cm_test(1)" on page 393, "xstat_fs_test(1)" on page 395

Copyright

afsmonitor

Name

`afsmonitor` — Monitors File Servers and Cache Managers

Synopsis

afsmonitor [initcmd] [-config <*configuration file*>] [-frequency <*poll frequency, in seconds*>] [-output <*storage file name*>] [-detailed] [-debug <*debug output file*>] [-fshosts <*list of file servers to monitor*>+] [-cmhosts <*list of cache managers to monitor*>+] [-buffers <*number of buffer slots*>] [-help]

afsmonitor [i] [-co <*configuration file*>] [-fr <*poll frequency, in seconds*>] [-o <*storage file name*>] [-det] [-deb <*debug output file*>] [-fs <*list of file servers to monitor*>+] [-cm <*list of cache managers to monitor*>+] [-b <*number of buffer slots*>] [-h]

Description

The afsmonitor command initializes a program that gathers and displays statistics about specified File Server and Cache Manager operations. It allows the issuer to monitor, from a single location, a wide range of File Server and Cache Manager operations on any number of machines in both local and foreign cells.

There are 271 available File Server statistics and 571 available Cache Manager statistics, listed in the appendix about afsmonitor statistics in the *IBM AFS Administration Guide*. By default, the command displays all of the relevant statistics for the file server machines named by the -fshosts argument and the client machines named by the -cmhosts argument. To limit the display to only the statistics of interest, list them in the configuration file specified by the -config argument. In addition, use the configuration file for the following purposes:

- To set threshold values for any monitored statistic. When the value of a statistic exceeds the threshold, the afsmonitor command displays it in reverse video. There are no default threshold values.
- To invoke a program or script automatically when a statistic exceeds its threshold. The AFS distribution does not include any such scripts.
- To list the file server and client machines to monitor, instead of using the -fshosts and -cmhosts arguments.

For a description of the configuration file, see "afsmonitor(5)" on page 447.

Cautions

The following software must be accessible to a machine where the afsmonitor program is running:

- The AFS xstat libraries, which the afsmonitor program uses to gather data.

- The curses graphics package, which most UNIX distributions provide as a standard utility.

The afsmonitor screens format successfully both on so-called dumb terminals and in windowing systems that emulate terminals. For the output to looks its best, the display environment needs to support reverse video and cursor addressing. Set the TERM environment variable to the correct terminal type, or to a value that has characteristics similar to the actual terminal type. The display window or terminal must be at least 80 columns wide and 12 lines long.

The afsmonitor program must run in the foreground, and in its own separate, dedicated window or terminal. The window or terminal is unavailable for any other activity as long as the afsmonitor program is running. Any number of instances of the afsmonitor program can run on a single machine, as long as each instance runs in its own dedicated window or terminal. Note that it can take up to three minutes to start an additional instance.

Options

initcmd

Accommodates the command's use of the AFS command parser, and is optional.

-config <*file*>

Names the configuration file which lists the machines to monitor, statistics to display, and threshold values, if any. A partial pathname is interpreted relative to the current working directory. Provide this argument if not providing the -fshosts argument, -cmhosts argument, or neither. For instructions on creating this file, see the preceding DESCRIPTION section, and the section on the afsmonitor program in the *IBM AFS Administration Guide*.

-frequency <*poll frequency*>

Specifies in seconds how often the afsmonitor program probes the File Servers and Cache Managers. Valid values range from `1` to `86400` (which is 24 hours); the default value is `60`. This frequency applies to both File Servers and Cache Managers, but the afsmonitor program initiates the two types of probes, and processes their results, separately. The actual interval between probes to a host is the probe frequency plus the time required for all hosts to respond.

-output <*file*>

Names the file to which the afsmonitor program writes all of the statistics that it collects. By default, no output file is created. See the section on the afsmonitor command in the *IBM AFS Administration Guide* for information on this file.

-detailed

Formats the information in the output file named by -output argument in a maximally readable format. Provide the -output argument along with this one.

-fshosts <*host*>+

Names one or more machines from which to gather File Server statistics. For each machine, provide either a fully qualified host name, or an unambiguous abbreviation (the ability to resolve an abbreviation depends on the state of the cell's name service at the time the command is issued). This argument can be combined with the -cmhosts argument, but not with the -config argument.

-cmhosts <*host*>+

Names one or more machines from which to gather Cache Manager statistics. For each machine, provide either a fully qualified host name, or an unambiguous abbreviation (the ability to resolve an abbreviation depends on the state of the cell's name service at the time the command is issued). This argument can be combined with the -fshosts argument, but not with the -config argument.

-buffers <*slots*>

Is nonoperational and provided to accommodate potential future enhancements to the program.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The afsmonitor program displays its data on three screens:

System Overview

This screen appears automatically when the afsmonitor program initializes. It summarizes separately for File Servers and Cache Managers the number of machines being monitored and how many of them have *alerts* (statistics that have exceeded their thresholds). It then lists the hostname and number of alerts for each machine being monitored, indicating if appropriate that a process failed to respond to the last probe.

File Server

This screen displays File Server statistics for each file server machine being monitored. It highlights statistics that have exceeded their thresholds, and identifies machines that failed to respond to the last probe.

Cache Managers

This screen displays Cache Manager statistics for each client machine being monitored. It highlights statistics that have exceeded their thresholds, and identifies machines that failed to respond to the last probe.

Fields at the corners of every screen display the following information:

- In the top left corner, the program name and version number.

- In the top right corner, the screen name, current and total page numbers, and current and total column numbers. The page number (for example, `p. 1 of 3`) indicates the index of the current page and the total number of (vertical) pages over which data is displayed. The column number (for example, `c. 1 of 235`) indicates the index of the current leftmost column and the total number of columns in which data appears. (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 the available commands. Enter the first letter in the command name to run that command. Only the currently possible options appear; for example, if there is only one page of data, the `next` and `prev` commands, which scroll the screen up and down respectively, do not appear. For descriptions of the commands, see the following section about navigating the display screens.
- In the bottom right corner, the `probes` field reports how many times the program has probed File Servers (`fs`), Cache Managers (`cm`), or both. The counts for File Servers and Cache Managers can differ. The `freq` field reports how often the program sends probes.

Navigating the afsmonitor Display Screens

As noted, the lower left hand corner of every display screen displays the names of the commands currently available for moving to alternate screens, which can either be a different type or display more statistics or machines of the current type. To execute a command, press the lowercase version of the first letter in its name. Some commands also have an uppercase version that has a somewhat different effect, as indicated in the following list.

`cm`

Switches to the `Cache Managers` screen. Available only on the `System Overview` and `File Servers` screens.

`fs`

Switches to the `File Servers` screen. Available only on the `System Overview` and the `Cache Managers` screens.

`left`

Scrolls horizontally to the left, to access the data columns situated to the left of the current set. Available when the <<< symbol appears at the top left of the screen. Press uppercase `L` to scroll horizontally all the way to the left (to display the first set of data columns).

`next`

Scrolls down vertically to the next page of machine names. Available when there are two or more pages of machines and the final page is not currently displayed. Press uppercase `N` to scroll to the final page.

`oview`

Switches to the `System Overview` screen. Available only on the `Cache Managers` and `File Servers` screens.

`prev`

Scrolls up vertically to the previous page of machine names. Available when there are two or more pages of machines and the first page is not currently displayed. Press uppercase `N` to scroll to the first page.

`right`

Scrolls horizontally to the right, to access the data columns situated to the right of the current set. This command is available when the `>>>` symbol appears at the upper right of the screen. Press uppercase `R` to scroll horizontally all the way to the right (to display the final set of data columns).

The System Overview Screen

The `System Overview` screen appears automatically as the afsmonitor program initializes. This screen displays the status of as many File Server and Cache Manager processes as can fit in the current window; scroll down to access additional information.

The information on this screen is split into File Server information on the left and Cache Manager information on the right. The header for each grouping reports two pieces of information:

- The number of machines on which the program is monitoring the indicated process.
- The number of alerts and the number of machines affected by them (an *alert* means that a statistic has exceeded its threshold or a process failed to respond to the last probe).

A list of the machines being monitored follows. If there are any alerts on a machine, the number of them appears in square brackets to the left of the hostname. If a process failed to respond to the last probe, the letters `PF` (probe failure) appear in square brackets to the left of the hostname.

The File Servers Screen

The `File Servers` screen displays the values collected at the most recent probe for File Server statistics.

A summary line at the top of the screen (just below the standard program version and screen title blocks) specifies the number of monitored File Servers, the number of alerts, and the number of machines affected by the alerts.

The first column always displays the hostnames of the machines running the monitored File Servers.

To the right of the hostname column appear as many columns of statistics as can fit within the current width of the display screen or window; each column requires space for 10 characters. The name of the statistic appears at the top of each column. If the File Server on a machine did not respond to the most recent probe, a pair of dashes (`--`) appears in each column. If a value exceeds its configured threshold, it is highlighted in reverse video. If a value is too large to fit into the allotted column width, it overflows into the next row in the same column.

The Cache Managers Screen

The `Cache Managers` screen displays the values collected at the most recent probe for Cache Manager statistics.

A summary line at the top of the screen (just below the standard program version and screen title blocks) specifies the number of monitored Cache Managers, the number of alerts, and the number of machines affected by the alerts.

The first column always displays the hostnames of the machines running the monitored Cache Managers.

To the right of the hostname column appear as many columns of statistics as can fit within the current width of the display screen or window; each column requires space for 10 characters. The name of the statistic appears at the top of each column. If the Cache Manager on a machine did not respond to the most recent probe, a pair of dashes (`--`) appears in each column. If a value exceeds its configured threshold, it is highlighted in reverse video. If a value is too large to fit into the allotted column width, it overflows into the next row in the same column.

Writing to an Output File

Include the -output argument to name the file into which the afsmonitor program writes all of the statistics it collects. The output file can be useful for tracking performance over long periods of time, and enables the administrator to apply post-processing techniques that reveal system trends. The AFS distribution does not include any post-processing programs.

The output file is in ASCII format and records the same information as the `File Server` and `Cache Manager` display screens. Each line in the file uses the following format to record the time at which the afsmonitor program gathered the indicated statistic from the Cache Manager (`CM`) or File Server (`FS`) running on the machine called *host_name*. If a probe failed, the error code `-1` appears in the *statistic* field.

```
&lt;time&gt;  &lt;host_name&gt;  CM|FS   &lt;statistic&gt;
```

If the administrator usually reviews the output file manually, rather than using it as input to an automated analysis program or script, including the -detail flag formats the data in a more easily readable form.

Examples

For examples of commands, display screens, and configuration files, see the section about the afsmonitor program in the *IBM AFS Administration Guide*.

Privilege Required

None

See Also

"afsmonitor(5)" on page 447 "fstrace(8)" on page 783, "scout(1)" on page 261

Copyright

cmdebug

Name

cmdebug — Reports the status of a host Cache Manager

Synopsis

cmdebug -servers <*machine*> [-port <*IP port*>] [-long] [-refcounts] [-callbacks] [-addrs] [-cache]

cmdebug -s <*machine*> [-p <*IP port*>] [-l] [-r] [-c] [-a] [-h]

Description

The cmdebug command displays information about the Cache Manager and client cache status on an AFS client machine. By default, it displays all locked cache entries, but other information can be requested via various options.

Options

-servers <*machine*>

Names the client machine for which to display Cache Manager status. Provide the machine's IP address in dotted decimal format, its fully qualified host name (for example, fs1.abc.com), or the shortest abbreviated form of its host name that distinguishes it from other machines. Successful use of an abbreviated form depends on the availability of a name resolution service (such as the Domain Name Service or a local host table) at the time the command is issued.

-port <*IP port*>

Identifies the port on which to contact the Cache Manager. By default, the standard port 7001 is used, so this option is very rarely needed.

-long

Reports on all lock statuses and all cache entries, rather than only locked cache entries. Do not use this option with -refcounts, -callbacks, -addrs, or -cache.

-refcounts

Reports only those cache entries with non-zero reference counts. Do not use this option with -long, -callbacks, -addrs, or -cache.

-callbacks

Reports only those cache entries with callbacks. Do not use this option with -long, -refcounts, -addrs, or -cache.

-addrs

Rather than showing any cache entries, displays the interfaces the Cache Manager answers on, including their netmasks and MTUs. This is useful for analyzing clients that are multihomed and identifying problems with netmasks or MTU settings. Do not use this option with -long, -refcounts, -callbacks, or -cache.

-cache

Rather than showing any cache entries, displays the cache configuration for the client machine. The information displayed is essentially the information that can be configured via parameters to afsd. Do not use this option with -long, -refcounts, -callbacks, or -addrs.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

Displays all of the locked cache entries on `client1`:

```
% cmdebug client1
```

Displays the cache configuration for `client1.abc.com`:

```
% cmdebug client1.abc.com -cache
```

Displays all cache entries for `client2.abc.com`:

```
% cmdebug client2.abc.com -long
```

Privilege Required

None

See Also

"afsd(8)" on page 523

Copyright

dlog

Name

`dlog` — Authenticates to the DCE Security Service

Synopsis

dlog [-principal <*user name*>] [-cell <*cell name*>] [-password <*user's password*>] [-servers <*explicit list of servers*>+] [-lifetime <*ticket lifetime in hh[:mm[:ss]]*>] [-setpag] [-pipe] [-help]

dlog [-pr <*user name*>] [-c <*cell name*>] [-pw <*user's password*>] [-ser <*explicit list of servers*>+] [-l <*ticket lifetime in hh[:mm[:ss]]*>] [-set] [-pi] [-h]

Description

The dlog command obtains DCE credentials for the issuer from the DCE Security Service in the cell named by the -cell argument, and stores them on the AFS client machine on which the user issues the command. The AFS/DFS Migration Toolkit Protocol Translator processes running on machines in the DCE cell accept the credentials, which enables the user to access the DCE cell's filespace from the AFS client. The user's identity in the local file system is unchanged.

If the issuer does not provide the -principal argument, the dlog command interpreter uses the user name under which the issuer is logged into the local file system. Provide the DCE password for the appropriate user name. As with the klog command, the password does not cross the network in clear text (unless the issuer is logged into the AFS client from a remote machine).

The credentials are valid for a lifetime equivalent to the smallest of the following, all but the last of which is defined by the DCE cell's Security Server:

- The maximum certificate lifetime for the issuer's DCE account.
- The maximum certificate lifetime for the AFS principal's DCE account.
- The registry-wide maximum certificate lifetime.
- The registry-wide default certificate lifetime.
- The lifetime requested using the -lifetime argument.

If the previous maximum certificate lifetime values are set to `default-policy`, the maximum possible ticket lifetime is defined by the default certificate lifetime. Refer to the DCE vendor's administration guide for more information before setting any of these values.

The AFS Cache Manager stores the ticket in a credential structure associated with the name of the issuer (or the user named by the -principal argument. If the user already has a ticket for the DCE cell, the ticket resulting from this command replaces it in the credential structure.

The AFS tokens command displays the ticket obtained by the dlog command for the server principal `afs`, regardless of the principal to which it is actually granted. Note that the tokens command does not distinguish tickets for a DFSTM File Server from tickets for an AFS File Server.

Options

-principal <*user name*>

Specifies the DCE user name for which to obtain DCE credentials. If this option is omitted, the dlog command interpreter uses the name under which the issuer is logged into the local file system.

-cell <*cell name*>

Specifies the DCE cell in which to authenticate. During a single login session on a given machine, a user can authenticate in multiple cells simultaneously, but can have only one ticket at a time for each cell (that is, it is possible to authenticate under only one identity per cell per machine). It is legal to abbreviate the cell name to the shortest form that distinguishes it from the other cells listed in the `/usr/vice/etc/CellServDB` file on the local client machine.

If the issuer does not provide the -cell argument, the dlog command attempts to authenticate with the DCE Security Server for the cell defined by

- The value of the environment variable AFSCELL on the local AFS client machine, if defined. The issuer can set the AFSCELL environment variable to name the desired DCE cell.
- The cell name in the `/usr/vice/etc/ThisCell` file on the local AFS client machine. The machine's administrator can place the desired DCE cell's name in the file.

-password <*user's password*>

Specifies the password for the issuer (or for the user named by the -principal argument). Using this argument is not recommended, because it makes the password visible on the command line. If this argument is omitted, the command prompts for the password and does not echo it visibly.

-servers <*list of servers*>+

Specifies a list of DFS database server machines running the Translator Server through which the AFS client machine can attempt to authenticate. Specify each server by hostname, shortened machine name, or IP address. If this argument is omitted, the dlog command interpreter randomly selects a machine from the list of DFS Fileset Location (FL) Servers in the `/usr/vice/etc/CellServDB` file for the DCE cell specified by the -cell argument. This argument is useful for testing when authentication seems to be failing on certain server machines.

-lifetime <*ticket lifetime*>

Requests a ticket lifetime using the format *hh*:*mm*[:*ss*] (hours, minutes, and optionally a number seconds between 00 and 59). For example, the value `168:30` requests a ticket lifetime of 7 days and 30 minutes, and `96:00` requests a lifetime of 4 days. Acceptable values range from `00:05` (5 minutes) to `720:00` (30 days). If this argument is not provided and no other determinants of ticket lifetime have been changed from their defaults, ticket lifetime is 10 hours.

The requested lifetime must be smaller than any of the DCE cell's determinants for ticket lifetime; see the discussion in the preceding Description section.

-setpag

Creates a process authentication group (PAG) in which the newly created ticket is placed. If this flag is omitted, the ticket is instead associated with the issuers' local user ID (UID).

-pipe

Suppresses any prompts that the command interpreter otherwise produces, including the prompt for the issuer's password. Instead, the command interpreter accepts the password via the standard input stream.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If the dlog command interpreter cannot contact a Translator Server, it produces a message similar to the following:

```
dlog: server or network not responding -- failed to contact
authentication service
```

Examples

The following command authenticates the issuer as cell_admin in the `dce.abc.com` cell.

```
% dlog -principal cell_admin -cell dce.abc.com
Password: &lt;cell_admin's password&gt;
```

In the following example, the issuer authenticates as cell_admin to the `dce.abc.com` cell and request a ticket lifetime of 100 hours. The `tokens` command confirms that the user obtained DCE credentials as the user `cell_admin`: the AFS ID is equivalent to the UNIX ID of `1` assigned to `cell_admin` in `dce.abc.com` cell's DCE registry.

```
% dlog -principal cell_admin -cell dce.abc.com -lifetime 100
Password: &lt;cell_admin's password&gt;


% tokens
Tokens held by the Cache Manager:


User's (AFS ID 1) tokens for afs@dce.abc.com [Expires Jul 6 14:12]
User's (AFS ID 4758) tokens for afs@abc.com [Expires Jul 2 13:14]
```

```
--End of list--
```

Privilege Required

None

See Also

"dpass(1)" on page 39, "klog(1)" on page 153, "tokens(1)" on page 269, "unlog(1)" on page 283

Copyright

dpass

Name

dpass — Returns the DCE password for a new DCE account

Synopsis

dpass [-cell <*original AFS cell name*>] [-help]

dpass [-c <*original AFS cell name*>] [-h]

Description

The dpass command returns the DCE password that an administrator assigned to the issuer when using the dm pass command to migrate AFS user accounts into a DCE cell.

The dpass command, issued on an AFS client, requests the issuer's new DCE password from the AFS cell specified with the -cell argument.

The issuer must be authenticated as the AFS user whose AFS account was moved into DCE, and be able to provide the user's AFS password when prompted by the dpass command.

Options

-cell <*cell name*>

Specifies the name of the AFS cell from which the AFS account was moved into DCE and from which to fetch the new DCE password.

-help

Prints the online help for this command. All other valid options are ignored.

Output

By default, the dpass command writes a message similar to the following to the standard output stream.

```
Please read the following message before entering your password.


This program will display your new, temporary DCE password on your
terminal, and you should change the assigned password as soon as
possible (from a DCE client).  The program assumes that the AFS cell
uses the AFS Authentication Server and that an administrator used the
utilities in the AFS/DFS Migration Toolkit to migrate the account from
AFS to DCE. The password you enter should be the AFS password that was
in effect when your DCE account was created; this is not necessarily
the same password you have at the moment.  The cell name (which you may
```

```
override with a command line option), must be the name of the AFS cell
from which the authentication information was taken.
```

To suppress this message, set the DPASS_NO_MESSAGE environment variable. It is then possible to substitute a customized message if desired by using a script similar to the following example:

```
#! /bin/csh
echo "I&lt;Start of customized message&gt;"
echo "I&lt;Continuation of customized message&gt;"
  .
  .
  .
echo "I&lt;Conclusion of customized message&gt;"
setenv DPASS_NO_MESSAGE
dpass $*
```

After the standard or customized message, if any, the dpass command generates the following prompt for the original AFS password:

```
Original password for AFS cell &lt;cell&gt;:
Re-enter password to verify:
```

If the AFS passwords match and are correct, the command reports the temporary DCE password in the following message.

```
The new DCE password is: &lt;Issuer’s_temporary_DCE_password
```

Examples

The following example returns the DCE password of the issuer, whose AFS account is in the `abc.com` cell. The DPASS_NO_MESSAGE variable has been set to suppress the standard message.

```
% dpass
Original password for AFS cell abc.com: &lt;Issuer’s_AFS_password&gt;
Re-enter password to verify: &lt;Issuer’s_AFS_password&gt;
The new DCE password is: 8655--eg8e-dcdc-8157
```

Privilege Required

The issuer must be authenticated as the AFS user for whom to display the corresponding DCE password.

See Also

"dlog(1)" on page 35

dm pass reference page in *IBM AFS/DFS Migration Toolkit Administration Guide and Reference*

Copyright

fs

Name

`fs` — Introduction to the fs command suite

Description

The commands in the fs command suite constitute the main administrative interface to the Cache Manager on an AFS client machine, which is responsible for fetching AFS data from file server machines on behalf of applications running on the client machine.

There are several categories of commands in the fs command suite:

- Commands to set and report how the Cache Manager interacts with server machines: fs checkservers, fs getcellstatus, fs getserverprefs, fs listcells, fs newcell, fs setcell, fs setserverprefs, fs sysname, and fs wscell.
- Commands to administer access control lists (ACLs): fs cleanacl, fs copyacl, fs listacl, and fs setacl.
- Commands to administer server machines, volumes or partitions that house a given file or directory: fs diskfree, fs examine, fs listquota, fs quota, fs setquota, fs setvol, fs whereis, and fs whichcell.
- Commands to administer the local client cache and related information: fs checkvolumes, fs flush, fs flushvolume, fs getcacheparms, and fs setcachesize.
- Commands to administer volume mount points: fs lsmount, fs mkmount, and fs rmmount.
- Commands to control monitoring and tracing: fs debug, and fs messages.
- A command to administer the Cache Manager's interaction with other file systems: fs exportafs.
- Commands to obtain help: fs apropos and fs help.

The Cache Manager and the fs commands use and maintain the following configuration files:

`/usr/vice/etc/CellServDB`

Lists the database server machines in the local cell and any foreign cell to which the administrator wishes to enable AFS access for users working on the machine. The database server machines run the Authentication, Backup, Protection and Volume Location (VL) Server processes, which maintain databases of administrative information. For users to access a cell, its `root.cell` volume must also be mounted in the local cell's AFS file tree.

`/usr/vice/etc/ThisCell`

Defines the machine's cell membership with respect to the AFS command suites and Cache Manager access to AFS data.

/usr/vice/etc/cacheinfo

Defines configuration parameters for the cache, including its size and whether it is in memory or on disk.

In addition, the Cache Manager automatically creates files on the cache partition (by default, */usr/vice/cache* for caching and tracking files fetched from file server machines.

For more details, see the reference page for each file.

Options

The following flag is available on every command in the fs suite. The reference page for each command also lists it, but it is described here in greater detail.

-help

Prints a command's online help message on the standard output stream. Do not combine this flag with any of the command's other options; when it is provided, the command interpreter ignores all other options, and only prints the help message.

Privilege Required

The privileges required for fs commands vary more than for other command suites. Pay special attention to the PRIVILEGE REQUIRED section of each command description.

The various types of necessary privilege include:

- Having permissions on a directory's ACL. For example, creating and removing mount points requires `a` (administer), `i` (insert), and `d` (delete) permissions on the ACL of the directory in which the mount point resides.
- Being logged onto the machine as the local superuser `root`. This is necessary when issuing commands that affect Cache Manager configuration.
- Belonging to the system:administrators group in the Protection Database.
- No privilege. Many fs commands simply list information.

See Also

"afs_cache(5)" on page 443, "CellServDB(5)" on page 415, "ThisCell(5)" on page 435 "cacheinfo(5)" on page 467, "fs_apropos(1)" on page 47, "fs_checkservers(1)" on page 49, "fs_checkvolumes(1)" on page 53, "fs_cleanacl(1)" on page 55, "fs_copyacl(1)" on page 57, "fs_diskfree(1)" on page 61, "fs_examine(1)" on page 65, "fs_exportafs(1)" on page 69, "fs_flush(1)" on page 73, "fs_flushmount(1)" on page 75, "fs_flushvolume(1)" on page 77, "fs_getcacheparms(1)" on page 79, "fs_getcellstatus(1)" on page 81, "fs_getclientaddrs(1)" on page 83, "fs_getserverprefs(1)" on page 85, "fs_help(1)" on page 89, "fs_listacl(1)" on page 91, "fs_listcells(1)" on page 95, "fs_listquota(1)" on page 97, "fs_lsmount(1)" on page 99, "fs_messages(1)" on page 103, "fs_mkmount(1)" on page 105, "fs_newcell(1)" on page 109, "fs_quota(1)" on page 113, "fs_rmmount(1)" on page 115, "fs_setacl(1)" on page 117,

"fs_setcachesize(1)" on page 123, "fs_setcell(1)" on page 125, "fs_setclientaddrs(1)" on page 127, "fs_setquota(1)" on page 131, "fs_setserverprefs(1)" on page 133, "fs_setvol(1)" on page 139, "fs_storebehind(1)" on page 141, "fs_sysname(1)" on page 145, "fs_whereis(1)" on page 147, "fs_whichcell(1)" on page 149, "fs_wscell(1)" on page 151

Copyright

fs

fs apropos

Name

`fs apropos` — Displays each help entry containing a keyword string

Synopsis

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

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

Description

The fs apropos command displays the first line of the online help entry for any fs command that has in its name or short description the string specified by the -topic argument.

To display the syntax for a command, use the fs help command.

Options

-topic <*help string*>

Specifies the keyword string to match, in lowercase letters only. If the string is more than a single word, surround it with double quotes ("") or other delimiters.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of a command's online help entry names it and briefly describes its function. This command displays the first line for any fs command where the string specified with the -topic argument is part of the command name or first line.

Examples

The following command lists all fs commands that include the word cache in their names or short online descriptions:

```
% fs apropos cache
setcachesize: set cache size
flush: flush file from cache
getcacheparms: get cache usage info
monitor: set cache monitor host address
```

Privilege Required

None

See Also

"fs(1)" on page 43, "fs_help(1)" on page 89

Copyright

fs checkservers

Name

`fs checkservers` — Displays the status of server machines

Synopsis

fs checkservers [-cell <*cell to check*>] [-all] [-fast] [-interval <*seconds between probes*>] [-help]

fs checks [-c <*cell to check*>] [-a] [-f] [-i <*seconds between probes*>] [-h]

Description

The fs checkservers command reports whether certain AFS server machines are accessible from the local client machine. The machines belong to one of two classes, and the Cache Manager maintains a list of them in kernel memory:

- The database server machines in every cell listed in the local */usr/vice/etc/CellServDB file*, plus any machines added to the memory list by the fs newcell command since the last reboot.
- All file server machines the Cache Manager has recently contacted, and which it probably needs to contact again soon. In most cases, the Cache Manager holds a callback on a file or volume fetched from the machine.

If the Cache Manager is unable to contact the vlserver process on a database server machine or the fileserver process on a file server machine, it marks the machine as inaccessible. (Actually, if a file server machine is multihomed, the Cache Manager attempts to contact all of the machine's interfaces, and only marks the machine as down if the fileserver fails to reply via any of them.) The Cache Manager then periodically (by default, every three minutes) sends a probe to each marked machine, to see if it is still inaccessible. If a previously inaccessible machine responds, the Cache Manager marks it as accessible and no longer sends the periodic probes to it.

The fs checkservers command updates the list of inaccessible machines by having the Cache Manager probe a specified set of them:

- By default, only machines that are marked inaccessible and belong to the local cell (the cell listed in the local */usr/vice/etc/ThisCell* file).
- If the -cell argument is included, only machines that are marked inaccessible and belong to the specified cell.
- If the -all flag is included, all machines marked inaccessible.

If the -fast flag is included, the Cache Manager does not probe any machines, but instead reports the results of the most recent previous probe.

To set the interval between probes rather than produce a list of inaccessible machines, use the -interval argument. The non-default setting persists until the machine reboots; to preserve it

across reboots, put the appropriate fs checkservers command in the machine's AFS initialization files.

Cautions

The command can take quite a while to complete, if a number of machines do not respond to the Cache Manager's probe. The Cache Manager probes machines sequentially and waits a standard timeout period before marking the machine as unresponsive, to allow for slow network communication. To make the command shell prompt return quickly, put the command in the background. It is harmless to interrupt the command by typing Ctrl-C or another interrupt signal.

Note that the Cache Manager probes only server machines marked inaccessible in its memory list. A server machine's absence from the output does not necessarily mean that it is functioning, because it possibly is not included in the memory list at all (if, for example, the Cache Manager has not contacted it recently). For the same reason, the output is likely to vary on different client machines.

Unlike most fs commands, the fs checkservers command does not refer to the AFSCELL environment variable.

Options

-cell <*cell to check*>

Names each cell in which to probe server machines marked as inaccessible. Provide the fully qualified domain name, or a shortened form that disambiguates it from the other cells listed in the local `/usr/vice/etc/CellServDB` file. Combine this argument with the -fast flag if desired, but not with the -all flag. Omit both this argument and the -all flag to probe machines in the local cell only.

-all

Probes all machines in the Cache Manager's memory list that are marked inaccessible. Combine this argument with the -fast flag if desired, but not with the -cell argument. Omit both this flag and the -cell argument to probe machines in the local cell only.

-fast

Displays the Cache Manager's current list of machines that are inaccessible, rather than sending new probes. The output can as old as the current setting of the probe interval (by default three minutes, and maximum ten minutes).

-interval <*seconds between probes*>

Sets or reports the number of seconds between the Cache Manager's probes to machines in the memory list that are marked inaccessible:

- To set the interval, specify a value from the range between 1 and `600` (10 minutes); the default is `180` (three minutes). The issuer must be logged in as the local superuser `root`.

The altered setting persists until again changed with this command, or until the machine reboots, at which time the setting returns to the default.

- Provide a value of `0` (zero) to display the current interval setting. No privilege is required. Do not combine this argument with any other.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If there are no machines marked as inaccessible, or if all of them now respond to the Cache Manager's probe, the output is:

```
All servers are running.
```

Note that this message does not mean that all server machines in each relevant cell are running. The output indicates the status of only those machines that the Cache Manager probes.

If a machine fails to respond to the probe within the timeout period, the output begins with the string

```
These servers unavailable due to network or server problems:
```

and lists the hostname of each machine on its own line. The Cache Manager stores machine records by Internet address, so the format of each hostname (uppercase or lowercase letters, or an Internet address in dotted decimal format) depends on how the local cell's name service translates it at the time the command is issued. If a server machine is multihomed, the output lists only one of its interfaces (usually, the currently most preferred one).

If the -interval argument is provided with a value between `1` and `600`, there is no output. If the value is `0`, the output reports the probe interval as follows:

```
The current down server probe interval is &lt;interval&gt; secs
```

Examples

The following command displays the Cache Manager's current list of unresponsive machines in the local cell, rather than probing them again. The output indicates that if there were any machines marked inaccessible, they all responded to the previous probe.

```
% fs checkservers -fast
All servers are running.
```

The following example probes machines in the Cache Manager's memory list that belong to the `stateu.edu` cell:

```
% fs checkservers -cell stateu.edu
All servers are running.
```

The following example probes all server machines in the Cache Manager's memory list. It reports that two machines did not respond to the probe.

```
% fs checkservers -all
These servers unavailable due to network or server problems:
fs1.abc.com SV3.STATE.EDU.
```

Privilege Required

To set the probe interval, the issuer must be logged in as the local superuser `root`. Otherwise, no privilege is required.

See Also

"CellServDB(5)" on page 415, "ThisCell(5)" on page 435, "fs_newcell(1)" on page 109

Copyright

fs checkvolumes

Name

`fs checkvolumes` — Forces the Cache Manager to update volume information

Synopsis

fs checkvolumes [-help]

fs checkv [-h]

Description

The fs checkvolumes command discards the table of mappings between volume names and volume ID numbers that the Cache Manager stores in memory and uses when fetching data from volumes. The next time an application requests AFS data, the Cache Manager must contact the Volume Location (VL) Server for volume location information, and then an appropriate file server machine for the actual data.

The Cache Manager updates the table of mappings periodically (by default, hourly), but this command is useful if the issuer knows that a volume's name has changed, or that new read-only replicas of a volume have been released, because issuing it forces the Cache Manager to reference the changed volume.

Options

-help

Prints the online help for this command. All other valid options are ignored.

Output

The following message confirms that the command ran successfully.

```
All volumeID/name mappings checked.
```

Privilege Required

None

Copyright

fs cleanacl

Name

`fs cleanacl` — Remove obsolete entries from an ACL

Synopsis

fs cleanacl [-path <*dir/file path*>+] [-help]

fs cl [-p <*dir/file path*>+] [-h]

Description

The fs cleanacl command removes from the access control list (ACL) of each specified directory or file any entry that refers to a user or group that no longer has a Protection Database entry. Such an entry appears on the ACL as an AFS user ID number (UID) rather than a name, because without a Protection Database entry, the File Server cannot translate the UID into a name.

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 AFS UID. (Note that recycling UIDs is not recommended in any case.)

Options

-path <*dir/file path*>+

Names each directory for which to clean the ACL (specifying a filename cleans its directory's ACL). If this argument is omitted, the current working directory's ACL is cleaned.

Specify the read / write path to each directory, to avoid the failure that results from attempting to change a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). For further discussion of the concept of read / write and read-only paths through the filespace, see the fs mkmount reference page.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If there are no obsolete entries on the ACL, the following message appears:

```
Access list for &lt;path&gt; is fine.
```

Otherwise, the output reports the resulting state of the ACL, following the header

```
Access list for &lt;path&gt; is now
```

At the same time, the following error message appears for each file in the cleaned directories:

```
fs: '&lt;filename&gt;': Not a directory
```

Examples

The following example illustrates the cleaning of the ACLs on the current working directory and two of its subdirectories. Only the second subdirectory had obsolete entries on it.

```
% fs cleanacl -path . ./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

The issuer must have the a (administer) permission on each directory's ACL (or the ACL of each file's parent directory); the directory's owner and the members of the system:administrators group have the right implicitly, even if it does not appear on the ACL.

See Also

"fs_listacl(1)" on page 91, "fs_mkmount(1)" on page 105

Copyright

fs copyacl

Name

`fs copyacl` — Copies an ACL from a directory to one or more other directories

Synopsis

fs copyacl -fromdir <*source directory (or DFS file)*> -todir <*destination directory (or DFS file)*>+ [-clear] [-id] [-if] [-help]

fs co -f <*source directory (or DFS file)*> -t <*destination directory (or DFS file)*>+ [-c] [-id] [-if] [-h]

Description

The fs copyacl command copies the access control list (ACL) from a source directory to each specified destination directory. The source directory's ACL is unchanged, and changes to the destination directory's ACL obey the following rules:

- If an entry on the source ACL does not already exist on the destination ACL, it is added.
- If an entry exists on both the source and destination ACLs, the permissions from the source ACL entry replace the current permissions on the destination ACL entry.
- If an entry on the destination ACL has no corresponding entry on the source ACL, it is removed if the -clear flag is included and is unchanged otherwise. In other words, if the -clear flag is provided, the source ACL completely replaces the destination ACL.

When using this command to copy ACLs between objects in DFS filespace accessed via the AFS / DFS Migration Toolkit Protocol Translator, it is possible to specify files, as well as directories, with the -fromdir and -todir arguments. For more information on copying ACLs between DFS directories and files, refer to the *IBM AFS/DFS Migration Toolkit Administration Guide and Reference*.

Cautions

Do not copy ACLs between AFS and DFS files or directories. The ACL formats are incompatible.

Options

-fromdir <*source directory*>

Specifies the source directory from which to copy the ACL. (Specifying an AFS file copies its directory's ACL, but specifying a DFS file copies its own ACL.) A partial pathname is interpreted relative to the current working directory.

-todir <*destination directory*>

Specifies each directory for which to alter the ACL to match the source ACL. (Specifying an AFS file halts the command with an error, but specifying a DFS file alters the file's ACL). A partial pathname is interpreted relative to the current working directory.

Specify the read / write path to each directory (or DFS file), to avoid the failure that results from attempting to change a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, `/afs/.abc.com`). For further discussion of the concept of read / write and read-only paths through the filespace, see the fs mkmount reference page.

-clear

Replaces the ACL of each destination directory with the source ACL.

-id

Modifies the Initial Container ACL of each DFS directory named by the -todir argument, rather than the regular Object ACL. This argument is supported only when both the source and each destination directory reside in DFS and are accessed via the AFS / DFS Migration Toolkit Protocol Translator.

-if

Modifies the Initial Object ACL of each DFS directory named by the -todir argument, rather than the regular Object ACL. This argument is supported only when both the source and each destination directory reside in DFS and are accessed via the AFS / DFS Migration Toolkit Protocol Translator.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example command copies the current working directory's ACL to its subdirectory called *`reports`*. Note that the source directory's ACL is unaffected. Entries on the *`reports`* directory's that are not on the source ACL of the current directory remain unaffected as well, because the -clear flag is not used.

```
% fs listacl . 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 copyacl -fromdir . -todir reports


% fs listacl . 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

To copy an ACL between AFS objects, the issuer must have the `l` (lookup) permission on the source directory's ACL and the `a` (administer) permission on each destination directory's ACL. If the -fromdir argument names a file rather than a directory, the issuer must have both the `l` and `r` (read) permissions on the ACL of the file's directory.

To copy an ACL between DFS objects, the issuer must have the r permission on the source directory or file's ACL and the c (control) permission on each destination directory or file's ACL.

See Also

"fs_listacl(1)" on page 91, "fs_mkmount(1)" on page 105, "fs_setacl(1)" on page 117

IBM AFS/DFS Migration Toolkit Administration Guide and Reference

Copyright

fs diskfree

Name

`fs diskfree` — Shows data about the partition housing a directory or file

Synopsis

fs diskfree [-path <*dir/file path*>+] [-help]

fs df [-p <*dir/file path*>+] [-h]

fs di [-p <*dir/file path*>+] [-h]

Description

The fs diskfree command formats and displays information about the partition that houses the volume containing the specified directory or file, including its size and how much space is currently used.

To display information about the volume itself, use the fs examine command. The fs examine and fs quota commands also display information about a volume.

Cautions

The partition-related statistics in this command's output do not always agree with the corresponding values in the output of the standard UNIX df command. The statistics reported by this command can be up to five minutes old, because the Cache Manager polls the File Server for partition information at that frequency. Also, on some operating systems, the df command's report of partition size includes reserved space not included in this command's calculation, and so is likely to be about 10% larger.

Options

-path <*dir/file path*>+

Names a file or directory that resides on the partition about which to produce output. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output reports the following information about the volume and partition that houses each file or directory:

Volume Name

The name of the volume.

kbytes

The partition's total size in kilobytes.

used

The number of kilobytes used on the partition.

avail

The number of kilobytes available on the partition.

%used

The percentage of the partition's total space that is used (the used statistic divided by the kbytes statistic, times 100).

If the %used statistic is greater than 90%, it is marked with the string <<WARNING at the right margin.

If the volume is a read-only volume, the output includes information about only one of the partitions that houses it, generally the one on the file server machine with the lowest preference rank. To verify which machine the output is referring to, use the vos listvldb command to list the volume's locations, and the vos partinfo command to display the size of each one.

Examples

The following example shows the output for the partitions housing the volumes user.smith and sun4x_56.bin:

```
% fs diskfree -path /afs/abc.com/usr/smith /afs/abc.com/sun4x_56/bin
Volume Name     kbytes  used     avail     %used
user.smith     4177920 3841258  336662       92% &lt;&lt;WARNING
sun4x_56.bin   4423680 3174500 1249180       72%
```

Privilege Required

The issuer must have the l (lookup) permission on the ACL of the root directory of the volume that houses the file or directory named by the -path argument, and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_examine(1)" on page 65

Copyright

fs examine

Name

`fs examine` — Shows data about the volume containing a directory or file

Synopsis

fs examine [-path <*dir/file path*>+] [-help]

fs exa [-p <*dir/file path*>+] [-h]

fs listvol [-p <*dir/file path*>+] [-h]

fs listv [-p <*dir/file path*>+] [-h]

fs lv [-p <*dir/file path*>+] [-h]

Description

The fs examine command displays information about the volume containing each specified directory or file, including its volume ID number, quota and the percentage of its quota that is used.

This command provides the most information about a volume, but the fs listquota command displays similar information in tabular format, and the fs quota command reports only the percentage of quota used.

To set volume quota, use the fs setquota or fs setvol command.

Cautions

The partition-related statistics in this command's output do not always agree with the corresponding values in the output of the standard UNIX df command. The statistics reported by this command can be up to five minutes old, because the Cache Manager polls the File Server for partition information at that frequency. Also, on some operating systems, the df command's report of partition size includes reserved space not included in this command's calculation, and so is likely to be about 10% larger.

Options

-path <*dir/file path*>+

Names a file or directory that resides in the volume about which to produce output. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output displays information about the volume that houses each specified directory or file, in the following format

```
Volume status for vid = I&lt;volume ID&gt; named I&lt;volume name&gt;
Current offline message is I&lt;message&gt;
Current disk quota is I&lt;quota in kilobytes&gt;
Current blocks used are I&lt;volume size in kilobytes&gt;
The partition has I&lt;available partition&gt; blocks available out of
   I&lt;partition size&gt;
```

where the first line specifies the volume's ID number and name. The `Current offline message` line appears only if an administrator has included the -offlinemsg argument to the fs setvol command. The remaining lines report, respectively,

- The volume's quota in kilobytes, or the string `unlimited` to indicate an unlimited quota.
- The volume's current size in kilobytes.
- The number of blocks available and total size of the host partition, both in kilobytes.

Examples

The following example shows the output for the volume `user.smith` and the partition housing it:

```
% fs examine -path /afs/abc.com/usr/smith
Volume status for vid = 50489902 named user.smith
Current maximum quota is 15000
Current blocks used are 5073
The partition has 336662 blocks available out of 4177920
```

Privilege Required

The issuer must have the `l` (lookup) permission on the ACL of the root directory of the volume that houses the file or directory named by the -path argument, and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_listquota(1)" on page 97, "fs_quota(1)" on page 113, "fs_setquota(1)" on page 131, "fs_setvol(1)" on page 139

Copyright

fs exportafs

Name

`fs exportafs` — Configures export of AFS to clients of other file systems

Synopsis

fs exportafs -type <*exporter name*> [-start <*start/stop translator (on | off)*>] [-convert <*convert from afs to unix mode (on | off)*>] [-uidcheck <*run on strict 'uid check' mode (on | off)*>] [-submounts <*allow nfs mounts to subdirs of /afs/.. (on | off)*>] [-help]

fs exp -t <*exporter name*> [-st <*start/stop translator (on | off)*>] [-c <*convert from afs to unix mode (on | off)*>] [-u <*run on strict 'uid check' mode (on | off)*>] [-su <*allow nfs mounts to subdirs of /afs/.. (on | off)*>] [-h]

Description

The fs exportafs command sets (if the -start argument is provided) or reports (if it is omitted) whether the machine can reexport the AFS filespace to clients of a non-AFS file system. To control certain features of the translation protocol, use the following arguments:

- To control whether the UNIX group and other mode bits on an AFS file or directory are set to match the owner mode bits when it is exported to the non-AFS file system, use the -convert argument.
- To control whether tokens can be placed in a credential structure identified by a UID that differs from the local UID of the entity that is placing the tokens in the structure, use the -uidcheck argument. The most common use is to control whether issuers of the knfs command can specify a value for its -id argument that does not match their local UID on the NFS / AFS translator machine.
- To control whether users can create mounts in the non-AFS filespace to an AFS directory other than `/afs`, use the -submounts argument.

Options

-type <*exporter name*>

Names the alternate file system to which to reexport the AFS filespace. The only acceptable value is `nfs`, in lowercase letters only.

-start on =item -start off

Enables the local machine to reexport the AFS filespace if the value is on, or disables it if the value is `off`. Omit this argument to report the current setting for all of the configurable parameters.

-convert on =item -convert off

Controls the setting of the UNIX group and other mode bits on AFS files and directories exported to the non-AFS file system. If the value is on, they are set to match the owner mode bits. If the value is `off`, the bits are not changed. If this argument is omitted, the default value is on.

-uidcheck on =item -uidcheck off

Controls whether tokens can be placed in a credential structure identified by a UID that differs from the local UID of the entity that is placing the tokens in the structure.

- If the value is on, the UID that identifies the credential structure must match the local UID.

  With respect to the knfs command, this value means that the value of -id argument must match the issuer's local UID on the translator machine. In practice, this setting makes it pointless to include the -id argument to the knfs command, because the only acceptable value (the issuer's local UID) is already used when the -id argument is omitted.

  Enabling UID checking also makes it impossible to issue the klog and pagsh commands on a client machine of the non-AFS file system even though it is a system type supported by AFS. For an explanation, see "klog(1)" on page 153.

- If the value is off (the default), tokens can be assigned to a local UID in the non-AFS file system that does not match the local UID of the entity assigning the tokens.

  With respect to the knfs command, it means that the issuer can use the -id argument to assign tokens to a local UID on the NFS client machine that does not match his or her local UID on the translator machine. (An example is assigning tokens to the MFS client machine's local superuser `root`.) This setting allows more than one issuer of the knfs command to make tokens available to the same user on the NFS client machine. Each time a different user issues the knfs command with the same value for the -id argument, that user's tokens overwrite the existing ones. This can result in unpredictable access for the user on the NFS client machine.

-submounts on =item -submounts off

Controls whether a user of the non-AFS filesystem can mount any directory in the AFS filespace other than the top-level */afs* directory. If the value is on, such submounts are allowed. If the value is `off`, only mounts of the */afs* directory are allowed. If this argument is omitted, the default value is `off`.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If the machine is not even configured as a server of the non-AFS file system, the following message appears:

```
Sorry, the &lt;file_system&gt;-exporter type is currently not supported on
this AFS client
```

If the machine is configured as a server of the non-AFS file system but is not currently enabled to reexport AFS to it (because the -start argument to this command is not set to on), the message is as follows:

```
'&lt;file_system&gt;' translator is disabled
```

If the machine is enabled to reexport AFS, the following message precedes messages that report the settings of the other parameters.

```
'&lt;file_system&gt;' translator is enabled with the following options:
```

The following messages indicate that the -convert argument is set to on or off respectively:

```
Running in convert owner mode bits to world/other mode
Running in strict unix mode
```

The following messages indicate that the -uidcheck argument is set to on or off respectively:

```
Running in strict 'passwd sync' mode
Running in no 'passwd sync' mode
```

The following messages indicate that the -submounts argument is set to on or off respectively:

```
Allow mounts of /afs/.. subdirs
Only mounts to /afs allowed
```

Examples

The following example shows that the local machine can export AFS to NFS client machines.

```
% fs exportafs nfs
'nfs' translator is enabled with the following options:
     Running in convert owner mode bits to world/other mode
     Running in no 'passwd sync' mode
     Only mounts to /afs allowed
```

The following example enables the machine as an NFS server and converts the UNIX group and other mode bits on exported AFS directories and files to match the UNIX owner mode bits.

```
% fs exportafs -type nfs -start on -convert on
```

The following example disables the machine from reexporting AFS to NFS client machines:

```
% fs exportafs -type nfs -start off
```

Privilege Required

The issuer must be logged in as the local superuser root.

See Also

"klog(1)" on page 153, "knfs(1)" on page 159

Copyright

fs flush

Name

`fs flush` — Forces the Cache Manager to discard a cached file or directory

Synopsis

fs flush [-path <*dir/file path*>+] [-help]

fs flush [-p <*dir/file path*>+] [-h]

Description

The fs flush command removes from the cache all data and status information associated with each specified file or directory. The next time an application requests data from the flushed directory or file, the Cache Manager fetches the most current version from a File Server, along with a new callback (if necessary) and associated status information. This command has no effect on two types of data:

- Data in application program buffers.
- Data that has been changed locally and written to the cache but not yet written to the copy on the file server machine.

To flush all data in the cache that was fetched from the same volume as a specified file or directory, use the fs flushvolume command. To flush a corrupted mount point, use the fs flushmount command.

Options

-path <*dir/file path*>+

Names each file or directory to flush from the cache. If it is a directory, only the directory element itself is flushed, not data cached from files or subdirectories that reside in it. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command flushes from the cache the file `projectnotes` in the current working directory and all data from the subdirectory `plans`:

```
% fs flush -path projectnotes ./plans/*
```

Privilege Required

The issuer must have the `l` (lookup) permission on the ACL of the root directory of the volume that houses the file or directory named by the -path argument, and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_flushmount(1)" on page 75, "fs_flushvolume(1)" on page 77

Copyright

fs flushmount

Name

`fs flushmount` — Forces the Cache Manager to discard a mount point

Synopsis

fs flushmount [-path <*dir/file path*>+] [-help]

fs flushm [-p <*dir/file path*>+] [-h]

Description

The fs flushmount command removes from the cache all information associated with each mount point named by the -path argument. The next time an application accesses the mount point, the Cache Manager fetches the most current version of it from the File Server. Data cached from the associated volume is not affected.

The command's intended use is to discard information about mount points that has become corrupted in the cache. (The Cache Manager periodically refreshes cached mount points, but the only other way to discard them immediately is to reinitialize the Cache Manager by rebooting the machine.) Symptoms of a corrupted mount point included garbled output from the fs lsmount command, and failed attempts to change directory to or list the contents of a mount point.

To flush cached data rather than a mount point, use the fs flush or fs flushvolume command.

Options

-path <*dir/file path*>+

Names each mount point to flush from the cache. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command flushes from the cache the mount point for user `pat`'s home directory:

```
% fs flushm /afs/abc.com/usr/pat
```

Privilege Required

The issuer must have the `l` (lookup) permission on the ACL of the root directory of the volume that houses the file or directory named by the -path argument, and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_flush(1)" on page 73, "fs_flushvolume(1)" on page 77, "fs_lsmount(1)" on page 99

Copyright

fs flushvolume

Name

`fs flushvolume` — Forces the Cache Manager to discard cached data from a volume

Synopsis

fs flushvolume [-path <*dir/file path*>+] [-help]

fs flushv [-p <*dir/file path*>+] [-h]

Description

The fs flushvolume command removes from the cache all data that was fetched from the same volume as each specified directory or file. It does not discard cached status information. The next time an application requests data from a flushed directory or file, the Cache Manager fetches the most current version from a File Server, along with a new callback (if necessary) and associated status information. This command has no effect on two types of data:

- Data in application program buffers.
- Data that has been changed locally and written to the cache but not yet written to the copy on the file server machine.

To discard the data and status information associated with individual files and directories, use the fs flush command. To flush a corrupted mount point, use the fs flushmount command.

Options

-path <*dir/file path*>+

Names a file or directory from each volume for which to discard all cached data. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command flushes from the cache all data fetched from the volume that contains the current working directory:

```
% fs flushvolume
```

Privilege Required

The issuer must have the `l` (lookup) permission on the ACL of the root directory of the volume that houses the file or directory named by the -path argument, and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_flush(1)" on page 73, "fs_flushmount(1)" on page 75

Copyright

fs getcacheparms

Name

`fs getcacheparms` — Displays the current size and usage of the cache

Synopsis

fs getcacheparms [-help]

fs getca [-h]

Description

The fs getcacheparms command displays the current size of the cache (which can be in memory or on disk), and the amount currently in use.

The reported statistics are from kernel memory, so the reported size can differ from the setting specified in the */usr/vice/etc/cacheinfo* file on a machine using a disk cache, if the fs setcachesize command has been used to alter cache size.

Options

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output reports

```
AFS using &lt;amount&gt; of the cache’s available &lt;size&gt; 1K byte blocks.
```

where <amount> is the number of kilobyte blocks currently used to cache data and status information, and <size> is the total current cache size.

Examples

The following example shows the output on a machine with a 25000 kilobyte cache.

```
% fs getcacheparms
AFS using 22876 of the cache’s available 25000 1K byte blocks.
```

Privilege Required

None

See Also

"fs_setcachesize(1)" on page 123

Copyright

fs getcellstatus

Name

`fs getcellstatus` — Reports whether setuid programs are honored in a cell

Synopsis

fs getcellstatus -cell <*cell name*>+ [-help]

fs getce -c <*cell name*>+ [-h]

Description

The fs getcellstatus command reports whether the Cache Manager allows programs fetched from each specified cell to run with setuid permission. To set a cell's setuid status, use the fs setcell command; "fs_setcell(1)" on page 125 fully describes how AFS treats setuid programs.

Options

-cell <*cell name*>+

Names each cell for which to report setuid status. Provide the fully qualified domain name, or a shortened form that disambiguates it from the other cells listed in the local */usr/vice/etc/CellServDB* file.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output reports one of the following two values as appropriate:

```
Cell &lt;cell&gt; status: setuid allowed
Cell &lt;cell&gt; status: no setuid allowed
```

Examples

The following example indicates that programs from the cell `abc.com` are not allowed to run with setuid permission.

```
% fs getcellstatus abc.com
Cell abc.com status: no setuid allowed
```

Privilege Required

None

See Also

"CellServDB(5)" on page 415, "fs_setcell(1)" on page 125

Copyright

fs getclientaddrs

Name

`fs getclientaddrs` — Displays the client interfaces to register

Synopsis

fs getclientaddrs [-help]

fs gc [-h]

fs getcl [-h]

Description

The fs getclientaddrs command displays the IP addresses of the interfaces that the local Cache Manager registers with a File Server when first establishing a connection to it.

The File Server uses the addresses when it initiates a remote procedure call (RPC) to the Cache Manager (as opposed to responding to an RPC sent by the Cache Manager). There are two common circumstances in which the File Server initiates RPCs: when it breaks callbacks and when it pings the client machine to verify that the Cache Manager is still accessible.

If an RPC to that interface fails, the File Server simultaneously sends RPCs to all of the other interfaces in the list, to learn which of them are still available. Whichever interface replies first is the one to which the File Server then sends pings and RPCs to break callbacks.

"fs_setclientaddrs(1)" on page 127 explains how the Cache Manager constructs the list automatically in kernel memory as it initializes, and how to use that command to alter the kernel list after initialization.

Cautions

The File Server uses the list of interfaces displayed by this command only when selecting an alternative interface after a failed attempt to break a callback or ping the Cache Manager. When responding to the Cache Manager's request for file system data, the File Server replies to the interface which the Cache Manager used when sending the request. If the File Server's reply to a data request fails, the file server machine's network routing configuration determines which alternate network routes to the client machine are available for resending the reply.

The displayed list applies to all File Servers to which the Cache Manager connects in the future. It is not practical to register different sets of addresses with different File Servers, because it requires using the fs setclientaddrs command to change the list and then rebooting each relevant File Server immediately.

The displayed list is not necessarily governing the behavior of a given File Server, if an administrator has issued the fs setclientaddrs command since the Cache Manager first contacted that File Server. It determines only which addresses the Cache Manager registers when connecting to File Servers in the future.

The list of interfaces does not influence the Cache Manager's choice of interface when establishing a connection to a File Server.

Options

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output displays the IP address of each interface that the Cache Manager is currently registering with File Server processes that it contacts, with one address per line. The File Server initially uses the first address for breaking callbacks and pinging the Cache Manager, but the ordering of the other interfaces is not meaningful.

Examples

The following example displays the two interfaces that the Cache Manager is registering with File Servers.

```
% fs getclientaddrs
192.12.105.68
192.12.108.84
```

Privilege Required

None

See Also

"fileserver(8)" on page 771, "fs_setclientaddrs(1)" on page 127

Copyright

fs getserverprefs

Name

`fs getserverprefs` — Displays preference ranks for file servers or VL servers

Synopsis

fs getserverprefs [-file <*output to named file*>] [-numeric] [-vlservers] [-help]

fs gets [-f <*output to named file*>] [-n] [-v] [-h]

fs gp [-f <*output to named file*>] [-n] [-v] [-h]

Description

The fs getserverprefs command displays preference ranks for file server machine interfaces (file server machines run the fs process) or, if the -vlserver flag is provided, for Volume Location (VL) Server machines (which run the vlserver process). For file server machines, the Cache Manager tracks up to 15 interfaces per machine and assigns a separate rank to each interface. The ranks indicate the order in which the local Cache Manager attempts to contact the interfaces of machines that are housing a volume when it needs to fetch data from the volume. For VL Server machines, the ranks indicate the order in which the Cache Manager attempts to contact a cell's VL Servers when requesting VLDB information. For both types of rank, lower integer values are more preferred.

The Cache Manager stores ranks in kernel memory. Once set, a rank persists until the machine reboots, or until the fs setserverprefs command is used to change it. The reference page for the fs setserverprefs command explains how the Cache Manager sets default ranks, and how to use that command to change the default values.

Default VL Server ranks range from 10,000 to 10,126, and the Cache Manager assigns them to every machine listed in its copy of the */usr/vice/etc/CellServDB* file. When the Cache Manager needs to fetch VLDB information from a cell, it compares the ranks for the VL Server machines belonging to that cell, and attempts to contact the VL Server with the lowest integer rank. If the Cache Manager cannot reach the VL Server (because of server process, machine or network outage), it tries to contact the VL Server with the next lowest integer rank, and so on. If all of a cell's VL Server machines are unavailable, the Cache Manager cannot fetch data from the cell.

Default file server ranks range from 5,000 to 40,000, excluding the range used for VL Servers (10,000 to 10,126); the maximum possible rank is 65,534. When the Cache Manager needs to fetch data from a volume, it compares the ranks for the interfaces of machines that house the volume, and attempts to contact the interface that has the lowest integer rank. If it cannot reach the fileserver process via that interface (because of server process, machine or network outage), it tries to contact the interface with the next lowest integer rank, and so on. If it cannot reach any of the interfaces for machines that house the volume, it cannot fetch data from the volume.

For both file server machines and VL Server machines, it is possible for a machine or interface in a foreign cell to have the same rank as a machine or interface in the local cell. This does not

present a problem, because the Cache Manager only ever compares ranks for machines belonging to one cell at a time.

Options

-file <*output file*>

Specifies the full pathname of a file to which to write the preference ranks. If the specified file already exists, the command overwrites its contents. If the pathname is invalid, the command fails. If this argument is not provided, the preference ranks appear on the standard output stream.

-numeric

Displays the IP addresses of file server machine interfaces or VL Server machines, rather than their hostnames. If this argument is not provided, the fs command interpreter has the IP addresses translated to hostnames such as `fs1.abc.com`.

-vlservers

Displays preference ranks for VL Server machines rather than file server machine interfaces.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output consists of a separate line for each file server machine interface or VL Server machine, pairing the machine's hostname or IP address with its rank. The Cache Manager stores IP addresses in its kernel list of ranks, but the command by default identifies interfaces by hostname, by calling a translation routine that refers to either the cell's name service (such as the Domain Name Server) or the local host table. If an IP address appears in the output, it is because the translation attempt failed. To bypass the translation step and display IP addresses rather than hostnames, include the -numeric flag. This can significantly speed the production of output.

By default, the command writes to the standard output stream. Use the -file argument to write the output to a file instead.

Examples

The following example displays the local Cache Manager's preference ranks for file server machines. The local machine belongs to the AFS cell named abc.com, and in this example the ranks of file server machines in its local cell are lower than the ranks of file server machines from the foreign cell, `def.com`. It is not possible to translate the IP addresses of two machines on the 138.255 network.

```
% fs getserverprefs
fs2.abc.com           20007
fs3.abc.com           30002
fs1.abc.com           20011
fs4.abc.com           30010
server1.def.com       40002
138.255.33.34         40000
server6.def.com       40012
138.255.33.37         40005
```

The following example shows hows the output displays IP addresses when the -numeric flag is included, and illustrates how network proximity determines default ranks (as described on the fs setserverprefs reference page). The local machine has IP address 192.12.107.210, and the two file server machines on its subnetwork have ranks of 20,007 and 20,011. The two file server machines on a different subnetwork of the local machine's network have higher ranks, 30,002 and 30,010, whereas the ranks of the remaining machines range from 40,000 to 40,012 because they are in a completely different network.

```
% fs getserverprefs -numeric
192.12.107.214          20007
192.12.105.99           30002
192.12.107.212          20011
192.12.105.100          30010
138.255.33.41           40002
138.255.33.34           40000
138.255.33.36           40012
138.255.33.37           40005
```

The example shows how the -vlservers flag displays preference ranks for VL Server machines:

```
% fs getserverprefs -vlservers
fs2.abc.com            10052
fs3.abc.com            10113
fs1.abc.com            10005
```

Privilege Required

None

See Also

"fs_setserverprefs(1)" on page 133

Copyright

fs help

Name

`fs help` — Displays help for fs commands

Synopsis

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

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

Description

The fs help command displays the complete online help entry (short description and syntax statement) for each command operation code specified by the -topic argument. If the -topic argument is omitted, the output includes the first line (name and short description) of the online help entry for every fs command.

To display every fs command whose name or short description includes a specified keyword, use the fs apropos command.

Options

-topic <*help string*>+

Indicates each command for which to display the complete online help entry. Omit the fs part of the command name, providing only the operation code (for example, specify `setacl`, not `fs setacl`). If this argument is omitted, the output briefly describes every fs command.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The first line names the command and briefly describes its function.
- The second line lists aliases for the command, if any.
- The final line, which begins with the string `Usage`, lists the command's options in the prescribed order. Online help entries use the same symbols (for example, brackets) as the reference pages in this document.

Examples

The following command 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 &lt;directory&gt;+
-acl &lt;access list entries&gt;+ [-clear] [-negative] [-help]
```

Privilege Required

None

See Also

"fs(1)" on page 43, "fs_apropos(1)" on page 47

Copyright

fs listacl

Name

`fs listacl` — Displays ACLs

Synopsis

fs listacl [-path <*dir/file path*>+] [-id] [-if] [-help]

fs la [-p <*dir/file path*>+] [-id] [-if] [-h]

fs lista [-p <*dir/file path*>+] [-id] [-if] [-h]

Description

The fs listacl command displays the access control list (ACL) associated with each specified file, directory, or symbolic link. The specified element can reside in the DFS filespace if the issuer is using the AFS/DFS Migration Toolkit Protocol Translator to access DFS data (and DFS does implement per-file ACLs). To display the ACL of the current working directory, omit the -path argument.

To alter an ACL, use the fs setacl command. To copy an ACL from one directory to another, use the fs copyacl command. To remove obsolete entries from an ACL, use the fs cleanacl command.

Cautions

Placing a user or group on the `Negative rights` section of the ACL does not guarantee denial of permissions, if the `Normal rights` section grants the permissions to members of the system:anyuser group. In that case, the user needs only to issue the unlog command to obtain the permissions granted to the system:anyuser group.

Options

-path <*dir/file path*>+

Names each directory or file for which to display the ACL. For AFS files, the output displays the ACL from the file's parent directory; DFS files do have their own ACL. Incomplete pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-id

Displays the Initial Container ACL of each DFS directory. This argument is supported only on DFS directories accessed via the AFS/DFS Migration Toolkit Protocol Translator.

-if

Displays the Initial Object ACL of each DFS directory. This argument is supported only on DFS directories accessed via the AFS/DFS Migration Toolkit Protocol Translator.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of the output for each file, directory, or symbolic link reads as follows:

```
Access list for &lt;directory&gt; is
```

If the issuer used shorthand notation in the pathname, such as the period (`.`) to represent the current current directory, that notation sometimes appears instead of the full pathname of the directory.

Next, the `Normal rights` header precedes a list of users and groups who are granted the indicated permissions, with one pairing of user or group and permissions on each line. If negative permissions have been assigned to any user or group, those entries follow a `Negative rights` header. The format of negative entries is the same as those on the `Normal rights` section of the ACL, but the user or group is denied rather than granted the indicated permissions.

AFS does not implement per-file ACLs, so for a file the command displays the ACL on its directory. The output for a symbolic link displays the ACL that applies to its target file or directory, rather than the ACL on the directory that houses the symbolic link.

The permissions for AFS enable the possessor to perform the indicated action:

a (administer)

Change the entries on the ACL.

d (delete)

Remove files and subdirectories from the directory or move them to other directories.

i (insert)

Add files or subdirectories to the directory by copying, moving or creating.

k (lock)

Set read locks or write locks on the files in the directory.

l (lookup)

List the files and subdirectories in the directory, stat the directory itself, and issue the fs listacl command to examine the directory's ACL.

r (read)

Read the contents of files in the directory; issue the `ls -l` command to stat the elements in the directory.

w (write)

Modify the contents of files in the directory, and issue the UNIX chmod command to change their mode bits

A, B, C, D, E, F, G, H

Have no default meaning to the AFS server processes, but are made available for applications to use in controlling access to the directory's contents in additional ways. The letters must be uppercase.

For DFS files and directories, the permissions are similar, except that the DFS `x` (execute) permission replaces the AFS `l` (lookup) permission, DFS `c` (control) replaces AFS `a` (administer), and there is no DFS equivalent to the AFS `k` (lock) permission. The meanings of the various permissions also differ slightly, and DFS does not implement negative permissions. For a complete description of DFS permissions, see the DFS documentation and the *IBM AFS/DFS Migration Toolkit Administration Guide and Reference*.

Examples

The following command displays the ACL on the home directory of the user `pat` (the current working directory), and on its `private` subdirectory.

```
% fs listacl -path . 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

If the -path argument names an AFS directory, the issuer must have the `l` (lookup) permission on its ACL and the ACL for every directory that precedes it in the pathname.

If the -path argument names an AFS file, the issuer must have the `l` (lookup) and `r` (read) permissions on the ACL of the file's directory, and the l permission on the ACL of each directory that precedes it in the pathname.

If the -path argument names a DFS directory or file, the issuer must have the `x` (execute) permission on its ACL and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_cleanacl(1)" on page 55, "fs_copyacl(1)" on page 57, "fs_setacl(1)" on page 117

IBM AFS/DFS Migration Toolkit Administration Guide and Reference

Copyright

fs listcells

Name

`fs listcells` — Displays the database server machines known to the Cache Manager

Synopsis

fs listcells [-numeric] [-help]

fs listc [-n] [-h]

Description

The fs listcells command formats and displays the list of the database server machines that the Cache Manager stores in kernel memory for its home cell and foreign cells.

At each reboot of the client machine, the Cache Manager copies the contents of */usr/vice/etc/CellServDB* into kernel memory. To modify the list between reboots, use the fs newcell command.

Options

-numeric

Displays each database server machine's IP address rather than hostname.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output includes a line for each cell included in the Cache Manager's kernel memory list, in the following format:

```
Cell &lt;cell&gt; on hosts &lt;database server machines&gt;
```

The Cache Manager stores IP addresses, but by default has them translated to hostnames before reporting them, by passing them to the cell's name service (such as the Domain Name Service or a local host table). The name service sometimes returns hostnames in uppercase letters, or an IP address if it cannot resolve a name.

Using the -numeric flag bypasses the translation to hostnames, which can result in significantly faster production of output. The output includes IP addresses only.

Examples

The following example shows output for several cells as illustrations of the different formats for machine names:

```
% fs listcells
Cell abc.com on hosts fs1.abc.com fs2.abc.com fs3.abc.com
Cell stateu.edu on hosts DB1.FS.STATEU.EDU
   DB2.FS.STATEU.EDU DB3.FS.STATEU.EDU
Cell def.gov on hosts 138.255.0.2 sv3.def.gov
```

Privilege Required

None

See Also

"CellServDB(5)" on page 415, "fs_newcell(1)" on page 109

Copyright

fs listquota

Name

`fs listquota` — Displays quota information for a volume

Synopsis

fs listquota [-path <*dir/file path*>+] [-help]

fs listq [-p <*dir/file path*>+] [-h]

fs lq [-p <*dir/file path*>+] [-h]

Description

The fs listquota command displays information about the volume containing each specified directory or file (its name, quota, and amount of disk space used), along with an indicator of the percentage of space used on the host partition.

To display more information about the host partition, use the fs examine command.

To set volume quota, use the fs setquota or fs setvol command.

Options

-path <*dir/file path*>+

Names a file or directory that resides in the volume about which to produce output. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output displays information about the volume that houses each specified directory or file, in a tabular format that uses the following headers:

Volume Name

The name of the volume.

Quota

The volume's quota in kilobytes, or the string `no limit` to indicate an unlimited quota.

Used

The number of kilobytes of quota used.

% Used

The percentage of the volume's quota that is used (the `Used` statistic divided by the `Quota` statistic, times 100).

Partition

The percentage of space used on the partition that houses the volume. Although not directly related to how much of the user's quota is used, it is reported because a full partition can cause writing of data back to the volume to fail even when the volume has not reached its quota.

Examples

The following example shows the output for the volume `user.smith`:

```
% fs listquota -path /afs/abc.com/usr/smith
Volume Name     Quota    Used    % Used   Partition
user.smith      15000    5071       34%         86%
```

Privilege Required

The issuer must have the `l` (lookup) permission on the ACL of the root directory of the volume that houses the file or directory named by the -path argument, and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_diskfree(1)" on page 61, "fs_examine(1)" on page 65, "fs_quota(1)" on page 113, "fs_setquota(1)" on page 131, "fs_setvol(1)" on page 139

Copyright

fs lsmount

Name

`fs lsmount` — Reports the volume for which a directory is the mount point.

Synopsis

fs lsmount -dir <*directory*>+ [-help]

fs ls -d <*directory*>+ [-h]

Description

The fs lsmount command reports the volume for which each specified directory is a mount point, or indicates with an error message that a directory is not a mount point or is not in AFS.

To create a mount point, use the fs mkmount command. To remove one, use the fs rmmount command.

Options

-dir <*directory*>+

Names the directory that serves as a mount point for a volume. The last element in the pathname provided must be an actual name, not a shorthand notation such as one or two periods (`.` or `..`).

-help

Prints the online help for this command. All other valid options are ignored.

Output

If the specified directory is a mount point, the output is of the following form:

```
'&lt;directory&gt;' is a mount point for volume '&lt;volume name&gt;'
```

where

- A number sign (#) precedes the <volume name> string for a regular mount point.
- A percent sign (%) precedes the <volume name> string for a read / write mount point.
- A cell name and colon (:) follow the number or percent sign and precede the <volume name> string for a cellular mount point.

The fs mkmount reference page explains how the Cache Manager interprets each of the three types of mount points.

If the directory is a symbolic link to a mount point, the output is of the form:

```
'&lt;directory&gt;' is a symbolic link, leading to a mount point for volume
'&lt;volume name&gt;'
```

If the directory is not a mount point or is not in AFS, the output reads:

```
'&lt;directory&gt;' is not a mount point.
```

If the output is garbled, it is possible that the mount point has become corrupted in the local AFS client cache. Use the fs flushmount command to discard it, which forces the Cache Manager to refetch the mount point.

Examples

The following example shows the mount point for the home directory of user `smith`:

```
% fs lsmount /afs/abc.com/usr/smith
'/afs/abc.com/usr/smith' is a mount point for volume '#user.smith'
```

The following example shows both the regular and read / write mount points for the ABC Corporation cell's `root.cell` volume.

```
% fs lsmount /afs/abc.com
'/afs/abc.com' is a mount point for volume '#root.cell'
```

```
% fs lsmount /afs/.abc.com
'/afs/.abc.com' is a mount point for volume '%root.cell'
```

The following example shows a cellular mount point: the State University cell's `root.cell` volume as mounted in the ABC Corporation cell's tree.

```
% fs lsmount /afs/stateu.edu
'/afs/stateu.edu' is a mount point for volume '#stateu.edu:root.cell'
```

Privilege Required

The issuer must have the `l` (lookup) permission on the ACL of the root directory of the volume that houses the file or directory named by the -dir argument, and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_flushmount(1)" on page 75, "fs_mkmount(1)" on page 105, "fs_rmmount(1)" on page 115

Copyright

fs messages

Name

`fs messages` — Sets whether the Cache Manager writes log messages

Synopsis

fs messages [-show (user | console | all | none)] [-help]

fs me [-s (user | console | all | none)] [-h]

Description

The fs messages command controls whether the Cache Manager displays status and warning messages on user screens, the client machine console, on both, or on neither.

There are two types of Cache Manager messages:

- User messages provide user-level status and warning information, and the Cache Manager directs them to user screens.
- Console messages provide system-level status and warning information, and the Cache Manager directs them to the client machine's designated console.

Disabling messaging completely is not recommended, because the messages provide useful status and warning information.

Options

-show (user | console | all | none)

Specifies the types of messages to display. Choose one of the following values:

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 (the default if the -show argument is omitted).

none

Do not send any messages to user screens or the console.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command instructs the Cache Manager to display both types of messages:

```
% fs messages -show all
```

Privilege Required

The issuer must be logged in as the local superuser root.

See Also

"afsd(8)" on page 523

Copyright

fs mkmount

Name

`fs mkmount` — Creates a mount point for a volume

Synopsis

fs mkmount -dir <*directory*> -vol <*volume name*> [-cell <*cell name*>] [-rw] [-fast] [-help]

fs mk -d <*directory*> -v <*volume name*> [-c <*cell name*>] [-r] [-f] [-h]

Description

The fs mkmount command creates a mount point for the volume named by the -vol argument at the location in the AFS file space specified by the -dir argument. The mount point looks like a standard directory element, and serves as the volume's root directory, but is actually a special file system object that refers to an AFS volume. When the Cache Manager first encounters a given mount point during pathname traversal, it contacts the VL Server to learn which file server machines house the indicated volume, then fetches a copy of the volume's root directory from the appropriate file server machine.

It is possible, although not recommended, to create more than one mount point to a volume. The Cache Manager can become confused if a volume is mounted in two places along the same path through the filespace.

The Cache Manager observes three basic rules as it traverses the AFS filespace and encounters mount points:

Rule 1: Access Backup and Read-only Volumes When Specified

When the Cache Manager encounters a mount point that specifies a volume with either a `.readonly` or a `.backup` extension, it accesses that type of volume only. If a mount point does not have either a `.backup` or `.readonly` extension, the Cache Manager uses Rules 2 and 3.

For example, the Cache Manager never accesses the read / write version of a volume if the mount point names the backup version. If the specified version is inaccessible, the Cache Manager reports an error.

Rule 2: Follow the Read-only Path When Possible

If a mount point resides in a read-only volume and the volume that it references is replicated, the Cache Manager attempts to access a read-only copy of the volume; if the referenced volume is not replicated, the Cache Manager accesses the read / write copy. The Cache Manager is thus said to prefer a *read-only path* through the filespace, accessing read-only volumes when they are available.

The Cache Manager starts on the read-only path in the first place because it always accesses a read-only copy of the root.afs volume if it exists; the volume is mounted at the root of a cell's AFS filespace (named `/afs` by convention). That is, if the `root.afs` volume

is replicated, the Cache Manager attempts to access a read-only copy of it rather than the read / write copy. This rule then keeps the Cache Manager on a read-only path as long as each successive volume is replicated. The implication is that both the `root.afs` and `root.cell` volumes must be replicated for the Cache Manager to access replicated volumes mounted below them in the AFS filespace. The volumes are conventionally mounted at the */afs* and */afs/cellname* directories, respectively.

Rule 3: Once on a Read / write Path, Stay There

If a mount point resides in a read / write volume and the volume name does not have a `.readonly` or a `.backup` extension, the Cache Manager attempts to access only the a read / write version of the volume. The access attempt fails with an error if the read / write version is inaccessible, even if a read-only version is accessible. In this situation the Cache Manager is said to be on a *read/write path* and cannot switch back to the read-only path unless mount point explicitly names a volume with a `.readonly` extension. (Cellular mount points are an important exception to this rule, as explained in the following discussion.

There are three types of mount points, each appropriate for a different purpose because of the manner in which the Cache Manager interprets them.

- When the Cache Manager crosses a *regular* mount point, it obeys all three of the mount point traversal rules previously described. To create a regular mount point, include only the required -dir and -vol arguments to the fs mkmount command.
- When the Cache Manager crosses a *read/write* mount point, it attempts to access only the volume version named in the mount point. If the volume name is the base (read / write) form, without a `.readonly` or `.backup` extension, the Cache Manager accesses the read / write version of the volume, even if it is replicated. In other words, the Cache Manager disregards the second mount point traversal rule when crossing a read / write mount point: it switches to the read / write path through the filespace.

  To create a read / write mount point, include the -rw flag on the fs mkmount command. It is conventional to create only one read / write mount point in a cell's filespace, using it to mount the cell's `root.cell` volume just below the AFS filespace root (by convention, */afs/.cellname*). See the *IBM AFS Quick Beginnings* for instructions and the chapter about volume management in the *IBM AFS Administration Guide* for further discussion.

  Creating a read / write mount point for a read-only or backup volume is acceptable, but unnecessary. The first rule of mount point traversal already specifies that the Cache Manager accesses them if the volume name in a regular mount point has a `.readonly` or `.backup` extension.
- When the Cache Manager crosses a *cellular* mount point, it accesses the indicated volume in the specified cell, which is normally a foreign cell. (If the mount point does not name a cell along with the volume, the Cache Manager accesses the volume in the cell where the mount point resides.) The Cache Manager disregards the third mount point traversal rule when crossing a regular cellular mount point: it accesses a read-only version of the volume if it is replicated, even if the volume that houses the mount point is read / write. Switching to the read-only path in this way is designed to avoid imposing undue load on the file server machines in foreign cells.

To create a regular cellular mount point, include the -cell argument on the fs mkmount command. It is conventional to create cellular mount points only at the second level in a cell's filespace, using them to mount foreign cells' root.cell volumes just below the AFS filespace root (by convention, at */afs/foreign_cellname*). The mount point enables local users to access the foreign cell's filespace, assuming they have the necessary permissions on the ACL of the volume's root directory and that there is an entry for the foreign cell in each local client machine's */usr/vice/etc/CellServDB* file. In the output of the fs lsmount command, the cell name and a colon (:) appear between the initial number sign and the volume name in a regular cellular mount point name.

Options

-dir <*directory*>+

Names the directory to create as a mount point. The directory must not already exist. Relative pathnames are interpreted with respect to the current working directory.

Specify the read / write path to the directory, to avoid the failure that results from attempting to create a new mount point in a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). For further discussion of the concept of read / write and read-only paths through the filespace, see "DESCRIPTION" on page .

-vol <*volume name*>

Specifies the name or volume ID number of the volume to mount. If appropriate, add the `.readonly` or `.backup` extension to the name, or specify the appropriate volume ID number.

-cell <*cell name*>

Names the cell in which the volume resides (creates a cellular mount point). Provide the fully qualified domain name, or a shortened form that disambiguates it from the other cells listed in the local */usr/vice/etc/CellServDB* file.

If this argument is omitted, no cell indicator appears in the mount point. When the Cache Manager interprets it, it assumes that the volume named in the mount point resides in the same cell as the volume that houses the mount point.

-rw

Creates a read / write mount point. Omit this flag to create a regular mount point.

-fast

Prevents the Volume Location (VL) Server from checking that the volume has a VLDB entry and printing a warning message if it does not. Whether or not this flag is included, the File Server creates the mount point even when the volume has no VLDB entry.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command creates a regular mount point, mounting the volume `user.smith` at */afs/abc.com/usr/smith*:

```
% cd /afs/abc.com/usr
% fs mkmount -dir smith -vol user.smith
```

The following commands create a read/write mount point and a regular mount point for the ABC Corporation cell's `root.cell` volume in that cell's file tree. The second command follows the convention of putting a period at the beginning of the read/write mount point's name.

```
% fs mkmount -dir /afs/abc.com -vol root.cell
% fs mkmount -dir /afs/.abc.com -vol root.cell -rw
```

The following command mounts the State University cell's `root.cell` volume in the ABC Corporation cell's file tree, creating a regular cellular mount point called */afs/stateu.edu*. When a ABC Corporation Cache Manager encounters this mount point, it crosses into the State University cell on a read-only path.

```
% fs mkmount -dir /afs/stateu.edu -vol root.cell -c stateu.edu
```

Privilege Required

The issuer must have the `i` (insert) and `a` (administer) permissions on the ACL of the directory that is to house the mount point.

See Also

"CellServDB(5)" on page 415, "fs_lsmount(1)" on page 99, "fs_rmmount(1)" on page 115

Copyright

fs newcell

Name

`fs newcell` — Changes the kernel-resident list of a cell's database servers

Synopsis

fs newcell -name <*cell name*> -servers <*primary servers*>+ [-linkedcell <*linked cell name*>] [-help]

fs n -n <*cell name*> -s <*primary servers*>+ [-l <*linked cell name*>] [-h]

Description

The fs newcell command removes the Cache Manager's kernel-resident list of database server machines for the cell specified by the -name argument and replaces it with the database server machines named by the -servers argument.

Each time the machine reboots, the Cache Manager constructs the kernel list of cells and database server machines by reading the local */usr/vice/etc/CellServDB* file. This command does not change the *CellServDB* file, so any changes made with it persist only until the next reboot, unless the issuer also edits the file. The output of the fs listcells command reflects changes made with this command, because that command consults the kernel-resident list rather than the *CellServDB* file.

This command can introduce a completely new cell into the kernel-resident list, but cannot make a cell inaccessible (it is not possible to remove a cell's entry from the kernel-resident list by providing no values for the -server argument). To make a cell inaccessible, remove its entry from the *CellServDB* file and reboot the machine.

If the -name argument names a DCE cell, then the -servers argument names DFS Fileset Location (FL) Server machines. The -linkedcell argument specifies the name of the AFS cell to link to a DCE cell for the purpose of DFS fileset location. Refer to the *IBM AFS/DFS Migration Toolkit Administration Guide and Reference* for more information on linking AFS clients to DCE cells using this command or by editing the */usr/vice/etc/CellServDB* file.

Cautions

Some commands, such as the klog command, work correctly only when the information is accurate for a cell in both the *CellServDB* file and the kernel-resident list.

Options

-name <*cell name*>

Specifies the fully-qualified cell name of the AFS or DCE cell.

-servers <*primary servers*>+

Specifies the fully-qualified hostnames of all AFS database server machines or DFS Fileset Location (FL) Server machines for the cell named by the -name argument. If FL Server machines are specified, the local machine must be running the AFS/DFS Migration Toolkit Protocol Translator.

-linkedcell <*linked cell name*>

Specifies the name of the AFS cell to link to a DCE cell for the purpose of DFS fileset location.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example changes the machine's kernel-resident list of database server machines for the ABC Corporation cell to include the machines `db1.abc.com` and `db2.abc.com`:

```
% fs newcell -name abc.com -servers db1.abc.com db2.abc.com
```

The following example links the DCE cell `dce.abc.com` to the AFS cell `abc.com`. The AFS client contacts the Fileset Location (FL) servers `db1.dce.abc.com` and `db2.dce.abc.com` for fileset location information as it interprets a DFS pathname.

```
% fs newcell -name dce.abc.com \
    -servers db1.dce.abc.com db2.dce.abc.com \
    -linkedcell abc.com
```

Privilege Required

The issuer must be logged in as the local superuser root.

See Also

"CellServDB(5)" on page 415, "fs_listcells(1)" on page 95

IBM AFS/DFS Migration Toolkit Administration Guide and Reference

IBM AFS/DFS Migration Toolkit Administration Installation and Configuration Guide

Copyright

fs quota

Name

`fs quota` — Displays the quota used in the volume containing a directory or file

Synopsis

fs quota [-path <*dir/file path*>+] [-help]

fs q [-p <*dir/file path*>+] [-h]

Description

The fs quota command displays the percent of quota consumed in the volume that contains each specified directory or file.

To display more detailed information about the volume and the partition it resides on, use the fs examine and fs listquota commands.

To set volume quota, use the fs setquota or fs setvol command.

Options

-path <*dir/file path*>

Names each file or directory for which to display the quota consumed in its parent volume. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output reports the percent of volume quota used, in the following format:

```
&lt;percent&gt;% of quota used.
```

Examples

The following command lists the percent quota used of the volume housing the current working directory:

```
% fs quota
17% of quota used.
```

The following command lists the percent quota used of both the volume housing the current working directory's parent directory and the volume housing the directory */afs/abc.com/usr/smith*:

```
% fs quota -path .. /afs/abc.com/usr/smith
43% of quota used.
92% of quota used.
```

Privilege Required

The issuer must have the `l` (lookup) permission on the ACL of the root directory of the volume that houses the file or directory named by the -path argument, and on the ACL of each directory that precedes it in the pathname.

See Also

"fs_examine(1)" on page 65, "fs_listquota(1)" on page 97, "fs_setquota(1)" on page 131, "fs_setvol(1)" on page 139

Copyright

fs rmmount

Name

`fs rmmount` — Removes a mount point

Synopsis

fs rmmount -dir <*directory*>+ [-help]

fs rm -d <*directory*>+ [-h]

Description

The fs rmmount command removes the mount point named by the -dir argument from the file system. The corresponding volume remains on its host partition or partitions, but is inaccessible if there are no other mount points for it.

Options

-dir <*directory*>+

Names the mount point to delete from the file system. The last element in the pathname must be an actual name, not a shorthand notation such as "dot" (`.`) or "dot dot" (`..`).

Specify the read / write path to the directory, to avoid the failure that results from attempting to delete a mount point from a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). For further discussion of the concept of read / write and read-only paths through the filespace, see the fs mkmount reference page.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command removes the mount points *jones* and *terry* from the current working directory (the */afs/abc.com/usr* directory).

```
% fs rmmount jones terry
```

Privilege Required

The issuer must have the `d` (delete) permission on the ACL of the directory that houses each mount point.

See Also

"fs_lsmount(1)" on page 99, "fs_mkmount(1)" on page 105

Copyright

fs setacl

Name

`fs setacl` — Sets the ACL for a directory

Synopsis

fs setacl -dir <*directory*>+ -acl <*access list entries*>+ [-clear] [-negative] [-id] [-if] [-help]

fs sa -d <*directory*>+ -a <*access list entries*>+ [-c] [-n] [-id] [-if] [-h]

fs seta -d <*directory*>+ -a <*access list entries*>+ [-c] [-n] [-id] [-if] [-h]

Description

The fs setacl command adds the access control list (ACL) entries specified with the -acl argument to the ACL of each directory named by the -dir argument.

If the -dir argument designates a pathname in DFS filespace (accessed via the AFS/DFS Migration Toolkit Protocol Translator), it can be a file as well as a directory. The ACL must already include an entry for `mask_obj`, however. For more details, refer to the *IBM AFS/DFS Migration Toolkit Administration Guide and Reference*.

Only user and group entries are acceptable values for the -acl argument. Do not place machine entries (IP addresses) directly on an ACL; instead, make the machine entry a group member and place the group on the ACL.

To completely erase the existing ACL before adding the new entries, provide the -clear flag. To add the specified entries to the `Negative rights` section of the ACL (deny rights to specified users or groups), provide the -negative flag.

To display an ACL, use the fs listacl command. To copy an ACL from one directory to another, use the fs copyacl command.

Cautions

If the ACL already grants certain permissions to a user or group, the permissions specified with the fs setacl command replace the existing permissions, rather than being added to them.

Setting negative permissions is generally unnecessary and not recommended. Simply omitting a user or group from the `Normal rights` section of the ACL is normally adequate to prevent access. In particular, note that it is futile to deny permissions that are granted to members of the system:anyuser group on the same ACL; the user needs only to issue the unlog command to receive the denied permissions.

When including the -clear option, be sure to reinstate an entry for each directory's owner that includes at least the `l` (lookup) permission. Without that permission, it is impossible to resolve the "dot" (`.`) and "dot dot" (`..`) shorthand from within the directory. (The directory's owner does implicitly have the `a` (administer) permission even on a cleared ACL, but must know to use it to add other permissions.)

Options

-dir <*directory*>+

Names each AFS directory, or DFS directory or file, for which the set the ACL. Partial pathnames are interpreted relative to the current working directory.

Specify the read / write path to each directory (or DFS file), to avoid the failure that results from attempting to change a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, `/afs/.abc.com`). For further discussion of the concept of read / write and read-only paths through the filespace, see the fs mkmount reference page.

-acl <*access list entries*>+

Defines a list of one or more ACL entries, each a pair that names:

- A user name or group name as listed in the Protection Database.
- One or more ACL permissions, indicated either by combining the individual letters or by one of the four acceptable shorthand words.

in that order, separated by a space (thus every instance of this argument has two parts). The accepted AFS abbreviations and shorthand words, and the meaning of each, are as follows:

a (administer)

Change the entries on the ACL.

d (delete)

Remove files and subdirectories from the directory or move them to other directories.

i (insert)

Add files or subdirectories to the directory by copying, moving or creating.

k (lock)

Set read locks or write locks on the files in the directory.

l (lookup)

List the files and subdirectories in the directory, stat the directory itself, and issue the fs listacl command to examine the directory's ACL.

r (read)

Read the contents of files in the directory; issue the `ls -l` command to stat the elements in the directory.

w (write)

Modify the contents of files in the directory, and issue the UNIX chmod command to change their mode bits.

A, B, C, D, E, F, G, H

Have no default meaning to the AFS server processes, but are made available for applications to use in controlling access to the directory's contents in additional ways. The letters must be uppercase.

all

Equals all seven permissions (`rlidwka`).

none

No permissions. Removes the user / group from the ACL, but does not guarantee they have no permissions if they belong to groups that remain on the ACL.

read

Equals the `r` (read) and `l` (lookup) permissions.

write

Equals all permissions except `a` (administer), that is, `rlidwk`.

It is acceptable to mix entries that combine the individual letters with entries that use the shorthand words, but not use both types of notation within an individual pairing of user or group and permissions.

To learn the proper format and acceptable values for DFS ACL entries, see the *IBM AFS/DFS Migration Toolkit Administration Guide and Reference*.

-clear

Removes all existing entries on each ACL before adding the entries specified with the -acl argument.

-negative

Places the specified ACL entries in the `Negative rights` section of each ACL, explicitly denying the rights to the user or group, even if entries on the accompanying `Normal rights` section of the ACL grant them permissions.

This argument is not supported for DFS files or directories, because DFS does not implement negative ACL permissions.

-id

Places the ACL entries on the Initial Container ACL of each DFS directory, which are the only file system objects for which this flag is supported.

-if

Places the ACL entries on the Initial Object ACL of each DFS directory, which are the only file system objects for which this flag is supported.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example adds two entries to the `Normal rights` section of the current working directory's ACL: the first entry grants `r` (read) and `l` (lookup) permissions to the group pat:friends, while the other (using the `write` shorthand) gives all permissions except `a` (administer) to the user `smith`.

```
% fs setacl -dir . -acl pat:friends rl smith write


% fs listacl -path .
Access list for . is
Normal rights:
   pat:friends rl
   smith rlidwk
```

The following example includes the -clear flag, which removes the existing permissions (as displayed with the fs listacl command) from the current working directory's *`reports`* subdirectory and replaces them with a new set.

```
% fs listacl -dir reports
Access list for reports is
Normal rights:
   system:authuser rl
   pat:friends rlid
   smith rlidwk
   pat rlidwka
Negative rights:
   terry rl


% fs setacl -clear -dir reports -acl pat all smith write system:anyuser rl


% fs listacl -dir reports
Access list for reports is
Normal rights:
   system:anyuser rl
   smith rlidwk
   pat rlidwka
```

The following example use the -dir and -acl switches because it sets the ACL for more than one directory (both the current working directory and its *`public`* subdirectory).

```
% fs setacl -dir . public -acl pat:friends rli


% fs listacl -path . public
Access list for . is
Normal rights:
   pat rlidwka
   pat:friends rli
```

```
Access list for public is
Normal rights:
   pat rlidwka
   pat:friends rli
```

Privilege Required

The issuer must have the a (administer) permission on the directory's ACL; the directory's owner and the members of the system:administrators group have the right implicitly, even if it does not appear on the ACL.

See Also

"fs_copyacl(1)" on page 57, "fs_listacl(1)" on page 91, "fs_mkmount(1)" on page 105

IBM AFS/DFS Migration Toolkit Administration Guide and Reference

Copyright

fs setcachesize

Name

`fs setcachesize` — Sets the size of the disk cache

Synopsis

fs setcachesize [-blocks <*size in 1K byte blocks (0* = reset)>>] [-reset] [-help]

fs setca [-b <*size in 1K byte blocks (0* = reset)>>] [-r] [-h]

fs cachesize [-b <*size in 1K byte blocks (0* = reset)>>] [-r] [-h]

fs ca [-b <*size in 1K byte blocks (0* = reset)>>] [-r] [-h]

Description

The fs setcachesize command changes the number of kilobyte blocks of local disk space available to the Cache Manager for its data cache, on machines that use a disk cache. The command is not operative on machines that use a memory cache.

To return the cache size to the default value specified in the third field of the local */usr/vice/etc/cacheinfo* file, provide a value of `0` to the -blocks argument.

To return the cache size to the value set when the machine was last rebooted, use the -reset flag instead of the -blocks argument. This is normally the amount specified in the *cacheinfo* file, unless the -blocks argument was included on the afsd command to override the cacheinfo value.

To display the current cache size and amount of cache in use, for both disk and memory caches, use the fs getcacheparms command.

Cautions

This command is not operative on machines using a memory cache, and results in an error message. To change memory cache size, edit the cacheinfo file and reboot, or reboot and provide the -blocks argument to the afsd command.

On machines using a disk cache, do not set the cache size to exceed 85% to 90% of the actual disk space available for the cache directory. The cache implementation itself requires a small amount of space on the partition.

Options

-blocks <*size in 1K byte blocks*>

Specifies the number of one-kilobyte blocks of disk space available for the Cache Manager to devote to the cache. Provide a value of `0` to set cache size to the default specified in the *cacheinfo* file.

-reset

Returns the cache size to the value set when the machine was last booted. This agrees with the value in the *cacheinfo* file unless the -blocks argument was used on the afsd command.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command sets the disk cache size to 25000 kilobyte blocks.

```
% fs setcachesize -blocks 25000
```

Both of the following commands reset the disk cache size to the value in the *cacheinfo* file, assuming that the -blocks argument to the afsd command was not used.

```
% fs setcachesize -blocks 0
% fs setcachesize -reset
```

Privilege Required

The issuer must be logged in as the local superuser root.

See Also

"cacheinfo(5)" on page 467, "afsd(8)" on page 523, "fs_getcacheparms(1)" on page 79

Copyright

fs setcell

Name

`fs setcell` — Configures permissions for setuid programs from specified cells

Synopsis

fs setcell -cell <*cell name*>+ [-suid] [-nosuid] [-help]

fs setce -c <*cell name*>+ [-s] [-n] [-h]

Description

The fs setcell command sets whether the Cache Manager allows programs (and other executable files) from each cell named by the -cell argument to run with setuid permission. By default, the Cache Manager allows programs from its home cell to run with setuid permission, but not programs from any foreign cells. A program belongs to the same cell as the file server machine that houses the volume in which the program's binary file resides, as specified in the file server machine's */usr/afs/etc/ThisCell* file. The Cache Manager determines its own home cell by reading the */usr/vice/etc/ThisCell* file at initialization.

To enable programs from each specified cell to run with setuid permission, include the -suid flag. To prohibit programs from running with setuid permission, include the -nosuid flag, or omit both flags.

The fs setcell command directly alters a cell's setuid status as recorded in kernel memory, so rebooting the machine is unnecessary. However, non-default settings do not persist across reboots of the machine unless the appropriate fs setcell command appears in the machine's AFS initialization file.

To display a cell's setuid status, issue the fs getcellstatus command.

Cautions

AFS does not recognize effective UID: if a setuid program accesses AFS files and directories, it does so using the current AFS identity of the AFS user who initialized the program, not of the program's owner. Only the local file system recognizes effective UID.

Only members of the system:administrators group can turn on the setuid mode bit on an AFS file or directory.

When the setuid mode bit is turned on, the UNIX `ls -l` command displays the third user mode bit as an `s` instead of an `x`. However, the `s` does not appear on an AFS file or directory unless setuid permission is enabled for the cell in which the file resides.

Options

-cell <*cell name*>+

Names each cell for which to set setuid status. Provide the fully qualified domain name, or a shortened form that disambiguates it from the other cells listed in the local */usr/vice/etc/CellServDB* file.

-suid

Allows programs from each specified cell to run with setuid privilege. Provide it or the -nosuid flag, or omit both flags to disallow programs from running with setuid privilege.

-nosuid

Prevents programs from each specified cell from running with setuid privilege. Provide it or the -suid flag, or omit both flags to disallow programs form running with setuid privilege.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command enables executable files from the State University cell to run with setuid privilege on the local machine:

```
% fs setcell -cell stateu.edu -suid
```

Privilege Required

The issuer must be logged in as the local superuser root.

See Also

"fs_getcellstatus(1)" on page 81

Copyright

fs setclientaddrs

Name

`fs setclientaddrs` — Sets the client interfaces to register with the File Server

Synopsis

fs setclientaddrs [-address <*client network interfaces*>+] [-help]

fs setcl [-a <*client network interfaces*>+] [-h]

fs sc [-a <*client network interfaces*>+] [-h]

Description

The fs setclientaddrs command defines the IP addresses of the interfaces that the local Cache Manager registers with a File Server when first establishing a connection to it.

The File Server uses the addresses when it initiates a remote procedure call (RPC) to the Cache Manager (as opposed to responding to an RPC sent by the Cache Manager). There are two common circumstances in which the File Server initiates RPCs: when it breaks callbacks and when it pings the client machine to verify that the Cache Manager is still accessible.

The list of interfaces specified with this command replaces the list that the Cache Manager constructs and records in kernel memory as it initializes. At that time, if the file */usr/vice/etc/NetInfo* exists on the client machine's local disk, the Cache Manager uses its contents as the basis for the list of interfaces addresses. If the file does not exist, the Cache Manager instead uses the network interfaces configured with the operating system. It then removes from the list any address included in the local */usr/vice/etc/NetRestrict* file. It records the final list in kernel memory. (An administrator must create the *NetInfo* and *NetRestrict* files; there are no default versions of them.)

If an RPC to that interface fails, the File Server simultaneously sends RPCs to all of the other interfaces in the list, to learn which of them are still available. Whichever interface replies first is the one to which the File Server then sends pings and RPCs to break callbacks.

To list the interfaces that the Cache Manager is currently registering with File Servers, use the fs getclientaddrs command.

Cautions

The list specified with this command persists in kernel memory only until the client machine reboots. To preserve it across reboots, either list the interfaces in the local */usr/vice/etc/NetInfo* file, or place the appropriate fs setclientaddrs command in the machine's AFS initialization script.

Changes made with this command do not propagate automatically to File Servers to which the Cache Manager has already established a connection. To force such File Servers to use the revised list, either reboot each file server machine, or change the *NetInfo* file and reboot the client machine.

The fs command interpreter verifies that each of the addresses specified as a value for the -address argument is actually configured with the operating system on the client machine. If it is not, the command fails with an error message that marks the address as a `Nonexistent interface`.

Options

-address <*client network interfaces*>+

Specifies each IP address to place in the list of interfaces, in dotted decimal format. Hostnames are not acceptable. Separate each address with one or more spaces.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The message

```
Adding &lt;interface&gt;
```

confirms that each new interface was added to the Cache Manager's list. The address appears in hexadecimal format to match the notation used in the File Server log, */usr/afs/logs/FileLog*.

Examples

The following example sets the two interfaces that the Cache Manager registers with File Servers.

```
% fs setclientaddrs 191.255.105.68 191.255.108.84
Adding 0xbfff6944
Adding 0xbfff6c54
```

Privilege Required

The issuer must be logged in as the local superuser root.

See Also

"NetInfo(5)" on page 425, "NetRestrict(5)" on page 427, "fileserver(8)" on page 771, "fs_getclientaddrs(1)" on page 83

Copyright

fs setquota

Name

`fs setquota` — Sets the quota for the volume containing a file or directory

Synopsis

fs setquota [-path <*dir/file path*>] -max <*max quota in kbytes*> [-help]

fs setq [-p <*dir/file path*>] -m <*max quota in kbytes*> [-h]

fs sq [-p <*dir/file path*>] -m <*max quota in kbytes*> [-h]

Description

The fs setquota command sets the quota (maximum possible size) of the read / write volume that contains the directory or file named by the -path argument.

To set the quota on multiple volumes at the same time, use the fs setvol command.

To display a volume's quota, use the fs examine, fs listquota, or fs quota command.

Options

-path <*dir/file path*>

Names the directory or file for which to set the host volume's quota. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

Specify the read / write path to the file or directory, to avoid the failure that results from attempting to change a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, `/afs/.abc.com`). For further discussion of the concept of read / write and read-only paths through the filespace, see the fs mkmount reference page.

-max <*max quota in kbytes*>

Sets the maximum amount of file server disk space the volume can occupy. Specify the number of one-kilobyte blocks as a positive integer (`1024` is one megabyte). A value of `0` sets an unlimited quota, but the size of the disk partition that houses the volume places an absolute limit on the volume's size.

If the -path argument is omitted (to set the quota of the volume housing the current working directory), the -max switch must be included with this argument.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command imposes a maximum quota of 3000 kilobytes on the volume that houses the */afs/abc.com/usr/smith* directory:

```
% fs setquota -path /afs/abc.com/usr/smith -max 3000
```

Privilege Required

The issuer must belong to the system:administrators group.

See Also

"fs_examine(1)" on page 65, "fs_listquota(1)" on page 97, "fs_quota(1)" on page 113, "fs_mkmount(1)" on page 105, "fs_setvol(1)" on page 139

Copyright

fs setserverprefs

Name

`fs setserverprefs` — Sets the preference ranks for file servers or VL servers

Synopsis

fs setserverprefs [-servers <*fileserver names and ranks*>+] [-vlservers <*VL server names and ranks*>+] [-file <*input from named file*>] [-stdin] [-help]

fs sets [-se <*fileserver names and ranks*>+] [-vl <*VL server names and ranks*>+] [-f <*input from named file*>] [-st] [-h]

fs sp [-se <*fileserver names and ranks*>+] [-vl <*VL server names and ranks*>+] [-f <*input from named file*>] [-st] [-h]

Description

The fs setserverprefs command sets the local Cache Manager's preference ranks for one or more file server machine interfaces or, if the -vlserver argument is provided, for Volume Location (VL) Server machines. For file server machines, the numerical ranks determine the order in which the Cache Manager attempts to contact the interfaces of machines that are housing a volume. For VL Server machines, the ranks determine the order in which the Cache Manager attempts to contact a cell's VL Servers when requesting VLDB information.

The fs getserverprefs reference page explains how the Cache Manager uses preference ranks when contacting file server machines or VL Server machines. The following paragraphs explain how the Cache Manager calculates default ranks, and how to use this command to change the defaults.

Calculation of Default Preference Ranks

The Cache Manager stores a preference rank in kernel memory as a paired IP address and numerical rank. If a file server machine is multihomed, the Cache Manager assigns a distinct rank to each of the machine's addresses (up to the number of addresses that the VLDB can store per machine, which is specified in the *IBM AFS Release Notes*). Once calculated, a rank persists until the machine reboots, or until this command is used to change it.

The Cache Manager sets default VL Server preference ranks as it initializes, randomly assigning a rank from the range 10,000 to 10,126 to each of the machines listed in the local `/usr/vice/etc/CellServDB` file. Machines from different cells can have the same rank, but this does not present a problem because the Cache Manager consults only one cell's ranks at a time.

The Cache Manager sets default preference ranks for file server machine as it fetches volume location information from the VLDB. Each time it learns about file server machine interfaces for which it has not already set ranks, it assigns a rank to each interface. If the local client machine has only one IP address, the Cache Manager compares it to the server interface's IP address and sets a rank according to the following algorithm. If the client machine is multihomed, the

Cache Manager applies the algorithm to each of the client machine's addresses and assigns to the file server machine interface the lowest rank that results.

- If the local machine is a file server machine, the base rank for each of its interfaces is 5,000.
- If the file server machine interface is on the same subnetwork as the client interface, its base rank is 20,000.
- If the file server machine interface is on the same network as the client interface, or is at the distant end of a point-to-point link with the client interface, its base rank is 30,000.
- If the file server machine interface is on a different network than the client interface, or the Cache Manager cannot obtain network information about it, its base rank is 40,000.

After assigning a base rank to a file server machine interface, the Cache Manager adds to it a number randomly chosen from the range 0 (zero) to 14. As an example, a file server machine interface in the same subnetwork as the local machine receives a base rank of 20,000, but the Cache Manager records the actual rank as an integer between 20,000 and 20,014. This process reduces the number of interfaces that have exactly the same rank. As with VL Server machine ranks, it is possible for file server machine interfaces from foreign cells to have the same rank as interfaces in the local cell, but this does not present a problem. Only the relative ranks of the interfaces that house a given volume are relevant, and AFS only supports storage of a volume in one cell at a time.

Setting Non-default Preference Ranks

Use the fs setserverprefs command to reset an existing preference rank, or to set the initial rank of a file server machine interface or VL Server machine for which the Cache Manager has no rank. To make a rank persist across a reboot of the local machine, place the appropriate fs setserverprefs command in the machine's AFS initialization file.

Specify each preference rank as a pair of values separated by one or more spaces:

- The first member of the pair is the fully-qualified hostname (for example, `fs1.abc.com`), or the IP address in dotted decimal format, of a file server machine interface or VL Server machine
- The second member of the pair is an integer. The possible ranks range from `1` through `65535`.

As with default ranks, the Cache Manager adds a randomly chosen integer to a rank specified by this command. For file server machine interfaces, the integer is from the range 0 (zero) to 14; for VL Server machines, it is from the range 0 (zero) to 126. For example, if the administrator assigns a rank of 15,000 to a file server machine interface, the Cache Manager stores an integer between 15,000 to 15,014.

There are several ways to provide ranks for file server machine interfaces (but not for VL Server machines):

- On the command line, following the -servers argument.

- In a file named by the -file argument. Place each pair on its own line in the file. Directing the output from the fs getserverprefs command to a file automatically generates a file with the proper format.
- Via the standard input stream, by providing the -stdin flag. This method enables the issuer to feed in values directly from a program or script that generates preference ranks by using an algorithm appropriate to the local cell. The AFS distribution does not include such programs or scripts.

When setting file server machine preference ranks, it is legal to combine the -servers, -file, and -stdin options on a single command line. If different options specify a different rank for the same interface, the Cache Manager stores and uses the rank assigned with the -servers argument.

The -vlservers argument is the only way to assign VL Server machine ranks. It can be combined with one or more of the -servers, -file, and -stdin options, but the Cache Manager applies the values provided for those options to file server machine ranks only.

The fs command interpreter does not verify hostnames or IP addresses, and so assigns preference ranks to invalid machine names or addresses. The Cache Manager never uses such ranks unless the same incorrect information is in the VLDB.

Options

-servers <*file server names and ranks*>+

Specifies one or more file server machine preference ranks. Each rank pairs the fully-qualified hostname or IP address (in dotted decimal format) of a file server machine's interface with an integer rank, separated by one or more spaces; also separate each pair with one or more spaces. Acceptable values for the rank range from `1` through `65521`; a lower value indicates a greater preference. Providing ranks outside this range can have unpredictable results. Providing a value no larger than `65521` guarantees that the rank does not exceed the maximum possible value of 65,535 even if the largest random factor (14) is added.

This argument can be combined with the -file argument, -stdin flag, or both. If more than one of the arguments sets a rank for the same interface, the rank set by this argument takes precedence. It can also be combined with the -vlservers argument, but does not interact with it.

-vlservers <*VL server names and ranks*>+

Specifies one or more VL Server preference ranks. Each rank pairs the fully-qualified hostname or IP address (in dotted decimal format) of a VL Server machine with an integer rank, separated by one or more spaces; also separate each pair with one or more spaces. Acceptable values for the rank range from `1` through `65521`; a lower value indicates a greater preference. Providing ranks outside this range can have unpredictable results. Providing a value no larger than `65521` guarantees that the rank does not exceed the maximum possible value of 65,535 even if the largest random factor (14) is added.

This argument can be combined with the -servers argument, -file argument, -stdin flag, or any combination of the three, but does not interact with any of them. They apply only to file server machine ranks.

-file <*input file*>

Specifies the full pathname of a file from which to read pairs of file server machine interfaces and their ranks, using the same notation and range of values as for the -servers argument. In the file, place each pair on its own line and separate the two parts of each pair with one or more spaces.

This argument can be combined with the -servers argument, -stdin flag, or both. If more than one of the arguments sets a rank for the same interface, the rank set by the -server argument takes precedence. It can also be combined with the -vlservers argument, but does not interact with it.

-stdin

Reads pairs of file server machine interface and integer rank from the standard input stream. The intended use is to accept input piped in from a user-defined program or script that generates ranks in the appropriate format, but it also accepts input typed to the shell. Format the interface and rank pairs as for the -file argument. If typing at the shell, type Ctrl-D after the final newline to complete the input.

This argument can be combined with the -servers argument, the -file argument, or both. If more than one of the arguments sets a rank for the same interface, the rank set by the -server argument takes precedence. It can also be combined with the -vlservers argument, but does not interact with it.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command sets the Cache Manager's preference ranks for the file server machines named `fs3.abc.com` and `fs4.abc.com`, the latter of which is specified by its IP address, 192.12.105.100. The machines reside in another subnetwork of the local machine's network, so their default base rank is 30,000. To increase the Cache Manager's preference for these machines, the issuer assigns a rank of `25000`, to which the Cache Manager adds an integer in the range from 0 to 15.

```
# fs setserverprefs -servers fs3.abc.com 25000 192.12.105.100 25000
```

The following command uses the -servers argument to set the Cache Manager's preference ranks for the same two file server machines, but it also uses the -file argument to read a collection of preference ranks from a file that resides in the local file */etc/fs.prefs*:

```
# fs setserverprefs -servers fs3.abc.com 25000 192.12.105.100 25000 \
    -file /etc/fs.prefs
```

The /etc/fs.prefs file has the following contents and format:

```
192.12.108.214        7500
192.12.108.212        7500
138.255.33.41         39000
138.255.33.34         39000
128.0.45.36           41000
128.0.45.37           41000
```

The following command uses the -stdin flag to read preference ranks from the standard input stream. The ranks are piped to the command from a program, calc_prefs, which was written by the issuer to calculate preferences based on values significant to the local cell.

```
# calc_prefs | fs setserverprefs -stdin
```

The following command uses the -vlservers argument to set the Cache Manager's preferences for the VL server machines named `fs1.abc.com`, `fs3.abc.com`, and `fs4.abc.com` to base ranks of 1, 11000, and 65521, respectively:

```
# fs setserverprefs -vlservers fs1.abc.com 1 fs3.abc.com 11000 \
    fs4.abc.com 65521
```

Privilege Required

The issuer must be logged in as the local superuser root.

See Also

"fs_getserverprefs(1)" on page 85

Copyright

fs setvol

Name

`fs setvol` — Set quota and messages for a volume containing a file or directory

Synopsis

fs setvol [-path <*dir/file path*>+] [-max <*disk space quota in 1K units*>] [-offlinemsg <*offline message*>] [-help]

fs setv [-p <*dir/file path*>+] [-ma <*disk space quota in 1K units*>] [-o <*offline message*>] [-h]

fs sv [-p <*dir/file path*>+] [-ma <*disk space quota in 1K units*>] [-o <*offline message*>] [-h]

Description

The fs setvol command sets the quota (maximum possible size) of the read / write volume that contains each directory or file named by the -path argument. To associate a message with the volume which then appears in the output of the fs examine command, include the -offlinemsg argument.

To display all of the settings made with this command, use the fs examine command. The fs listquota command reports a fileset's quota, and the fs quota command the percent of quota used.

To set quota on one volume at a time, use the fs setquota command.

Options

-path <*dir/file path*>+

Names each file or directory for which to set the host volume's quota and offline message. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

Specify the read / write path to the file or directory, to avoid the failure that results from attempting to change a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, `/afs/.abc.com`). For further discussion of the concept of read / write and read-only paths through the filespace, see the fs mkmount reference page.

-max <*disk space quota in 1K units*>

Sets the maximum amount of file server disk space the volume can occupy. Provide a positive integer to indicate the number of one-kilobyte blocks (`1024` is one megabyte). A value of `0` sets an unlimited quota, but the size of the disk partition that houses the volume places an absolute limit on the volume's size.

If the -path argument is omitted (so that the command sets the quota of the volume housing the current working directory), the -max switch must be provided.

-offlinemsg

Associates a message with the volume which then appears in the output of the fs examine command. Its intended use is to explain why the volume is currently offline.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command imposes a 6500 kilobyte quota on the volumes mounted at the home directories */afs/abc.com/usr/smith* and */afs/abc.com/usr/pat*:

```
% cd /afs/abc.com/usr
% fs setvol -path smith pat -max 6500B&lt;&gt;
```

Privilege Required

The issuer must belong to the system:administrators group.

See Also

"fs_examine(1)" on page 65, "fs_listquota(1)" on page 97, "fs_mkmount(1)" on page 105, "fs_quota(1)" on page 113, "fs_setquota(1)" on page 131

Copyright

fs storebehind

Name

`fs storebehind` — Enables asynchronous writes to the file server

Synopsis

fs storebehind [-kbytes <*asynchrony for specified names*>] [-files <*specific pathnames*>+] [-allfiles <*new default (KB)*>] [-verbose] [-help]

fs st [-k <*asynchrony for specified names*>] [-f <*specific pathnames*>+] [-a <*new default (KB)*>] [-v] [-h]

Description

The fs storebehind command enables the Cache Manager to perform a delayed asynchronous write to the File Server when an application closes a file. By default, the Cache Manager writes all data to the File Server immediately and synchronously when an application program closes a file -- that is, the close() system call does not return until the Cache Manager has actually transferred the final chunk of the file to the File Server. This command specifies the number of kilobytes of a file that can still remain to be written to the File Server when the Cache Manager returns control to the application. It is useful if users working on the machine commonly work with very large files, but also introduces the complications discussed in the "CAUTIONS" on page .

Set either or both of the following in a single command:

- To set a value that applies to all AFS files manipulated by applications running on the machine, use the -allfiles argument. This value is termed the *default store asynchrony* for the machine, and persists until the machine reboots. If it is not set, the default value is zero, indicating that the Cache Manager performs synchronous writes.

  As an example, the following setting means that when an application closes a file, the Cache Manager can return control to the application as soon as no more than 10 kilobytes of the file remain to be written to the File Server.

  ```
  -allfiles 10
  ```

- To set a value that applies to one or more individual files, and overrides the value of the -allfiles argument for them, combine the -kbytes and -files arguments. The setting persists as long as there is an entry for the file in the kernel table that the Cache Manager uses to track certain information about files. In general, such an entry persists at least until an application closes the file or exits, but the Cache Manager is free to recycle the entry if the file is inactive and it needs to free up slots in the table. To increase the certainty that there is an entry for the file in the table, issue the fs storebehind command shortly before closing the file.

  As an example, the following setting means that when an application closes either of the files bigfile and biggerfile, the Cache Manager can return control to the application as soon as no more than a megabyte of the file remains to be written to the File Server.

```
-kbytes 1024 -files bigfile biggerfile
```

Note that once an explicit value has been set for a file, the only way to make it subject to the default store asynchrony once again is to set -kbytes to that value. In other words, there is no combination of arguments that automatically makes a file subject to the default store asynchrony once another value has been set for the file.

To display the settings that currently apply to individual files or to all files, provide the command's arguments in certain combinations as specified in "OUTPUT" on page .

Cautions

For the following reasons, use of this command is not recommended in most cases.

In normal circumstances, an asynchronous setting results in the Cache Manager returning control to applications earlier than it otherwise does, but this is not guaranteed.

If a delayed write fails, there is no way to notify the application, since the close() system call has already returned with a code indicating success.

Writing asynchronously increases the possibility that the user will not notice if a write operation makes the volume that houses the file exceed its quota. As always, the portion of the file that exceeds the volume's quota is lost, which prompts a message such as the following:

```
No space left on device
```

To avoid losing data, it is advisable to verify that the volume housing the file has space available for the amount of data anticipated to be written.

Options

-kbytes <*asynchrony for specified names*>

Specifies the number of kilobytes of data from each file named by the -files argument that can remain to be written to the file server when the Cache Manager returns control to an application program that closed the file. The -files argument is required along with this argument. Provide an integer from the range `0` (which reinstates the Cache Manager's default behavior or writing synchronously) to the maximum AFS file size.

-files <*specific pathnames*>+

Names each file to which the value set with the -kbytes argument applies. The setting persists as long as there is an entry for the file in the kernel table that the Cache Manager uses to track certain information about files. Because closing a file generally erases the entry, when reopening a file the only way to guarantee that the setting still applies is to reissue the command. If this argument is provided without the -kbytes argument, the command reports the current setting for the specified files, and the default store asynchrony.

-allfiles *<new default (KB)>*

Sets the default store asynchrony for the local machine, which is the number of kilobytes of data that can remain to be written to the file server when the Cache Manager returns control to the application program that closed a file. The value applies to all AFS files manipulated by applications running on the machine, except those for which settings have been made with the -kbytes and -files arguments. Provide an integer from the range `0` (which indicates the default of synchronous writes) to the maximum AFS file size.

-verbose

Produces output confirming the settings made with the accompanying -kbytes and -files arguments, the -allfiles argument, or all three. If provided by itself, reports the current default store asynchrony.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If none of the command's options are included, or if only the -verbose flag is included, the following message reports the default store asynchrony (the setting that applies to all files manipulated by applications running on the local machine and for which not more specific asynchrony is set).

```
Default store asynchrony is &lt;x&gt; kbytes.
```

A value of `0` (zero) indicates synchronous writes and is the default if no one has included the -allfiles argument on this command since the machine last rebooted.

If the -files argument is provided without the -kbytes argument, the output reports the value that applies to each specified file along with the default store asynchrony. If a particular value has previously been set for a file, the following message reports it:

```
Will store up to &lt;y&gt; kbytes of &lt;file&gt; asynchronously.
Default store asynchrony is &lt;x&gt; kbytes.
```

If the default store asynchrony applies to a file because no explicit -kbytes value has been set for it, the message is instead as follows:

```
Will store &lt;file&gt; according to default.
Default store asynchrony is &lt;x&gt; kbytes.
```

If the -verbose flag is combined with arguments that set values (-files and -kbytes, or -allfiles, or all three), there is a message that confirms immediately that the setting has taken effect. When included without other arguments or flags, the -verbose flag reports the default store asynchrony only.

Examples

The following command enables the Cache Manager to return control to the application program that closed the file *test.data* when 100 kilobytes still remain to be written to the File Server. The -verbose flag produces output that confirms the new setting, and that the default store asynchrony is zero.

```
% fs storebehind -kbytes 100 -files test.data -verbose
Will store up to 100 kbytes of test.data asynchronously.
Default store asynchrony is 0 kbytes.
```

Privilege Required

To include the -allfiles argument, the issuer must be logged in as the local superuser `root`.

To include the -kbytes and -files arguments, the issuer must either be logged in as the local superuser `root` or have the `w` (write) permission on the ACL of each file's directory.

To view the current settings (by including no arguments, the -file argument alone, or the -verbose argument alone), no privilege is required.

See Also

"afsd(8)" on page 523

Copyright

fs sysname

Name

`fs sysname` — Reports or sets the CPU / operating system type

Synopsis

fs sysname [-newsys <*new sysname*>] [-help]

fs sy [-n <*new sysname*>] [-h]

Description

The fs sysname command sets or displays the local machine's CPU / operating system type as recorded in kernel memory. The Cache Manager substitutes the string for the *@sys* variable which can occur in AFS pathnames; the *IBM AFS Quick Beginnings* and *IBM AFS Administration Guide* explain how using *@sys* can simplify cell configuration. It is best to use it sparingly, however, because it can make the effect of changing directories unpredictable.

The command always applies to the local machine only. If issued on an NFS client machine accessing AFS via the NFS / AFS Translator, the string is set or reported for the NFS client machine. The Cache Manager on the AFS client machine serving as the NFS client's NFS / AFS translator machine stores the value in its kernel memory, and so can provide the NFS client with the proper version of program binaries when the user issues commands for which the pathname to the binaries includes *@sys*. There is a separate record for each user logged into the NFS client, which implies that if a user adopts a new identity (UNIX UID) during a login session on the NFS client -- perhaps by using the UNIX su command -- he or she must verify that the correct string is set for the new identity also.

Options

-newsys <*new sysname*>

Sets the CPU / operating system indicator string for the local machine. If this argument is omitted, the output displays the current setting instead. AFS uses a standardized set of strings; consult the *IBM AFS Quick Beginnings* or *AFS Release Notes*.

-help

Prints the online help for this command. All other valid options are ignored.

Output

When the -newsys argument is omitted, the output reports the machine's system type in the following format:

```
Current sysname is '&lt;system_type&gt;'
```

Examples

The following example shows the output produced on a Sun SPARCStation running Solaris 5.7:

```
% fs sysname
Current sysname is 'sun4x_57'
```

The following command defines a machine to be a IBM RS/6000 running AIX 4.2:

```
% fs sysname -newsys rs_aix42
```

Privilege Required

To display the current setting, no privilege is required. To include the -newsys argument on an AFS client machine, the issuer must be logged in as the local superuser `root`.

See Also

"fs_exportafs(1)" on page 69, "sys(1)" on page 267

IBM AFS Quick Beginnings

IBM AFS Administration Guide

Copyright

fs whereis

Name

`fs whereis` — Reports each file server housing a file or directory

Synopsis

fs whereis [-path <*dir/file path*>+] [-help]

fs whe [-p <*dir/file path*>+] [-h]

Description

The fs whereis command returns the name of each file server machine that houses the volume containing each directory or file named by the -path argument.

Options

-path <*dir/file path*>+

Names each AFS file or directory for which to return the host file server machine. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

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 the cell is not clear, use the fs whichcell command to display the cell in which the directory or file resides. To display the cell membership of the local machine, use the fs wscell command.

Examples

The following example indicates that volume housing the directory */afs/abc.com* resides is replicated on both `fs1.abc.com` and `fs3.abc.com`:

```
% fs whereis -path /afs/abc.com
File /afs/abc.com is on hosts fs1.abc.com fs3.abc.com
```

Privilege Required

None

See Also

"fs_whichcell(1)" on page 149, "fs_wscell(1)" on page 151

Copyright

fs whichcell

Name

`fs whichcell` — Returns the cell to which a file or directory belongs

Synopsis

fs whichcell [-path <*dir/file path*>+] [-help]

fs whi [-p <*dir/file path*>+] [-h]

Description

The fs whichcell command returns the name of the cell in which the volume that houses each indicated directory or file resides.

To display the file server machine on which the volume housing a directory or file resides, use the fs whichcell command. To display the cell membership of the local machine, use the fs wscell command.

Options

-path *Idir/file path*>+

Names each AFS file or directory for which to return the cell membership. Partial pathnames are interpreted relative to the current working directory, which is also the default value if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output includes a line for each directory or file, naming the cell to which the volume that houses the directory or file resides.

Examples

The following example shows that the current working directory resides in a volume in the ABC Corporation cell:

```
% fs whichcell
File . lives in cell 'abc.com'
```

Privilege Required

None

See Also

"fs_wscell(1)" on page 151, "fs_whereis(1)" on page 147

Copyright

fs wscell

Name

`fs wscell` — Returns the name of the cell to which a machine belongs

Synopsis

fs wscell [-help]

fs ws [-h]

Description

The fs wscell command returns the name of the local machine's home cell.

Options

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output displays the contents of the local */usr/vice/etc/ThisCell* file, in the format

```
This workstation belongs to cell '&lt;cellname&gt;'
```

Examples

The following example results when the fs wscell is issued on a machine in the State University cell:

```
% fs wscell
This workstation belongs to cell 'stateu.edu'
```

Privilege Required

None

See Also

"fs_whereis(1)" on page 147, "fs_whichcell(1)" on page 149

Copyright

klog

Name

`klog` — Authenticates with the Authentication Server

Synopsis

klog [-x] [-principal <*user name*>] [-password <*user's password*>] [-cell <*cell name*>] [-servers <*explicit list of servers*>+] [-pipe] [-silent] [-lifetime <*ticket lifetime in hh[:mm[:ss]]*>] [-setpag] [-tmp] [-help]

klog [-x] [-pr <*user name*>] [-pa <*user's password*>] [-c <*cell name*>] [-s <*explicit list of servers*>+] [-pi] [-si] [-l <*ticket lifetime in hh[:mm[:ss]]*>] [-se] [-t] [-h]

Description

The klog command obtains an AFS token from the Authentication Server. The Cache Manager on the local machine stores the token in a credential structure in kernel memory and uses it when obtaining authenticated access to the AFS filespace. This command does not affect the issuer's identity (UNIX UID) in the local file system.

By default, the command interpreter obtains a token for the AFS user name that matches the issuer's identity in the local file system. To specify an alternate user, include the -principal argument. The user named by the -principal argument does not have to appear in the local password file (the */etc/passwd* file or equivalent).

By default, the command interpreter obtains a token for the local cell, as defined by the AFSCELL environment variable set in the command shell or by the */usr/vice/etc/ThisCell* file on the local machine. To specify an alternate cell, include the -cell argument. The command interpreter contacts an Authentication Server chosen at random from the cell's entry in the local */usr/afs/etc/CellServDB* file, unless the -servers argument is used to name one or more database server machines.

A user can have tokens in multiple cells simultaneously, but only one token per cell per connection to the client machine. If the user's credential structure already contains a token for the requested cell, the token resulting from this command replaces it.

Sites that employ standard Kerberos authentication instead of the AFS Authentication Server must use the Kerberos version of this command, klog.krb, on all client machines. It automatically places the issuer's Kerberos tickets in the file named by the KRBTKFILE environment variable, which the pagsh.krb command defines automatically as */tmp/tktpX* where *X* is the number of the user's PAG.

The lifetime of the token resulting from this command is the smallest of the following.

- The lifetime specified 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 for the afs entry in the Authentication Database. The default is 100 hours.

- The maximum ticket lifetime recorded in the specified user's Authentication Database entry. The default is 25 hours for user entries created by an Authentication Server running AFS 3.1 or later.
- 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).

The output from the kas examine command displays an Authentication Database entry's maximum ticket lifetime as `Max ticket lifetime`. Administrators can display any entry, and users can display their own entries.

If none of the defaults have been changed, the token lifetime is 25 hours for user accounts created by an Authentication Server running AFS 3.1 or higher. The maximum lifetime for any token is 720 hours (30 days), and the minimum is 5 minutes.

Between the minimum and maximum values, the Authentication Server uses a defined set of values, according to the following rules. Requested 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 (as indicated by the `d` and `h` letters). For example, `282:22:17` means 282 hours, 22 minutes, and 17 seconds, which translates to approximately 11 days and 18 hours (`11d 18h`). The Authentication Server rounds up a requested lifetime to the next highest possible lifetime.

```
11:24:15 (0d 11h)    46:26:01 (1d 22h)  189:03:38 (7d 21h)
12:11:34 (0d 12h)    49:38:40 (2d 01h)  202:08:00 (8d 10h)
13:02:09 (0d 13h)    53:04:37 (2d 05h)  216:06:35 (9d 00h)
13:56:14 (0d 13h)    56:44:49 (2d 08h)  231:03:09 (9d 15h)
14:54:03 (0d 14h)    60:40:15 (2d 12h)  247:01:43 (10d 07h)
15:55:52 (0d 15h)    64:51:57 (2d 16h)  264:06:34 (11d 00h)
17:01:58 (0d 17h)    69:21:04 (2d 21h)  282:22:17 (11d 18h)
18:12:38 (0d 18h)    74:08:46 (3d 02h)  301:53:45 (12d 13h)
19:28:11 (0d 19h)    79:16:23 (3d 07h)  322:46:13 (13d 10h)
20:48:57 (0d 20h)    84:45:16 (3d 12h)  345:05:18 (14d 09h)
22:15:19 (0d 22h)    90:36:53 (3d 18h)  368:56:58 (15d 08h)
23:47:38 (0d 23h)    96:52:49 (4d 00h)  394:27:37 (16d 10h)
25:26:21 (1d 01h)   103:34:45 (4d 07h)  421:44:07 (17d 13h)
27:11:54 (1d 03h)   110:44:28 (4d 14h)  450:53:46 (18d 18h)
29:04:44 (1d 05h)   118:23:54 (4d 22h)  482:04:24 (20d 02h)
31:05:22 (1d 07h)   126:35:05 (5d 06h)  515:24:22 (21d 11h)
33:14:21 (1d 09h)   135:20:15 (5d 15h)  551:02:38 (22d 23h)
35:32:15 (1d 11h)   144:41:44 (6d 00h)  589:08:45 (24d 13h)
37:59:41 (1d 13h)   154:42:01 (6d 10h)  629:52:56 (26d 05h)
40:37:19 (1d 16h)   165:23:50 (6d 21h)  673:26:07 (28d 01h)
43:25:50 (1d 19h)   176:50:01 (7d 08h)
```

Cautions

By default, this command does not create a new process authentication group (PAG); see the description of the pagsh command to learn about PAGs. If a cell does not use an AFS-modified login utility, users must include -setpag option to this command, or issue the pagsh command before this one, to have their tokens stored in a credential structure that is identified by PAG rather than by local UID.

When a credential structure is identified by local UID, the potential security exposure is that the local superuser `root` can use the UNIX su command to assume any other identity and automatically inherit the tokens associated with that UID. Identifying the credential structure by PAG eliminates this exposure.

If the -password argument is used, the specified password cannot begin with a hyphen, because it is interpreted as another option name. Use of the -password argument is not recommended in any case.

By default, it is possible to issue this command on a properly configured NFS client machine that is accessing AFS via the NFS / AFS Translator, assuming that the NFS client machine is a supported system type. However, if the translator machine's administrator has enabled UID checking by including the -uidcheck on argument to the fs exportafs command, the command fails with an error message similar to the following:

```
Warning: Remote pioctl to &lt;translator_machine&gt; has failed (err=8). . .
Unable to authenticate to AFS because a pioctl failed.
```

Enabling UID checking means that the credential structure in which tokens are stored on the translator machine must be identified by a UID that matches the local UID of the process that is placing the tokens in the credential structure. After the klog command interpreter obtains the token on the NFS client, it passes it to the remote executor daemon on the translator machine, which makes the system call that stores the token in a credential structure on the translator machine. The remote executor generally runs as the local superuser `root`, so in most cases its local UID (normally zero) does not match the local UID of the user who issued the klog command on the NFS client machine.

Issuing the klog command on an NFS client machine creates a security exposure: the command interpreter passes the token across the network to the remote executor daemon in clear text mode.

Options

-x

Appears only for backwards compatibility. Its former function is now the default behavior of this command.

-principal <*user name*>

Specifies the user name to authenticate. If this argument is omitted, the Authentication Server attempts to authenticate the user logged into the local system.

-password <*user's password*>

Specifies the issuer's password (or that of the alternate user identified by the -principal argument). Omit this argument to have the command interpreter prompt for the password, in which case it does not echo visibly in the command shell.

-cell <*cell name*>

Specifies the cell for which to obtain a token. The command is directed to that cell's Authentication Servers. During a single login session on a given machine, a user can 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 session on a machine). It is acceptable to abbreviate the cell name to the shortest form that distinguishes it from the other cells listed in the `/usr/vice/etc/CellServDB` file 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 the `/usr/vice/etc/ThisCell` file on the client machine on which the command is issued.

-servers <*explicit list of servers*>+

Establishes a connection with the Authentication Server running on each specified database server machine. The command interpreter then chooses one of these at random to execute the command. It is best to provide fully-qualified hostnames, but abbreviated forms are possibly acceptable depending 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 copy of the `/usr/vice/etc/CellServDB` file, and then chooses one of them at random for command execution.

-pipe

Suppresses all output to the standard output stream, including prompts and error messages. The klog command interpreter expects to receive the password from the standard input stream. Do not use this argument; it is designed for use by application programs rather than human users.

-silent

Suppresses some of the trace messages that the klog command produces on the standard output stream by default. It still reports on major problems encountered.

-lifetime <*ticket lifetime*

Requests a specific lifetime for the token. Provide a number of hours and optionally minutes and seconds in the format *hh*[:*mm*[:*ss*]]. The value is used in calculating the token lifetime as described in "DESCRIPTION" on page .

-setpag

Creates a process authentication group (PAG) prior to requesting authentication. The token is associated with the newly created PAG.

-tmp

Creates a Kerberos-style ticket file in the */tmp* directory of the local machine. The file is called *tkt.AFS_UID* where *AFS_UID* is the AFS UID of the issuer.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The following message indicates that the limit on consecutive authentication failures has been exceeded. An administrator can use the kas unlock command to unlock the account, or the issuer can wait until the lockout time for the account has passed. (The time is set with the -locktime argument to the kas setfields command and displayed in the output from the kas examine command).

```
Unable to authenticate to AFS because ID is locked - see your system admin
```

If the -tmp flag is included, the following message confirms that a Kerberos-style ticket file was created:

```
Wrote ticket file to /tmp
```

Examples

Most often, this command is issued without arguments. The appropriate password is for the person currently logged into the local system. The ticket's lifetime is calculated as described in "DESCRIPTION" on page (if no defaults have been changed, it is 25 hours for a user whose Authentication Database entry was created in AFS 3.1 or later).

```
% klog
Password:
```

The following example authenticates the user as admin in the ABC Corporation's test cell:

```
% klog -principal admin -cell test.abc.com
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 "DESCRIPTION" on page , the token's lifetime is 110:44:28, which is the next largest possible value.

```
% klog -lifetime 104:30
Password:
```

Privilege Required

None

See Also

"fs_exportafs(1)" on page 69, "kas_examine(8)" on page 817, "kas_setfields(8)" on page 841, "kas_unlock(8)" on page 857, "kaserver(8)" on page 859, "pagsh(1)" on page 171, "tokens(1)" on page 269

Copyright

knfs

Name

knfs — Establishes authenticated access via the NFS / AFS Translator

Synopsis

knfs -host <*host name*> [-id <*user ID (decimal)*>] [-sysname <*host's '@sys' value*>] [-unlog] [-tokens] [-help]

knfs -ho <*host name*> [-i <*user ID (decimal)*>] [-s <*host's '@sys' value*>] [-u] [-t] [-he]

Description

The knfs command creates an AFS credential structure on the local machine, identifying it by a process authentication group (PAG) number associated with the NFS client machine named by the -hostname argument and by default with a local UID on the NFS client machine that matches the issuer's local UID on the local machine. It places in the credential structure the AFS tokens that the issuer has previously obtained (by logging onto the local machine if an AFS-modified login utility is installed, by issuing the klog command, or both). To associate the credential structure with an NFS UID that does not match the issuer's local UID, use the -id argument.

Issue this command only on the NFS(R) / AFS translator machine that is serving the NFS client machine, after obtaining AFS tokens on the translator machine for every cell to which authenticated access is required. The Cache Manager on the translator machine uses the tokens to obtain authenticated AFS access for the designated user working on the NFS client machine. This command is not effective if issued on an NFS client machine.

To enable the user on the NFS client machine to issue AFS commands, use the -sysname argument to specify the NFS client machine's system type, which can differ from the translator machine's. The NFS client machine must be a system type for which AFS is supported.

The -unlog flag discards the tokens in the credential structure, 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 user reissues the knfs command on the translator machine each time the user logs into the NFS client machine.

To display the tokens associated with the designated user on the NFS client machine, include the -tokens flag.

Users working on NFS client machines of system types for which AFS binaries are available can use the klog command rather than the knfs command.

Cautions

If the translator machine's administrator has enabled UID checking by issuing the fs exportafs command with the -uidcheck on argument, it is not possible to use the -id argument to assign

the tokens to an NFS UID that differs from the issuer's local UID. In this case, there is no point in including the -id argument, because the only acceptable value (the issuer's local UID) is the value used when the -id argument is omitted. Requiring matching UIDs is effective only when users have the same local UID on the translator machine as on NFS client machines. In that case, it guarantees that users assign their tokens only to their own NFS sessions.

This command does not make it possible for users working on non-supported system types to issue AFS commands. This is possible only on NFS clients of a system type for which AFS is available.

Options

-host <*host name*>

Names the NFS client machine on which the issuer is to work. Providing a fully-qualified hostname is best, but abbreviated forms are possibly acceptable depending on the state of the cell's name server at the time the command is issued.

-id <*user ID (decimal)*>

Specifies the local UID on the NFS client to which to assign the tokens. The NFS client identifies file requests by the NFS UID, so creating the association enables the Cache Manager on the translator machine to use the appropriate tokens when filling the requests. If this argument is omitted, the command interpreter uses an NFS UID that matches the issuer's local UID on the translator machine (as returned by the getuid() function).

-sysname <*host's '@sys' value*>

Specifies the value that the local (translator) machine's remote executor daemon substitutes for the *@sys* variable in pathnames when executing AFS commands issued on the NFS client machine (which must be a supported system type). If the NFS user's PATH environment variable uses the *@sys* variable in the pathnames for directories that house AFS binaries (as recommended), then setting this argument enables NFS users to issue AFS commands by leading the remote executor daemon to access the AFS binaries appropriate to the NFS client machine even if its system type differs from the translator machine's.

-unlog

Discards the tokens stored in the credential structure identified by the PAG associated with the -host argument and, optionally, the -id argument.

-tokens

Displays the AFS tokens assigned to the designated user on the indicated NFS client machine.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The following error message indicates that UID checking is enabled on the translator machine and that the value provided for the -id argument differs from the issuer's local UID.

```
knfs: Translator in 'passwd sync' mode; remote uid must be the same as
local uid
```

Examples

The following example illustrates a typical use of this command. The issuer `smith` is working on the machine `nfscli1.abc.com` and has user ID `1020` on that machine. The translator machine `tx4.abc.com` uses an AFS-modified login utility, so `smith` obtains tokens for the ABC Corporation cell automatically upon login via the telnet program. She then issues the klog command to obtain tokens as `admin` in the ABC Corporation's test cell, `test.abc.com`, and the knfs command to associate both tokens with the credential structure identified by machine name `nfs-cli1` and user ID `1020`. She breaks the connection to `tx4` and works on `nfscli1`.

```
% telnet tx4.abc.com
. . .
login: smith
Password:
AFS(R) login


% klog admin -cell test.abc.com
Password:


% knfs nfscli1.abc.com 1020


% exit
```

The following example shows user smith again connecting to the machine `tx4` via the telnet program and discarding the tokens.

```
% telnet translator4.abc.com
. . .
login: smith
Password:
AFS(R) login


% knfs nfscli1.abc.com 1020 -unlog


% exit
```

Privilege Required

None

See Also

"klog(1)" on page 153, "pagsh(1)" on page 171

Copyright

kpasswd

Name

`kpasswd` — Changes the issuer's password in the Authentication Database

Synopsis

kpasswd [-x] [-principal <*user name*>] [-password <*user's password*>] [-newpassword <*user's new password*>] [-cell <*cell name*>] [-servers <*explicit list of servers*>+] [-pipe] [-help]

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

The kpasswd command changes the password recorded in an Authentication Database entry. By default, the command interpreter changes the password for the AFS user name that matches the issuer's local identity (UNIX UID). To specify an alternate user, include the -principal argument. The user named by the -principal argument does not have to appear in the local password file (the `/etc/passwd` file or equivalent).

By default, the command interpreter sends the password change request to the Authentication Server running on one of the database server machines listed for the local cell in the `/usr/afs/etc/CellServDB` file on the local disk; it chooses the machine at random. It consults the `/usr/vice/etc/ThisCell` file on the local disk to learn the local cell name. To specify an alternate cell, include the -cell argument.

Unlike the UNIX passwd command, the kpasswd command does not restrict passwords to eight characters or less; it accepts passwords of virtually any length. All AFS commands that require passwords (including the klog, kpasswd, and AFS-modified login utilities, and the commands in the kas suite) accept passwords longer than eight characters, but some other applications and operating system utilities do not. Selecting an AFS password of eight characters or less enables the user to maintain matching AFS and UNIX passwords.

The command interpreter makes the following checks:

- If the program kpwvalid exists in the same directory as the kpasswd command, the command interpreter pass the new password to it for verification. For details, see "kpwvalid(8)" on page 867.
- If the -reuse argument to the kas setfields command has been used to prohibit reuse of previous passwords, the command interpreter verifies that the password is not too similar too any of the user's previous 20 passwords. It generates the following error message at the shell:

  ```
  Password was not changed because it seems like a reused password
  ```

  To prevent a user from subverting this restriction by changing the password twenty times in quick succession (manually or by running a script), use the -minhours argument on the

kaserver initialization command. The following error message appears if a user attempts to change a password before the minimum time has passed:

```
Password was not changed because you changed it too
recently; see your systems administrator
```

Options

-x

Appears only for backwards compatibility.

-principal <*user name*>

Names the Authentication Database entry for which to change the password. If this argument is omitted, the database entry with the same name as the issuer's local identity (UNIX UID) is changed.

-password <*user's password*>

Specifies the current password. Omit this argument to have the command interpreter prompt for the password, which does not echo visibly:

```
Old password: current_password
```

-newpassword <*user's new password*>

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.

Omit this argument to have the command interpreter prompt for the password, which does not echo visibly:

```
New password (RETURN to abort): &lt;new_password&gt;
Retype new password: &lt;new_password&gt;
```

-cell <*cell name*>

Specifies the cell in which to change the password, by directing the command to that cell's Authentication Servers. The issuer can abbreviate the cell name to the shortest form that distinguishes it from the other cells listed in the local */usr/vice/etc/CellServDB* file.

By default, the command is executed in the local cell, as defined

- First, by the value of the environment variable AFSCELL.
- Second, in the */usr/vice/etc/ThisCell* file on the client machine on which the command is issued.

-servers <*explicit list of servers*>

Establishes a connection with the Authentication Server running on each specified machine, rather than with all of the database server machines listed for the relevant cell in the local copy of the */usr/vice/etc/CellServDB* file. The kpasswd command interpreter then sends the password-changing request to one machine chosen at random from the set.

-pipe

Suppresses all output to the standard output stream or standard error stream. The kpasswd command interpreter expects to receive all necessary arguments, each on a separate line, from the standard input stream. Do not use this argument, which is provided for use by application programs rather than human users.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example shows user pat changing her password in the ABC Corporation cell.

```
% kpasswd
Changing password for 'pat' in cell 'abc.com'.
Old password:
New password (RETURN to abort):
Verifying, please re-enter new_password:
```

Privilege Required

None

See Also

"kas_setfields(8)" on page 841, "kas_setpassword(8)" on page 847, "klog(1)" on page 153, "kpwvalid(8)" on page 867

Copyright

livesys

Name

`livesys` — Reports the configured CPU / operating system type

Synopsis

livesys

Description

The livesys command displays the string stored in kernel memory that indicates the local machine's CPU / operating system (OS) type, conventionally called the *sysname*. The Cache Manager substitutes this string for the *@sys* variable which can occur in AFS pathnames; the *IBM AFS Quick Beginnings* and *IBM AFS Administration Guide* explain how using *@sys* can simplify cell configuration.

To set a new value in kernel memory, use the fs sysname command, which can also be used to view the current value. If a sysname list was set using fs sysname, only the first value in the list will be reported by livesys.

Cautions

To see the full sysname list, use fs sysname rather than this command. livesys is mostly useful for scripts that need to know the primary sysname for the local system (to create directories that will later be addressed using *@sys*, for example).

livesys first appeared in OpenAFS 1.2.2. Scripts that need to support older versions of AFS should parse the output of fs sysname or use sys.

Output

The machine's system type appears as a text string:

```
I&lt;system_type&gt;
```

Examples

The following example shows the output produced on a Linux system with a 2.6 kernel:

```
% livesys
i386_linux26
```

Privilege Required

None

See Also

"fs_sysname(1)" on page 145, "sys(1)" on page 267

IBM AFS Quick Beginnings

IBM AFS Administration Guide

Copyright

package_test

Name

`package_test` — Tests the validity of a package configuration file

Synopsis

package_test <*config file*>

Description

The package_test command tests the validity of a package configuration file created when a prototype file is compiled. The command interpreter prints error messages on the standard output stream.

This command does not use the syntax conventions of the AFS command suites. Provide the command name in full.

Options

<*config file*>

Specifies the package configuration file to validate.

Examples

The following example tests the validity of the package configuration file `staff.sun4x_56`.

```
% package_test staff.sun4x_56
```

Privilege Required

None

See Also

"package(5)" on page 475, "package(8)" on page 869

Copyright

pagsh

Name

pagsh — Creates a new PAG

Synopsis

pagsh

Description

The pagsh command 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 the issuer in the credential structure that the Cache Manager creates to track each user.

Any tokens acquired subsequently (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 the local superuser `root`) that the AFS server processes recognize only as `anonymous`. Unless PAGs are used, such daemons cannot access files in directories whose access control lists (ACLs) do not extend permissions to the system:anyuser group.
- It closes a potential security loophole: UNIX allows anyone already logged in as the local superuser `root` on a machine to assume any other identity by issuing the UNIX su command. If the credential structure is identified by a UNIX UID rather than a PAG, then the local superuser `root` can assume a UNIX UID and use any tokens associated with that UID. Use of a PAG as an identifier eliminates that possibility.

Cautions

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.

In cells that do not use an AFS-modified login utility, use this command to obtain a PAG before issuing the klog command (or include the -setpag argument to the klog command). If a PAG is not acquired, the Cache Manager stores the token in a credential structure identified by local UID rather than PAG. This creates the potential security exposure described in "DESCRIPTION" on page .

If users of NFS client machines for which AFS is supported are to issue this command as part of authenticating with AFS, do not use the fs exportafs command's -uidcheck on argument to

enable UID checking on NFS/AFS Translator machines. Enabling UID checking prevents this command from succeeding. See "klog(1)" on page 153.

If UID checking is not enabled on Translator machines, then by default it is possible to issue this command on a properly configured NFS client machine that is accessing AFS via the NFS/AFS Translator, assuming that the NFS client machine is a supported system type. The pagsh binary accessed by the NFS client must be owned by, and grant setuid privilege to, the local superuser `root`. The complete set of mode bits must be `-rwsr-xr-x`. This is not a requirement when the command is issued on AFS client machines.

However, if the translator machine's administrator has enabled UID checking by including the -uidcheck on argument to the fs exportafs command, the command fails with an error message similar to the following:

```
Warning: Remote setpag to &lt;translator_machine&gt; has failed (err=8). . .
setpag: Exec format error
```

Examples

In the following example, the issuer invokes the C shell instead of the default Bourne shell:

```
# pagsh -c /bin/csh
```

Privilege Required

None

See Also

"fs_exportafs(1)" on page 69, "klog(1)" on page 153, "tokens(1)" on page 269

Copyright

pts

Name

`pts` — Introduction to the pts command suite

Description

The commands in the pts command suite are the administrative interface to the Protection Server, which runs on each database server machine in a cell and maintains the Protection Database. The database stores the information that AFS uses to augment and refine 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 users and groups and specifies which of seven possible access permissions they have for the directory and the files it contains. (It is still possible to set a directory or file's mode bits, but AFS interprets them in its own way; see the chapter on protection in the *IBM AFS Administration Guide* for details.)

AFS enables users to define groups in the Protection Database and place them on ACLs to extend a set of rights to multiple users simultaneously. Groups simplify administration by making it possible to add someone to many ACLs by adding them to a group that already exists on those ACLs. Machines can also be members of a group, so that users logged into the machine automatically inherit the permissions granted to the group.

There are several categories of commands in the pts command suite:

- Commands to create and remove Protection Database entries: pts creategroup, pts createuser, and pts delete.
- Commands to administer and display group membership: pts adduser, pts listowned, pts membership, and pts removeuser.
- Commands to administer and display properties of user and group entries other than membership: pts chown, pts examine, pts listentries, pts rename, and pts setfields.
- Commands to set and examine the counters used when assigning IDs to users and groups: pts listmax and pts setmax.
- Commands to obtain help: pts apropos and pts help.

Options

The following arguments and flags are available on many commands in the pts suite. The reference page for each command also lists them, but they are described here in greater detail.

-cell <*cell name*>

Names the cell in which to run the command. It is acceptable to abbreviate the cell name to the shortest form that distinguishes it from the other entries in the `/usr/vice/etc/CellServDB` file on the local machine. If the -cell argument is omitted, the

command interpreter determines the name of the local cell by reading the following in order:

- The value of the AFSCELL environment variable.
- The local */usr/vice/etc/ThisCell* file.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution immediately. Without it, the command halts as soon as the first error is encountered. In either case, the pts command interpreter reports errors at the command shell. This flag is especially useful if the issuer provides many values for a command line argument; if one of them is invalid, the command interpreter continues on to process the remaining arguments.

-help

Prints a command's online help message on the standard output stream. Do not combine this flag with any of the command's other options; when it is provided, the command interpreter ignores all other options, and only prints the help message.

-noauth

Establishes an unauthenticated connection to the Protection Server, in which the server treats the issuer as the unprivileged user anonymous. It is useful only when authorization checking is disabled on the server machine (during the installation of a file server machine or when the bos setauth command has been used during other unusual circumstances). In normal circumstances, the Protection Server allows only privileged users to issue commands that change the Protection Database, and refuses to perform such an action even if the -noauth flag is provided.

Privilege Required

Members of the system:administrators group can issue all pts commands on any entry in the Protection Database.

Users who do not belong to the system:administrators group can list information about their own entry and any group entries they own. The privacy flags set with the pts setfields command control access to entries owned by other users.

See Also

"pts_adduser(1)" on page 177, "pts_apropos(1)" on page 181, "pts_chown(1)" on page 183, "pts_creategroup(1)" on page 185, "pts_createuser(1)" on page 189, "pts_delete(1)" on page 193, "pts_examine(1)" on page 195, "pts_help(1)" on page 199, "pts_listentries(1)" on page 201, "pts_listmax(1)" on page 205, "pts_listowned(1)" on page 207, "pts_membership(1)" on page 211, "pts_removeuser(1)" on page 215, "pts_rename(1)" on page 217, "pts_setfields(1)" on page 221, "pts_setmax(1)" on page 225

Copyright

pts

pts adduser

Name

`pts adduser` — Adds a user or machine to a Protection Database group

Synopsis

pts adduser -user <*user name*>+ -group <*group name*>+ [-cell <*cell name*>] [-noauth] [-force] [-help]

pts ad -u <*user name*>+ -g <*group name*>+ [-c <*cell name*>] [-n] [-f] [-h]

Description

The pts adduser command adds each user or machine entry named by the -user argument as a member of each group named by the -group argument.

To remove members of a group, use the pts removeuser command. To list the groups to which a user or machine belongs, or the members of a specified group, use the pts membership command.

Cautions

After being added as a group member, a currently authenticated user must reauthenticate (for example, by issuing the klog command) to obtain permissions granted to the group on an access control list (ACL).

Options

-user <*user name*>+

Specifies the name of each user or machine entry to add to each group named by the -group argument. The name of a machine entry resembles an IP address and can use the wildcard notation described on the pts createuser reference page. The user or machine entry must already exist in the Protection Database.

-group <*group name*>+

Specifies the complete name (including the owner prefix if applicable) of each group to which to add members. The group entry must already exist in the Protection Database.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example adds user smith to the group system:administrators.

```
% pts adduser -user smith -group system:administrators
```

The following example adds users `jones`, `terry`, and pat to the smith:colleagues group.

```
% pts adduser -user jones terry pat -group smith:colleagues
```

The following example adds the machine entries in the ABC Corporation subnet to the group `bin-prot`. Because of the IP address range of the ABC Corporation subnet, the system administrator was able to group the machines into three machine entries (using the wildcard notation discussed on the pts createuser reference page).

```
% pts adduser -user 138.255.0.0 192.12.105.0 192.12.106.0 -group bin-prot
```

Privilege Required

The required privilege depends on the setting of the fourth privacy flag in the Protection Database entry for each group named by the -group argument (use the pts examine command to display the flags):

- If it is the hyphen, only the group's owner and members of the system:administrators group can add members.
- If it is lowercase `a`, current members of the group can add new members.
- If it is uppercase `A`, anyone who can access the cell's database server machines can add new members.

See Also

"pts(1)" on page 173, "pts_createuser(1)" on page 189, "pts_examine(1)" on page 195, "pts_membership(1)" on page 211, "pts_removeuser(1)" on page 215, "pts_setfields(1)" on page 221

Copyright

pts apropos

Name

`pts apropos` — Displays each help entry containing a keyword string

Synopsis

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

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

Description

The pts apropos command displays the first line of the online help entry for any pts command that has in its name or short description the string specified by the -topic argument.

To display the syntax for a command, use the pts help command.

Options

-topic <*help string*>

Specifies the keyword string to match, in lowercase letters only. If the string is more than a single word, surround it with double quotes ("") or other delimiters.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of a command's online help entry names it and briefly describes its function. This command displays the first line for any pts command in which the string specified by the -topic argument is part of the command name or first line.

Examples

The following command lists all pts commands that include the word `create` in their names or short descriptions:

```
% pts apropos create
creategroup: create a new group
createuser: create a new user
```

Privilege Required

None

See Also

"pts(1)" on page 173, "pts_help(1)" on page 199

Copyright

pts chown

Name

`pts chown` — Changes the owner of a Protection Database entry

Synopsis

pts chown -name <*group name*> -owner <*new owner*> [-cell <*cell name*>] [-noauth] [-force] [-help]

pts cho -na <*group name*> -o <*new owner*> [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts chown command designates the user or group named by the -owner argument as the owner of the group named by the -name argument, and records the new owner in the owner field of the group's Protection Database entry.

In the case of regular groups, this command automatically changes the group name's owner prefix (the part of the group name before the colon) to match the new owner. If the new owner is itself a group, then only its owner prefix, not its complete name, becomes the owner prefix in the new name. The change to the owner prefix does not propagate to any groups owned by the group, however. To make the owner prefix of such group-owned groups reflect the new owning group, use the pts rename command.

It is not possible to change a user or machine entry's owner from the default set at creation time, the system:administrators group.

Cautions

While designating a machine as a group's owner does not cause an error, it is not recommended. The Protection Server does not extend the usual privileges of group ownership to users logged onto the machine.

Options

-name <*group name*>

Specifies the current name of the group to which to assign a new owner.

-owner <*new owner*>

Names the user or group to become the group's owner.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example changes the owner of the group `terry:friends` from the user `terry` to the user `pat`. A side effect is that the group name changes to `pat:friends`.

```
% pts chown -name terry:friends -owner pat
```

The following example changes the owner of the group `terry:friends` from the user `terry` to the group `pat:buddies`. A side effect is that the group name changes to `pat:friends`.

```
% pts chown -name terry:friends -owner pat:buddies
```

Privilege Required

The issuer must belong to the system:administrators group or currently own the group.

See Also

"pts(1)" on page 173, "pts_rename(1)" on page 217

Copyright

pts creategroup

Name

`pts creategroup` — Creates an (empty) Protection Database group entry

Synopsis

pts creategroup -name <*group name*>+ [-owner <*owner of the group*>] [-id <*id (negated) for the group*>+] [-cell <*cell name*>] [-noauth] [-force] [-help]

pts createg -na <*group name*>+ [-o <*owner of the group*>] [-i <*id (negated) for the group*>+] [-c <*cell name*>] [-no] [-f] [-h]

pts cg -na <*group name*>+ [-o <*owner of the group*>] [-i <*id (negated) for the group*>+] [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts creategroup command creates an entry in the Protection Database for each group specified by the -name argument. The entry records the issuer of the command as the group's creator, and as the group's owner unless the -owner argument names an alternate user or group as the owner.

There are two types of groups:

- *regular*, the names of which have two parts separated by a colon. The part before the colon names the group's owner. Any user can create such groups.
- *prefix-less*, which do not have an owner prefix. Only members of the system:administrators group can create prefix-less groups.

Creating a group lowers the issuer's group-creation quota by one. This is true even if the -owner argument is used to assign ownership to an alternate user or group. To display a user's group-creation quota, use the pts examine command; to set it, use the pts setfields command.

AFS group ID (AFS GID) numbers are negative integers and by default the Protection Server assigns a GID that is one less (more negative) than the current value of the `max group id` counter in the Protection Database, decrementing the counter by one for each group. Members of the system:administrators group can use the -id argument to assign specific AFS GID numbers. If any of the specified GIDs is lower (more negative) than the current value of the `max group id` counter, the counter is reset to that value. It is acceptable to specify a GID greater (less negative) than the current value of the counter, but the creation operation fails if an existing group already has it. To display or set the value of the `max group id` counter, use the pts listmax or pts setmax command, respectively.

Output

The command generates the following string to confirm creation of each group:

```
group &lt;name&gt; has id &lt;AFS GID&gt;
```

Cautions

Although using the -owner argument to designate a machine entry as a group's owner does not generate an error, it is not recommended. The Protection Server does not extend the usual privileges of group ownership to users logged onto the machine.

Options

-name <*group name*>

Specifies the name of each group to create. Provide a string of up to 63 characters, which can include lowercase (but not uppercase) letters, numbers, and punctuation marks. A regular name includes a single colon (:) to separate the two parts of the name; the colon cannot appear in a prefix-less group name.

A regular group's name must have the following format:

```
&lt;owner_name&gt;:&lt;group_name&gt;
```

and the <owner_name> field must reflect the actual owner of the group, as follows:

- If the optional -owner argument is not included, the field must match the AFS username under which the issuer is currently authenticated.
- If the -owner argument names an alternate AFS user, the field must match that AFS username.
- If the -owner argument names another regular group, the field must match the owning group's owner field (the part of its name before the colon). If the -owner argument names a prefix-less group, the field must match the owning group's complete name.

-owner <*owner of the group*>

Specifies a user or group as the owner for each group, rather than the issuer of the command. Provide either an AFS username or the name of a regular or prefix-less group. An owning group must already have at least one member. This requirement prevents assignment of self-ownership to a group during its creation; use the pts chown command after issuing this command, if desired.

-id <*id for the group*>

Specifies a negative integer AFS GID number for each group, rather than allowing the Protection Server to assign it. Precede the integer with a hyphen (-) to indicate that it is negative.

If this argument is used and the -name argument names multiple new groups, it is best to provide an equivalent number of AFS GIDs. The first GID is assigned to the first group,

the second to the second group, and so on. If there are fewer GIDs than groups, the Protection Server assigns GIDs to the unmatched groups based on the `max group id` counter. If there are more GIDs than groups, the excess GIDs are ignored. If any of the GIDs is lower (more negative) than the current value of the `max group id` counter, the counter is reset to that value.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

In the following example, the user pat creates groups called `pat:friends` and `pat:colleagues`.

```
% pts creategroup -name pat:friends pat:colleagues
```

The following example shows a member of the system:administrators group creating the prefix-less group `staff` and assigning its ownership to the system:administrators group rather than to herself.

```
% pts creategroup -name staff -owner system:administrators
```

In the following example, the user pat creates a group called `smith:team-members`, which is allowed because the -owner argument specifies the required value (`smith`).

```
% pts creategroup -name smith:team-members -owner smith
```

Privilege Required

The issuer must belong to the system:administrators group to create prefix-less groups or include the -id argument.

To create a regular group, the issuer must

- Be authenticated. The command fails if the -noauth flag is provided.
- Have a group-creation quota greater than zero. The pts examine command displays this quota.

See Also

"pts(1)" on page 173, "pts_examine(1)" on page 195, "pts_listmax(1)" on page 205, "pts_setfields(1)" on page 221, "pts_setmax(1)" on page 225

Copyright

pts createuser

Name

`pts createuser` — Creates a user or machine entry in the Protection Database

Synopsis

pts createuser -name <*user name*>+ [-id <*user id*>+] [-cell <*cell name*>] [-noauth] [-force] [-help]

pts createu -na <*user name*>+ [-i <*user id*>+] [-c <*cell name*>] [-no] [-f] [-h]

pts cu -na <*user name*>+ [-i <*user id*>+] [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts createuser command creates an entry in the Protection Database for each user or machine specified by the -name argument. A user entry name becomes the user's AFS username (the one to provide when authenticating with the AFS Authentication Server). A machine entry's name is the machine's IP address or a wildcard notation that represents a range of consecutive IP addresses (a group of machines on the same network). It is not possible to authenticate as a machine, but a group to which a machine entry belongs can appear on a directory's access control list (ACL), thereby granting the indicated permissions to any user logged on to the machine.

AFS user IDs (AFS UIDs) are positive integers and by default the Protection Server assigns an AFS UID that is one greater than the current value of the `max user id` counter in the Protection Database, incrementing the counter by one for each user. To assign a specific AFS UID, use the -id argument. If any of the specified AFS UIDs is greater than the current value of the `max user id` counter, the counter is reset to that value. It is acceptable to specify an AFS UID smaller than the current value of the counter, but the creation operation fails if an existing user or machine entry already has it. To display or set the value of the `max user id` counter, use the pts listmax or pts setmax command, respectively.

The issuer of the pts createuser command is recorded as the entry's creator and the group system:administrators as its owner.

Cautions

The Protection Server reserves AFS UID 0 (zero) and returns an error if the -id argument has that value.

Options

-name <*user name*>+

Specifies either a username for a user entry, or an IP address (complete or wildcarded) for a machine entry:

- A username can include up to 63 numbers and lowercase letters, but it is best to make it shorter than eight characters, because many application programs cannot handle longer names. Also, it is best not to include shell metacharacters or other punctuation marks. In particular, the colon (`:`) and at-sign (`@`) characters are not acceptable. The period is generally used only in special administrative names, to separate the username and an *instance*, as in the example `pat.admin`.
- A machine identifier is its IP address in dotted decimal notation (for example, 192.12.108.240), or a wildcard notation that represents a set of IP addresses (a group of machines on the same network). The following are acceptable wildcard formats. The letters `W`, `X`, `Y` and `Z` each represent an actual number from the range 1 through 255.
  - W.X.Y.Z represents a single machine, for example `192.12.108.240`.
  - W.X.Y.0 matches all machines whose IP addresses start with the first three numbers. For example, `192.12.108.0` matches both `192.12.108.119` and `192.12.108.120`, but does not match `192.12.105.144`.
  - W.X.0.0 matches all machines whose IP addresses start with the first two numbers. For example, the address `192.12.0.0` matches both `192.12.106.23` and `192.12.108.120`, but does not match `192.5.30.95`.
  - W.0.0.0 matches all machines whose IP addresses start with the first number in the specified address. For example, the address `192.0.0.0` matches both `192.5.30.95` and `192.12.108.120`, but does not match `138.255.63.52`.

  Do not define a machine entry with the name `0.0.0.0` to match every machine. The system:anyuser group is equivalent.

-id <*user id*>+

Specifies an AFS UID for each user or machine entry, rather than allowing the Protection Server to assign it. Provide a positive integer.

If this argument is used and the -name argument names multiple new entries, it is best to provide an equivalent number of AFS UIDs. The first UID is assigned to the first entry, the second to the second entry, and so on. If there are fewer UIDs than entries, the Protection Server assigns UIDs to the unmatched entries based on the `max user id` counter. If there are more UIDs than entries, the excess UIDs are ignored. If any of the UIDs is greater than the current value of the `max user id` counter, the counter is reset to that value.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The command generates the following string to confirm creation of each user:

```
User &lt;name&gt; has id &lt;id&gt;
```

Examples

The following example creates a Protection Database entry for the user `johnson`.

```
% pts createuser -name johnson
```

The following example creates three wildcarded machine entries in the ABC Corporation cell. The three entries encompass all of the machines on the company's networks without including machines on other networks:

```
% pts createuser -name 138.255.0.0 192.12.105.0 192.12.106.0
```

Privilege Required

The issuer must belong to the system:administrators group.

See Also

"pts(1)" on page 173, "pts_listmax(1)" on page 205, "pts_setmax(1)" on page 225

Copyright

pts delete

Name

`pts delete` — Deletes a Protection Database entry

Synopsis

pts delete -nameorid <*user or group name or id*>+ [-cell <*cell name*>] [-noauth] [-force] [-help]

pts d -na <*user or group name or id*>+ [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts delete command removes each entry specified by the -nameorid argument from the Protection Database. Deleting entries affects other parts of the system in various ways:

- Deleted users and groups still appear on access control lists (ACLs), but are listed by AFS UID or GID rather than by name, because there is no longer an associated name to which to translate the ID. To remove these obsolete entries from ACLs, use the fs cleanacl command.
- Deleting a user or machine's entry removes it from the membership list of any group to which it belonged.
- Deleting a group entry removes it from the membership list of any user or machine entry that belonged to the group, and also increments the group-creation quota of the group's creator by one, even if the creator no longer owns the group.

To remove a user or machine from a group without actually deleting the entry, use the pts removeuser command.

Options

-nameorid <*user or group name or ID*>+

Specifies the name or AFS UID of each user, the name or AFS GID of each group, or the IP address (complete or wildcard-style) or AFS UID of each machine entry to delete. It is acceptable to mix users, machines, and groups on the same command line, as well as names (IP addresses for machines) and IDs. Precede the GID of each group with a hyphen to indicate that it is negative.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example deletes the user entries `pat` and `terry`:

```
% pts delete pat terry
```

The following example deletes the Protection Database entry of the group with AFS GID -215.

```
% pts delete -215
```

Privilege Required

The issuer must belong to the system:administrators group to delete user and machine entries. To delete group entries, the issuer must either own the group or belong to the system:administrators group.

See Also

"fs_cleanacl(1)" on page 55, "pts(1)" on page 173, "pts_removeuser(1)" on page 215

Copyright

pts examine

Name

`pts examine` — Displays a Protection Database entry

Synopsis

pts examine -nameorid <*user or group name or id*>+ [-cell <*cell name*>] [-noauth] [-force] [-help]

pts e -na <*user or group name or id*>+ [-c <*cell name*>] [-no] [-f] [-h]

pts check -na <*user or group name or id*>+ [-c <*cell name*>] [-no] [-f] [-h]

pts che -na <*user or group name or id*>+ [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts examine command displays information from the Protection Database entry of each user, machine or group specified by the -nameorid argument.

Options

-nameorid <*user or group name or id*>+

Specifies the name or AFS UID of each user, the name or AFS GID of each group, or the IP address (complete or wildcard-style) or AFS UID of each machine for which to display the Protection Database entry. It is acceptable to mix users, machines, and groups on the same command line, as well as names (IP addresses for machines) and IDs. Precede the GID of each group with a hyphen to indicate that it is negative.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output for each entry consists of two lines that include the following fields:

Name

The contents of this field depend on the type of entry:

- For a user entry, it is the username that the user types when authenticating with AFS.
- For a machine entry, it is either the IP address of a single machine in dotted decimal format, or a wildcard notation that represents a group of machines on the same network. See the pts createuser reference page for an explanation of the wildcard notation.
- For a group entry, it is one of two types of group name. If the name has a colon between the two parts, it represents a regular group and the part before the prefix reflects the group's owner. A prefix-less group does not have the owner field or the colon. For more details on group names, see the pts creategroup reference page.

id

A unique number that the AFS server processes use to identify AFS users, machines and groups. AFS UIDs for user and machine entries are positive integers, and AFS GIDs for group entries are negative integers. AFS UIDs and GIDs are similar in function to the UIDs and GIDs used in local file systems such as UFS, but apply only to AFS operations.

owner

The user or group that owns the entry and thus can administer it (change the values in most of the fields displayed in the output of this command), or delete it entirely. The Protection Server automatically records the system:administrators group in this field for user and machine entries at creation time.

creator

The user who issued the pts createuser or pts creategroup command to create the entry. This field serves as an audit trail, and cannot be changed.

membership

An integer that for users and machines represents the number of groups to which the user or machine belongs. For groups, it represents the number of group members.

flags

A string of five characters, referred to as *privacy flags*, which indicate who can display or administer certain aspects of the entry.

s

Controls who can issue the pts examine command to display the entry.

o

Controls who can issue the pts listowned command to display the groups that a user or group owns.

m

Controls who can issue the pts membership command to display the groups a user or machine belongs to, or which users or machines belong to a group.

a

Controls who can issue the pts adduser command to add a user or machine to a group. It is meaningful only for groups, but a value must always be set for it even on user and machine entries.

r

Controls who can issue the pts removeuser command to remove a user or machine from a group. It is meaningful only for groups, but a value must always be set for it even on user and machine entries.

Each flag can take three possible types of values to enable a different set of users to issue the corresponding command:

- A hyphen (-) designates the members of the system:administrators group and the entry's owner. For user entries, it designates the user in addition.
- The lowercase version of the letter applies meaningfully to groups only, and designates members of the group in addition to the individuals designated by the hyphen.
- The uppercase version of the letter designates everyone.

For example, the flags `SOmar` on a group entry indicate that anyone can examine the group's entry and display the groups that it owns, and that only the group's members can display, add, or remove its members.

The default privacy flags for user and machine entries are `S----`, meaning that anyone can display the entry. The ability to perform any other functions is restricted to members of the system:administrators group and the entry's owner (as well as the user for a user entry).

The default privacy flags for group entries are `S-M--`, meaning that all users can display the entry and the members of the group, but only the entry owner and members of the system:administrators group can perform other functions.

group quota

The number of additional groups the user is allowed to create. The pts createuser command sets it to 20 for both users and machines, but it has no meaningful interpretation for a machine, because it is not possible to authenticate as a machine. Similarly, it has no meaning in group entries and the pts creategroup command sets it to 0 (zero); do not change this value.

Examples

The following example displays the user entry for `terry` and the machine entry `158.12.105.44`.

```
% pts examine terry 158.12.105.44
Name: terry, id: 1045, owner: system:administrators, creator: admin,
  membership: 9, flags: S----, group quota: 15.
Name: 158.12.105.44, id: 5151, owner: system:administrators,
  creator: byu, membership: 1, flags: S----, group quota: 20.
```

The following example displays the entries for the AFS groups with GIDs -673 and -674.

```
% pts examine -673 -674
Name: terry:friends, id: -673, owner: terry, creator: terry,
  membership: 5, flags: S-M--, group quota: 0.
Name: smith:colleagues, id: -674, owner: smith, creator: smith,
  membership: 14, flags: SOM--, group quota: 0.
```

Privilege Required

The required privilege depends on the setting of the first privacy flag in the Protection Database entry of each entry specified by the -nameorid argument:

- If it is lowercase `s`, members of the system:administrators group and the user associated with a user entry can examine it, and only members of the system:administrators group can examine a machine or group entry.
- If it is uppercase `S`, anyone who can access the cell's database server machines can examine the entry.

See Also

"pts(1)" on page 173, "pts_adduser(1)" on page 177, "pts_chown(1)" on page 183, "pts_creategroup(1)" on page 185, "pts_createuser(1)" on page 189, "pts_listowned(1)" on page 207, "pts_membership(1)" on page 211, "pts_removeuser(1)" on page 215, "pts_rename(1)" on page 217, "pts_setfields(1)" on page 221

Copyright

pts help

Name

`pts help` — Displays help for pts commands

Synopsis

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

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

Description

The pts help command displays the complete online help entry (short description and syntax statement) for each command operation code specified by the -topic argument. If the -topic argument is omitted, the output includes the first line (name and short description) of the online help entry for every pts command.

To list every pts command whose name or short description includes a specified keyword, use the pts apropos command.

Options

-topic <*help string*>+

Indicates each command for which to display the complete online help entry. Omit the pts part of the command name, providing only the operation code (for example, specify `membership`, not `pts membership`). If this argument is omitted, the output briefly describes every pts command.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The first line names the command and briefly describes its function.
- The second line lists aliases for the command, if any.
- The final line, which begins with the string `Usage`, lists the command's options in the prescribed order. Online help entries use the same symbols (for example, brackets) as the reference pages in this document.

Examples

The following command 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 &lt;user or group name or id&gt;+
[-cell &lt;cell name&gt;] [-noauth] [-force] [-help]
```

Privilege Required

None

See Also

"pts(1)" on page 173, "pts_apropos(1)" on page 181

Copyright

pts listentries

Name

`pts listentries` — Displays all users or groups in the Protection Database

Synopsis

pts listentries [-users] [-groups] [-cell <*cell name*>] [-noauth] [-force] [-help]

pts liste [-u] [-g] [-c <*cell name*>] [-n] [-f] [-h]

Description

The pts listentries command displays the name and AFS ID of all Protection Database entries of the indicated type. It also displays the AFS ID of each entry's owner and creator.

To display all user and machine entries, either include the -users flag or omit both it and the -groups flag. To display all group entries, include the -groups flag. To display all entries, provide both flags.

Options

-users

Displays user and machine entries.

-groups

Displays group entries.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output includes a line for each entry, with information in four columns that have the following headers:

Name

The entry's name.

ID

The entry's AFS ID (AFS UID for a user or machine, negative AFS GID for a group).

Owner

The AFS ID of the user or group that owns the entry.

Creator

The AFS ID of the user who created the entry (the system:administrators group is listed as the creator of the entry for anonymous and the system groups, but it is not otherwise possible for a group to create groups).

In general, the entries appear in the order in which they were created.

Examples

The following example displays both user and group entries.

```
% pts listentries -users -groups
Name                          ID  Owner Creator
system:administrators       -204   -204    -204
system:anyuser              -101   -204    -204
system:authuser             -102   -204    -204
anonymous                  32766   -204    -204
admin                          1   -204   32766
pat                          100   -204       1
smith                        101   -204       1
pat:friends                 -206    100     100
staff                       -207   -204       1
```

Privilege Required

The issuer must belong to the system:administrators group.

See Also

"pts(1)" on page 173, "pts_creategroup(1)" on page 185, "pts_createuser(1)" on page 189, "pts_examine(1)" on page 195

Copyright

pts listmax

Name

`pts listmax` — Displays the max user id and max group id counters

Synopsis

pts listmax [-cell <*cell name*>] [-noauth] [-force] [-help]

pts listm [-c <*cell name*>] [-n] [-f] [-h]

Description

The pts listmax command displays the values of the `max user id` and `max group id` counters, which the Protection Server uses to track the AFS user IDs (AFS UIDs) it allocates to new users or machines, and the AFS group IDs (AFS GIDs) it allocates to new groups, respectively. When an administrator next issues the pts createuser command and does not include the -id argument, the new user or machine receives an AFS UID one greater than the `max user id` counter, and when a user issues the pts creategroup command and does not include the -id argument, the new group receives an AFS UID one less (more negative) than the `max group id` counter.

To reset one or both counters, members of the system:administrators group can issue the pts setmax command.

Options

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The command displays the counters in the following format:

```
Max user id is &lt;user_counter&gt; and max group id is &lt;group_counter&gt;.
```

Examples

The following example displays the output of this command:

```
% pts listmax
Max user name is 1271 and max group id is -382.
```

Privilege Required

None

See Also

"pts(1)" on page 173, "pts_setmax(1)" on page 225

Copyright

pts listowned

Name

`pts listowned` — Show the Protection Database groups owned by a user or group

Synopsis

pts listowned -nameorid <*user or group name or id*>+ [-cell <*cell name*>] [-noauth] [-force] [-help]

pts listo -na <*user or group name or id*>+ [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts listowned command lists the groups owned by each user or group specified by the -nameorid argument.

To list any *orphaned group*s, whose owners have themselves been deleted from the Protection Database, provide a value of `0` (zero) for the -nameorid argument. To change the owner to a user or group that still exists, use the pts chown command.

Options

-nameorid <*user or group name or id*>+

Specifies the name or AFS UID of each user, or the name or AFS GID of each group, for which to display the list of owned groups. It is acceptable to mix users and groups on the same command line, as well as names and IDs. Precede the GID of each group with a hyphen to indicate that it is negative.

A value of 0 (zero) lists group entries for groups whose owners no longer have entries in the Protection Database.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of the output indicates the name and AFS UID or AFS GID of each user or group for which ownership information is requested, in the following format:

```
Groups owned by &lt;name&gt; (id: &lt;ID&gt;) are:
```

A list of groups 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, only the header line appears.

The following error message appears if the issuer is not privileged to view ownership information. By default, for both user and group entries the second privacy flag is the hyphen, which denies permission to anyone other than the user (for a user entry) and the members of the system:administrators group.

```
pts: Permission denied so failed to get owner list for &lt;name&gt; (id: &lt;ID&gt;)
```

Examples

The following example lists the groups owned by user terry and shows that the group `terry:friends` does not own any groups:

```
% pts listowned terry terry:friends
Groups owned by terry (id: 1045) are:
  terry:friends
  terry:project1
  terry:project2
Groups owned by terry:friends (id: -673) are:
```

Privilege Required

The required privilege depends on the setting of the second privacy flag in the Protection Database entry of each user or group indicated by the -nameorid argument (use the pts examine command to display the flags):

- If it is the hyphen and the -nameorid argument specifies a group, only the members of the system:administrators group and the owner of a group can list the groups it owns.
- If it is the hyphen and the -nameorid argument specifies a user, only the members of the system:administrators group and the associated user can list the groups he or she owns.
- If it is uppercase letter `O`, anyone who can access the cell's database server machines can list the groups owned by this user or group.

See Also

"pts(1)" on page 173, "pts_chown(1)" on page 183, "pts_examine(1)" on page 195, "pts_setfields(1)" on page 221

Copyright

pts membership

Name

`pts membership` — Displays the membership list for a user or group

Synopsis

pts membership -nameorid <*user or group name or id*>+ [-cell <*cell name*>] [-noauth] [-force] [-help]

pts m -na <*user or group name or id*>+ [-c <*cell name*>] [-no] [-f] [-h]

pts groups -na <*user or group name or id*>+ [-c <*cell name*>] [-no] [-f] [-h]

pts g -na <*user or group name or id*>+ [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts membership command lists the groups to which each user or machine specified by the -nameorid argument belongs, or lists the users and machines that belong to each group specified by the -nameorid argument.

It is not possible to list the members of the system:anyuser or system:authuser groups, and they do not appear in the list of groups to which a user belongs.

To add users or machine to groups, use the pts adduser command; to remove them, use the pts removeuser command.

Options

-nameorid <*user or group name or id*>+

Specifies the name or AFS UID of each user entry, the IP address (complete or wildcard-style) or AFS UID of each machine entry, or the name or AFS GID of each group, for which to list group membership. It is acceptable to mix users, machines, and groups on the same command line, as well as names and IDs. Precede the GID of each group with a hyphen to indicate that it is negative.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Output

For each user and machine, the output begins with the following header line, followed by a list of the groups to which the user or machine belongs:

```
Groups &lt;name&gt; (id: &lt;AFS UID&gt;) is a member of:
```

For each group, the output begins with the following header line, followed by a list of the users and machines who belong to the group:

```
Members of &lt;group_name&gt; (id: &lt;AFS GID&gt;) are:
```

Examples

The following example lists the groups to which the user `pat` belongs and the members of the group `smith:friends`. Note that third privacy flag for the `pat` entry was changed from the default hyphen to enable a non-administrative user to obtain this listing.

```
% pts membership 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
```

Privilege Required

The required privilege depends on the setting of the third privacy flag in the Protection Database entry of each user or group indicated by the -nameorid argument (use the pts examine command to display the flags):

- If it is the hyphen and the -nameorid argument specifies a user, only the associated user and members of the system:administrators group can list the groups to which the user belongs.
- If it is the hyphen and the -nameorid argument specifies a machine, only the members of the system:administrators group can list the groups to which the machine belongs.

- If it is the hyphen and the -nameorid argument specifies a group, only the owner of the group and members of the system:administrators group can list the members of the group.
- If it is lowercase `m` and the -nameorid argument specifies a user or machine entry, the meaning is equivalent to the hyphen.
- If it is lowercase `m` and the -nameorid argument specifies a group, members of the group can also list the other members.
- If it is uppercase `M`, anyone who can access the cell's database server machines can list group memberships.

See Also

"pts(1)" on page 173, "pts_adduser(1)" on page 177, "pts_examine(1)" on page 195, "pts_removeuser(1)" on page 215, "pts_setfields(1)" on page 221

Copyright

pts removeuser

Name

`pts removeuser` — Removes a user from a Protection Database group

Synopsis

pts removeuser -user <*user name*>+ -group <*group name*>+ [-cell <*cell name*>] [-noauth] [-force] [-help]

pts rem -u <*user name*>+ -g <*group name*>+ [-c <*cell name*>] [-n] [-f] [-h]

Description

The pts removeuser command removes each user or machine named by the -user argument from each group named by the -group argument.

To add users to a group, use the pts adduser command. To list group membership, use the pts membership command. To remove users from a group and delete the group's entry completely in a single step, use the pts delete command.

Cautions

AFS compiles each user's group membership as he or she authenticates. Any users who have valid tokens when they are removed from a group retain the privileges extended to that group's members until they discard their tokens or reauthenticate.

Options

-name <*user name*>+

Specifies the name of each user entry or the IP address (complete or wildcard-style) of each machine entry to remove.

-group <*group name*>+

Names each group from which to remove members.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example removes user smith from the groups `staff` and `staff:finance`. Note that no switch names are necessary because only a single instance is provided for the first argument (the username).

```
% pts removeuser smith staff staff:finance
```

The following example removes three machine entries, which represent all machines in the ABC Corporation network, from the group `bin-prot`:

```
% pts removeuser -user 138.255.0.0 192.12.105.0 192.12.106.0 -group bin-prot
```

Privilege Required

The required privilege depends on the setting of the fifth privacy flag in the Protection Database for the group named by the -group argument (use the pts examine command to display the flags):

- If it is the hyphen, only the group's owner and members of the system:administrators group can remove members.
- If it is lowercase `r`, members of the group can also remove other members.

(It is not possible to set the fifth flag to uppercase `R`.)

See Also

"pts(1)" on page 173, "pts_adduser(1)" on page 177, "pts_examine(1)" on page 195, "pts_membership(1)" on page 211, "pts_setfields(1)" on page 221

Copyright

pts rename

Name

`pts rename` — Changes the name of a Protection Database entry

Synopsis

pts rename -oldname <*old name*> -newname <*new name*> [-cell <*cell name*>] [-noauth] [-force] [-help]

pts ren -o <*old name*> -ne <*new name*> [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts rename command changes the name of the user, machine, or group entry specified by the -oldname argument to the name specified by the -newname argument. It is not possible to change a user or machine entry's name to look like a regular group entry's name (have a colon in it).

Members of the system:administrators group can change a regular group name into a prefix-less name 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 field of the new name (the part before the colon) must correctly reflect the group's owner.

Changing a regular group's owner with the pts chown command automatically changes the owner field (the part before the colon) of the group's name, but does not change the owner field of any groups owned by the group. Use this command to rename those groups to a form that accurately reflects their ownership.

Cautions

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. When using this command to change a user name, also change the names of all related entities to maintain consistency. For instructions, see the chapter on user accounts in the *IBM AFS Administration Guide*.

Options

-oldname <*old name*>

Specifies the current full name of the entry.

-newname <*new name*>

Specifies the new full name for the entry. For regular groups, the owner field (the part before the colon) of the new name must reflect the actual ownership of the group.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example changes the name of the group staff, owned by the privileged user `admin`, to `admin:staff`:

```
% pts rename -oldname staff -newname admin:staff
```

The following example changes the name of the group `admin:finance` to the group `finance`. The issuer must belong to the system:administrators group.

```
% pts rename -oldname admin:finance -newname finance
```

Privilege Required

To change a regular group name to a prefix-less name or vice versa, or to change a user or machine entry’s name, the issuer must belong to the system:administrators group.

To change a group name to a new name of the same type (regular or prefix-less), the issuer must own the group or belong to the system:administrators group.

See Also

"pts(1)" on page 173, "pts_chown(1)" on page 183

Copyright

pts setfields

Name

`pts setfields` — Sets privacy flags or quota for a Protection Database entry

Synopsis

pts setfields -nameorid <*user or group name or id*>+ [-access <*set privacy flags*>] [-groupquota <*set limit on group creation*>] [-cell <*cell name*>] [-noauth] [-force] [-help]

pts setf -na <*user or group name or id*>+ [-a <*set privacy flags*>] [-g <*set limit on group creation*>] [-c <*cell name*>] [-no] [-f] [-h]

Description

The pts setfields command sets the group-creation quota, the privacy flags, or both, associated with each user, machine, or group entry specified by the -nameorid argument.

To examine the current quota and privacy flags, use the pts examine command.

Cautions

Changing a machine or group's group-creation quota is allowed, but not recommended. The concept is meaningless for machines and groups, because it is impossible to authenticate as a group or machine.

Similarly, some privacy flag settings do not have a sensible interpretation. "OPTIONS" on page specifies the appropriate settings.

Options

-nameorid <*user or group name or id*>+

Specifies the name or AFS UID of each user, the IP address (complete or wildcard-style) of each machine, or the name or AFS GID of each machine for which to set privacy flags or group-creation quota. It is acceptable to mix users, machines, and groups on the same command line, as well as names (IP addresses for machines) and IDs. Precede the GID of each group with a hyphen to indicate that it is negative.

-access <*privacy flags*>

Specifies the privacy flags to apply to each entry. Provide a string of five characters, one for each of the permissions. If this option is omitted, the current setting remains unchanged.

Set each flag to achieve the desired combination of permissions. If the following list does not mention a certain setting, it is not acceptable. For further discussion of the privacy flags, see "pts_examine(1)" on page 195.

- The first flag determines who can use the pts examine command to display information from a user, machine or group's Protection Database entry.
  - Set it to lowercase `s` to permit the members of the system:administrators group to display a user, machine, or group entry, and the associated user to display a user entry.
  - Set it to uppercase `S` to permit anyone who can access the cell's database server machines to display a user, machine, or group entry.

- The second flag determines who can use the pts listowned command to list the groups that a user or group owns.
  - Set it to the hyphen (`-`) to permit the members of the system:administrators group and a user to list the groups he or she owns, or to permit the members of the system:administrators group and a group's owner to list the groups that a group owns.
  - Set it to uppercase letter `O` to permit anyone who can access the cell's database server machines to list the groups owned by a machine or group entry.

- The third flag determines who can use the pts membership command to list the groups to which a user or machine belongs, or the users and machines that belong to a group.
  - Set it to the hyphen (`-`) to permit the members of the system:administrators group and a user to list the groups he or she belongs to, to permit the members of the system:administrators group to list the groups a machine belongs to, or to permit the members of the system:administrators group and a group's owner to list the users and machines that belong to it.
  - Set it to lowercase `m` to permit members of a group to list the other members. (For user and machine entries, this setting is equivalent to the hyphen.)
  - Set it to uppercase `M` to permit anyone who can access the cell's database server machines to list membership information for a user, machine or group.

- The fourth flag determines who can use the pts adduser command to add users and machines as members of a group. This flag has no sensible interpretation for user and machine entries, but must be set nonetheless, preferably to the hyphen.
  - Set it to the hyphen (`-`) to permit the members of the system:administrators group and the owner of the group to add members.
  - Set it to lowercase `a` to permit members of a group to add other members.
  - Set it to uppercase `A` to permit anyone who can access the cell's database server machines to add members to a group.

- The fifth flag determines who can use the pts removeuser command to remove users and machines from membership in a group. This flag has no sensible interpretation for user and machine entries, but must be set nonetheless, preferably to the hyphen.

- Set it to the hyphen (`-`) to permit the members of the system:administrators group and the owner of the group to remove members.
- Set it to lowercase `r` to permit members of a group to remove other members.

-groupquota <*group creation quota*>

Specifies the number of additional groups a user can create (it does not matter how many he or she has created already). Do not include this argument for a group or machine entry.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example changes the privacy flags on the group `operators`, retaining the default values of the first, second and third flags, but setting the fourth and fifth flags to enable the group's members to add and remove other members.

```
% pts setfields -nameorid operators -access S-Mar
```

The following example changes the privacy flags and sets group quota on the user entry `admin`. It retains the default values of the first, fourth, and fifth flags, but sets the second and third flags, to enable anyone to list the groups that `admin` owns and belongs to. Users authenticated as `admin` can create an additional 50 groups.

```
% pts setfields -nameorid admin -access SOM-- -groupquota 50
```

Privilege Required

To edit group entries or set the privacy flags on any type of entry, the issuer must own the entry or belong to the system:administrators group. To set group-creation quota on a user entry, the issuer must belong to the system:administrators group.

See Also

"pts(1)" on page 173, "pts_adduser(1)" on page 177, "pts_examine(1)" on page 195, "pts_listowned(1)" on page 207, "pts_membership(1)" on page 211, "pts_removeuser(1)" on page 215

Copyright

pts setmax

Name

`pts setmax` — Sets the value of the max group id or max user id counter

Synopsis

pts setmax [-group <*group max*>] [-user <*user max*>] [-cell <*cell name*>] [-noauth] [-force] [-help]

pts setm [-g *group max*>] [-u <*user max*>] [-c <*cell name*>] [-n] [-f] [-h]

Description

The pts setmax command sets the value of one or both counters that track the IDs the Protection Server allocates to new users, machines, or groups: the `max user id` counter for the AFS user IDs (AFS UIDs) assigned to users and machines, and the `max group id` counter for the AFS group IDs (AFS GIDs) assigned to groups.

Use the pts listmax command to display the current value of both counters.

Options

-group <*group max*>

Sets the `max group id` counter. Precede the value with a hyphen to indicate that it is negative. When an administrator next uses the pts creategroup command to create a group entry and does not include that command's -id argument, the Protection Server assigns the group an AFS GID one less (more negative) than this value.

-user <*user max*>

Sets the `max user id` counter. When an administrator next uses the pts createuser command to create a user or machine entry and does not include that command's -id argument, the Protection Server assigns the group an AFS UID one greater than this value.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "pts(1)" on page 173.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "pts(1)" on page 173.

-force

Enables the command to continue executing as far as possible when errors or other problems occur, rather than halting execution at the first error.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command sets the `max group id` counter to -500 and the `max user id` counter to 1000.

```
% pts setmax -group -500 -user 1000
```

Privilege Required

The issuer must belong to the system:administrators group.

See Also

"pts(1)" on page 173, "pts_creategroup(1)" on page 185, "pts_createuser(1)" on page 189, "pts_listmax(1)" on page 205

Copyright

rxdebug

Name

`rxdebug` — Provides debugging trace of Rx activity

Synopsis

rxdebug -servers <*server machine*> [-port <*IP port*>] [-nodally] [-allconnections] [-rxstats] [-onlyserver] [-onlyclient] [-onlyport <*show only port*>] [-onlyhost <*show only host*>] [-onlyauth <*show only auth level*>] [-version] [-noconns] [-peers] [-help]

rxdebug -s <*server machine*> [-po <*IP port*>] [-nod] [-a] [-r] [-onlys] [-onlyc] [-onlyp <*show only port*>] [-onlyh <*show only host*>] [-onlya <*show only auth level*>] [-v] [-noc] [-pe] [-h]

Description

The rxdebug command provides a trace of Rx activity for the server or client machine named by the -servers argument. Rx is AFS's proprietary remote procedure call (RPC) protocol, so this command enables the issuer to check the status of communication between the Cache Manager or an AFS server process (as specified with the -port argument) on the machine and one or more processes on other machines.

Options

-servers <*server machine*>

Specifies the machine that is running the Cache Manager or server process for which to trace Rx activity. Provide the machine's IP address in dotted decimal format, its fully qualified host name (for example, `fs1.abc.com`), or the shortest abbreviated form of its host name that distinguishes it from other machines. Successful use of an abbreviated form depends on the availability of a name resolution service (such as the Domain Name Service or a local host table) at the time the command is issued.

-port <*IP port*>

Specifies the process for which to trace Rx activity. Omit this argument to specify the File Server (fileserver process), or provide one of the following values:

`7000` for the File Server (fileserver process)

`7001` for the Cache Manager (specifically, its callback interface)

`7002` for the Protection Server (ptserver process)

`7003` for the Volume Location (VL) Server (vlserver process)

`7004` for the Authentication Server (kaserver process)

`7005` for the Volume Server (volserver process)

`7007` for the BOS Server (bosserver process)

`7008` for the Update Server (upserver process)

`7009` for the NFS / AFS Translator's rmtsysd daemon

`7021` for the Backup Server (buserver process)

Finally, specify `7025` through `65535` for the Backup Tape Coordinator (butc process) that has the port offset number derived by subtracting 7025 from this value.

-nodally

Produces output only for connections that are not in dally mode.

-allconnections

Produces output for all connections, even inactive ones. By default, the output includes information only for connections that are active or in dally mode when the rxdebug command is issued.

-rxstats

Produces detailed statistics about Rx history and performance (for example, counts of the number of packets of various types the process has read and sent, calculations of average and minimum roundtrip time, and so on).

-onlyserver

Produces output only for connections in which the process designated by the -port argument is acting as the server.

-onlyclient

Produces output only for connections in which the process designated by the -port argument is acting as the client.

-onlyport <*port*>

Produces output only for connections between the process designated by the -port argument and the specified port on any another machine. Use the same port identifiers as for the -port argument.

-onlyhost <*host*>

Produces output only for connections between the process designated by the -port argument and any process on the specified machine. To identify the machine, use the same notation as for the -servers argument.

-onlyauth

Produces output only for connections that are using the specified authentication level. Provide one of the following values:

auth

Cconnections at authentication level rxkad_auth

clear

Connections at authentication level rxkad_clear

crypt

Connections at authentication level rxkad_crypt

none

Unauthenticated connections (equivalents are `null`, `noauth`, and `unauth`)

-version

Reports the AFS build level of the binary file for the process designated by the -port argument (or of the kernel extensions file for port 7001, the Cache Manager's callback interface). Any other options combined with this one are ignored.

-noconns

Produces only the standard statistics that begin the output produced by every option (other than -version), without reporting on any connections. Any other options combined with this one are ignored.

-peers

Outputs information from the *peer structure* maintained for each port on another machine to which the process designated by the -port argument has a connection. There is information about roundtrip time and numbers of packets sent and received, for example.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If any options other than -version or -help are provided, the output written to the standard output stream begins with basic statistics about packet usage and availability, how many calls are waiting for a thread, how many threads are free, and so on (this is the only information provided by the -noconns flag). Adding other options produces additional information as described in "OPTIONS" on page . The output is intended for debugging purposes and is meaningful to someone familiar with the implementation of Rx.

Privilege Required

None.

See Also

"afsd(8)" on page 523, "bosserver(8)" on page 759, "buserver(8)" on page 763, "butc(8)" on page 767, "fileserver(8)" on page 771, "kaserver(8)" on page 859, "ptserver(8)" on page 875, "upclient(8)" on page 885, "upserver(8)" on page 889, "vlserver(8)" on page 915, "volserver(8)" on page 925

Copyright

rxgen

Name

`rxgen` — Stub generator for the Rx remote procedure call package

Synopsis

rxgen [-h | -c | -C | -S | -r] [-dkpR] [-I *dir*] [-P *prefix*] [-o *outfile*] [*infile*]

rxgen -s *transport* [-o *outfile*] [*infile*]

rxgen -l [-o *outfile*] [*infile*]

rxgen -m [-o *outfile*] [*infile*]

Description

rxgen is a tool that generates C code to implement the Rx RPC protocol; it takes as input a description of an application interface similar to C and produces a number of server and/or client stub routines to be linked with RPC-based programs. These stubs allow programs to invoke remote procedures through local procedure calls. rxgen is an extension of Sun's rpcgen (version 3.9) and retains full rpcgen functionality (at least as of that version). Please refer to rpcgen(1) for more details on the Sun's RPC specific flags, and to the RPC programming guide regarding the RPC language along with useful examples.

Options

rxgen operates in several different modes. The generated output files can be produced individually (using one of -h, -c, -C, or -S) or collectively. All output files are created when the default is used (i.e., no options), or the output is limited to the server stubs (-C and -S) when the -r flag is used. The following describes the types of generated output files (for simplicity, *filename* refers to the main output filename):

-h

Generate C data definitions (a header file) from standard RPCL definitions (default extension: *filename*.h).

-c

Compile the XDR routines required to serialize the protocol described by RPCL. Generate XDR routines for all declarations (default extension: *filename*.xdr.c).

-C

Generate all the client-side stub routines (default extension: *filename*.cs.c). Calling a routine in this file will cause the arguments to be packed up and sent via Rx (or R).

-S

Generate all the server-side stub routines (default extension: *filename*.ss.c). Arguments are unpacked, and the corresponding server routine is called.

-r

Generate the two default extension files produced by the -C and -S options.

The following options can be used on any combination of rxgen calls:

-R

Generate code for the older \R protocol, as opposed to Rx, which is the default.

-k

Must be specified when the generated code is intended to be used by the kernel; special "includes" and other specifics are produced when the target output is for the kernel.

-p

Package combination flag: when multiple packages are included within a single specification file, a single Execute Request routine will be used for all of them as a result of this flag. The default is to generate individual Execute Request stubs for each package.

-I *dir*

Similar to the -I flag in the C compiler (cc). This flag is passed to the pre-processor (cpp) so that directory *dir* is searched before the standard lookup list for #include files. As expected, multiple -I flags can be used simultaneously.

-P *prefix*

The *prefix* string following this switch is prepended to all generated output files; useful when multiple runs want to produce different versions of the same interface (say, kernel and non-kernel versions).

-d

Debugging mode; only needed when rxgen is to be debugged (say, via dbx).

-o *outfile*

Specify the name of the output file. If none is specified, the standard output is used (-c, -h, -C, and -S modes only). Note that if an output file is specified in a multi-output file option (such as the default, or with option -r), then the *outfile* replaces the name generated by default (which is based on the configuration's main file name).

The -s, -l, and -m options are present only for rpcgen support. See rpcgen(1) for information on their use.

rxgen SYNTAX SUMMARY

```
Specification file:
```

```
    &lt;Package description option&gt; |
    &lt;Prefix description option&gt; |
    &lt;StartingOpcode description option&gt; |
    &lt;SplitPrefix description option&gt; |
    &lt;Procedure description option&gt; |
    &lt;RPCL language description option&gt;


&lt;Package description option&gt;:


    "package" &lt;Package_ident&gt;


&lt;Prefix description option&gt;:


    "prefix" &lt;Prefix_ident&gt;


&lt;StartingOpcode description option&gt;:


    "startingopcode" &lt;constant&gt;


&lt;SplitPrefix description option&gt;:


    "splitprefix" &lt;split options&gt; ";"


&lt;Split options&gt;:


    "IN =" &lt;Start_prefix_ident&gt; "|"
    "OUT =" &lt;End_prefix_ident&gt; "|"
    &lt;Split options&gt;


&lt;Procedure description option&gt;:


    ["proc"] [&lt;Procedure_ident&gt;] [&lt;ServerStub_ident&gt;]
        &lt;Argument list&gt; ["split" | "multi"]
        ["=" &lt;Opcode_ident&gt;] ";"


&lt;Argument list&gt;:


    "(" &lt;Argument definition&gt; &lt;Comma_joined argument&gt; ")"
```

```
&lt;Argument definition&gt;:

    &lt;Direction option&gt; &lt;Standard RPCL type decl&gt; &lt;Arg_ident&gt;
        ["&lt;" &lt;Max_size&gt; "&gt;" | "[" &lt;Max_size&gt; "]"] | NULL

&lt;Comma_joined argument&gt;:

    "," &lt;Argument definition&gt; | NULL

&lt;Direction option&gt;:

    "IN" | "OUT" | "INOUT" | NULL

&lt;Max_size&gt;:

    &lt;constant&gt; | NULL

&lt;Package_ident&gt;:
&lt;Prefix_ident&gt;:
&lt;String_ident&gt;:
&lt;Start_prefix_ident&gt;:
&lt;End_prefix_ident&gt;:
&lt;Procedure_ident&gt;:
&lt;ServerStub_ident&gt;:
&lt;Arg_ident&gt;:
&lt;Opcode_ident&gt;:

    &lt;identifier&gt;

&lt;RPCL language description option&gt;:
&lt;Standard RPCL type decl&gt;:

    Sun’s RPCL language syntax (see rpcgen(1))
```

rxgen COMMANDS

Comments and Preprocessing

The input interface may contain preprocessor directives which are passed through the C preprocessor (i.e. cpp). Since the preprocessor runs on all input files before they are actually

interpreted by rxgen, all cpp directives (#include, #ifdefs, #defines, etc.) are legal and welcomed within an rxgen input file. Of course, none of these preprocessor directives will be included in any of the generated files. To facilitate distinctions between the different types of output files, rxgen defines certain special cpp symbols for use by the rxgen programmer. These are RPC_HDR (defined when compiling into header, *filename*.h, files), RPC_XDR (defined when compiling into xdr, *filename*.xdr.c, files), RPC_CLIENT (defined when compiling into client stubs, *filename*.cs.c, files), and RPC_SERVER (defined when compiling into server stubs, *filename*.ss.c, files).

In addition, rxgen does a little preprocessing of its own. Any line beginning with `%` is passed directly into the output file, uninterpreted by rxgen. For a more heavy en masse dumping of uninterpreted code, it would be adviced to include all such code in an `#include` file and pass it in preceded by `%`. The input interface may also contain any C-style comments which are, of course, ignored. Interpretation is token-based, thus special line-orientation of separate statements is not necessary. rxgen also provides a quite rich and helpful set of error reports, identifying them by exact line location and error type. Also, rxgen will automatically generate #include lines for standard include files, such as *`rx/xdr.h`* and *`rx/rx.h`*, along with the generated header file from this interface.

Prefixing stub procedures

The *package* statement tells rxgen the name of the interface package. It is used for prefixing the naming of all generated stub routines and the execute request procedure. For example:

```
package AFS_
```

causes the execute request procedure to be named AFS_ExecuteRequest (Warning: in the older version an additional _ was appended after the package name to the ExecuteRequest name; thus make sure you don't have an ExecuteRequest interface routine) and a given stub routine, say Fetch, to be actually named AFS_Fetch. Multiple package statements (current maximum size is 10) per configuration are permitted and are useful when multiple sets of interfaces are implemented (see the example at the end). Note that in such cases, use of the -p flag results in the generation of just one ExecuteRequest procedure which recognizes the multiple interfaces and whose name is prefixed by the first package statement. In the default case, independent ExecuteRequest procedures will be created for each packaged group of remote procedure calls.

The *prefix* statement supplies a name to prepend to all calls to remote procedure names in the ExecuteRequest stub routine. It is useful when the server makes RPC calls to other servers (say, for debugging purposes). For example:

```
prefix S
```

causes the name `S` to be prepended to the name of all routines called from the server stubs. The server can then call the original name and get the client stubs.

rxgen procedure declaration

The *proc* statement is the most common (and meaningful) in the rxgen interface. Its syntax description is:

```
[proc] [&lt;proc_name&gt;] [&lt;server_stub&gt;] (&lt;arg&gt;, ..., &lt;arg&gt;)
    [split | multi] [= &lt;opcode&gt;] ;
```

where:

- `proc` is an optional prefix of the procedure statement. This is just a stylistic item and not a required procedure delimiter.
- <proc_name> is the name of the procedure. Note that even the name of the procedure is optional. This only makes sense when the name of the given procedure is identical to the name of the last *package* statement (i.e., `package RCallBack` and the declaration of the `RCallBack` procedure).
- <server_stub>, if present, causes the ExecuteRequest procedure to call that stub instead of the automatically generated stub when a call with that opcode is decoded.
- <opcode> is a constant or symbol that is the opcode for that procedure. One might use the preprocessor features (i.e., #define), the *const* RPC-language feature, or the old good constants as opcodes. Some further evaluation / processing of opcodes is done. Particularly, checks for duplicate and non-existent opcodes are performed, along with checks for "holes" (i.e., gaps in consecutive opcodes) in the opcode sequences. For example, we use the fact that when "holes" in opcodes exist, the ExecuteRequest procedure uses the *case* statement rather than the faster (and smaller, codewise) indexed array method.

  Also, rxgen defines (i.e., appends to the header file) three valuable macros for each package group: <package-name>LOWEST_OPCODE, <package-name>HIGHEST_OPCODE, and <package-name>NUMBER_OPCODES. These may be useful to the rxgen programmer. Also, notice that the *opcode* statement is an optional feature, and can be omitted. In such cases, automatic opcode numbers are generated sequentially, starting from 0.

  One can change the initial opcode number by using the *startingopcode* (for lack of a better name) rxgen command. Its syntax is:

  ```
  startingopcode &lt;constant&gt;
  ```

  where <constant> must be reasonable! Note that one can not mix procedures, some with opcodes and some without, nor allow opcodes after the specification of the *startingopcode* statement. rxgen will complain in all such cases.
- The *argument* entry represents a given parameter of the procedure. Its syntax is:

  ```
  [IN | INOUT | OUT | &lt;null&gt;] &lt;type_decl&gt; &lt;arg_name&gt;
      [&lt;max&gt;|&lt;&gt;|[max]|[]]
  ```

  If the type is an indirect type (i.e., is followed by *), it is assumed that the pointer should be followed one level and the data pointed to is to be transmitted. This should normally be used for all structures / arrays and out parameters. A noticeable exception is when explicit array / structure maximum size is given; since no array-of-pointer declarations are allowed

one should use typedefs to achieve the similar effect. The parameters could be input parameters (preceded by IN), output parameters (preceded by OUT), or input/output parameters (preceded by INOUT). If not specified, then the direction of the previous parameter in the procedure is used. (Note: the first parameter must be preceded by the directional primitive!)

- `split` is a hack to handle stub routines that do things such as file transfers or any other operation that has to exchange information (e.g., length of a file) before the call returns its output parameters. Because of the particular handshake that is involved when doing remote file transfer, we currently break all such calls into two client-side stub routines. The first (with the default prefix of `Begin`) is used to pass all IN and INOUT parameters to the server side. The second (with the default prefix of `End`) is used to get back the INOUT and OUT parameters from the server. Between the two calls, the user is supposed to do the appropriate calls for the file transfer. For example, the following procedure declaration in package AFS_

  ```
  Fetch (IN a, b,INOUT c, OUT d) split = FETCHOPCODE;
  ```

  will roughly generate the two independent client stub routines:

  ```
  BeginAFS_Fetch (IN a, b, c)
  ```

  and

  ```
  EndAFS_Fetch(OUT c, d)
  ```

  The *splitprefix* statement is used to change the default prefix names used by the two client-side stub generated routines when dealing with file transfer-related procedure calls. For example:

  ```
  splitprefix IN=Before_ OUT=After_
  ```

  will cause the naming of the two client stubs for a file transfer-related routine, say Fetch(), to be Before_AFS_Fetch() and After_AFS_Fetch(), respectively.

- The `multi` option is nearly identical to the `split` feature described above. The only significant visible difference is that along with the two client stubs, the standard client stub is also generated. Since the intention is to handle the multi-Rx calls, we need the whole standard procedure stub in the cases where no multi-Rx call of the procedure is performed. A side effect of the `multi` option is the generation of a special macro (i.e., `multi_<Procedure-name>` which passes back as arguments the `Begin` and `End` stubs in the header output file. This macro is used directly by the Rx code when a multi-Rx call of this procedure is performed.

OBSOLETE rxgen FEATURES

Although the following rxgen commands are still in effect, they will soon be removed since there are better alternatives. DO NOT USE THEM!

The *special* statement is a temporary hack used to handle certain inefficiencies of standard xdr routines to handle some user-customized declarations. In particular, this applies to a string pointer specified as part of a declaration. For example,

```
special struct BBS SeqBody;
```

tells rxgen that the entry `SeqBody` in the user-defined BBS xdr routine is a string (note that more than one string can be "special" per structure -- multiple ones are separated by commas); it will thus allocate and de-allocate space properly in the server-generated stubs that contain this structure as an IN or INOUT parameter.

A better alternative to *special* is the *customized* statement, which is simply the `customized` token followed by the regular declaration of a struct based on the RPCL rules. In this case, the declaration will be included in the generated header file (-h option) but no xdr routine will be generated for this structure -- the user will supply this. All pointer entries in this structure will be remembered so when the structure is used as an IN or INOUT in the server stub, no core leaks will occur. For example, consider

```
customized struct CBS {
    long Seqlen;
    char *SeqBody;
}
```

The `xdr_CBS` routine would be provided by the user where during the DECODE xdr opcode, appropriate space for the `SeqBody` string is allocated. Similarly, that space is freed during the FREE xdr opcode.

Note: Old style "Array parameter specifications" are not supported any more.

Examples

In case there are some requirements not available by the current RPC language, one can customize some XDR routines by leaving those data types undefined. For every data type that is undefined, it will be assumed that a routine exists with the name `xdr_` prepended to it. A selected set of rxgen features is presented below, but for a more comprehensive one (unions, complex examples, etc) please refer to the *rpcgen Programming Guide* and *eXternal Data Representation: Sun Technical Notes*.

Typedefs

The RPC typedef statement is identical to the C typedef (i.e. `typedef <declaration>`). By default, most user declarations (i.e. structs, unions, etc) are automatically typedef'ed by rxgen. Since it makes parsing simpler, its usage is recommended by rxgen scripts.

Strings

The C `char *` string convention is kind of ambiguous, since it is usually intended to mean a null-terminated string of characters, but it could also represent a pointer to a single character, a pointer to an array of characters, etc. In the RPC language, a null-terminated string is unambiguously called a "string". Examples,

```
string bigname&lt;&gt;;
string name&lt;MAXNAMELEN&gt;;
typedef string volname&lt;MAXVOLNAME&gt;;
```

Notice that the maximum size of string can be arbitrary (like `bigname` above) or, preferably, or specified in angle brackets (i.e. `name` and `volname` above). In practice, one should always use only bounded strings in interfaces. A sample calling proc using the declarations above would be:

```
GetEntryByName (IN volname name,
    OUT struct vldbentry *entry) = VL_GETENTRYBYNAME;
```

or, of course,

```
GetEntryByName (IN string volname&lt;MAXVOLNAME&gt;,
    OUT struct vldbentry *entry) = VL_GETENTRYBYNAME;
```

It is very important for the user to understand when the string parameters should be allocated and / or freed by the his / her client and / or server programs. A short analysis on string parameters handling follows (note that a similar method is used for the handling of variable length arrays as it will be shown later on):

- In the client side: IN and INOUT string parameters are the programmer's responsibility and should be allocated (static or via malloc) before calling the rpc and freed (if malloc was used) after the rpc's return in the user's client program; of course, for INOUT parameters, the returned string can't be bigger than the malloced input string.

  OUT string parameters are automatically malloced (based on the length of the returned string and not the maxsize) by the rxgen client stubs (in *filename*.cs.c) and must be freed by the client program; admittedly, this could be somewhat confusing since the user needs to free something that he / she didn't allocate.}
- In the server side: IN and INOUT string parameters are automatically malloced (based on the size of incoming strings) by the rxgen server stubs (in *filename*.ss.c) before they are passed to the user's server procedure; that space is automatically freed just before the rxgen server stub returns; therefore the user need not do anything special for IN and INOUT string parameters.

  OUT string parameters must be malloced by the user's server procedure (i.e. null pointer is passed to it by the rxgen server stub) and it is automatically freed at the end of the rxgen server stub. Like in the client side, the OUT parameters are somewhat unorthodox (i.e. the server routine must malloc a string without ever freeing it itself; this is done by the rxgen server stub).

Note that for INOUT and OUT string parameters, in both the client and server sides their arguments must be char of pointers (i.e. char **).

Pointers

Pointer declarations in RPC are also exactly as they are in C (i.e. `struct single_vldbentry *vldblist;`). Of course, one can't send pointers over the network, but one can use XDR pointers for sending recursive data types such as lists and trees (an example of a linked list will be demonstrated shortly).

Arrays

Fixed arrays are just like standard C array declarations (i.e. `struct UpdateEntry entries[20]`) without any side effect problems in rxgen. Since variable-length arrays have no explicit syntax in C, the angle-brackets are used for it and the array declarations are actually compiled into "struct"s. For example, declarations such as:

```
const   MAXBULKSIZE     = 10000;
const   MAXENTRIES      = 100;
opaque  bulk&lt;MAXBULKSIZE&gt;;          /* At most 10000 items */
int     hosts&lt;&gt;;                    /* any number of items */
typedef vldbentry blkentries&lt;100&gt;;   /* Preferable array decl */
```

are compiled into the following structs:

```
struct {
    u_int   bulk_len;       /* no of items */
    char    *bulk_val;      /* pointer to array */
} bulk;
```

for the `bulk` array, and similarly for the `blkentries<100>` array,

```
struct {
    u_int      blkentries_len;   /* no of items in array */
    vldbentry  *blkentries_val;  /* pointer to array */
} blkentries;
```

Therefore the user should be aware of the "magically" generated structure entries such as the number of items in the array (<array_name>_len) and the pointer to the array (<array_name>_val) since some of the entries will have to be filled in from the client/server programs. A sample proc would be:

```
typedef vldbentry blkentries&lt;MAXENTRIES&gt;;
proc GetBlk (OUT blkentries *vlentries) = VL_GETBLK;
```

or, more directly,

```
GetBlk(OUT vldbentry vlentries&lt;MAXENTRIES&gt;) = VL_GETBLK;
```

Note that although the latest method is preferable since one does not have to first use the typedef statement (and admittedly, programmers prefer avoiding typedefs), one should realize that rxgen does the structure expansion and the xdr creation implicitly; therefore the user should be aware of the `vldbentries_val` and `vldbentries_len` fields as before (see following examples).

Array example I (least desirable)

Procedure declaration in the interface configuration:

```
proc ListAttributes (IN vldblistbyattributes *attributes,
             INOUT blkentries *vldbentries) = VL_LISTATTRIBUTES;
```

Sample CLIENT code:

```
blkentries entries, *pnt;
entries.blkentries_len = 10;   /* max # returned entries */
entries.blkentries_val = (vldbentry *)malloc(LEN);
                               /* It must be set */


code = VL_ListAttributes(&attributes, &entries);
if (!code) {
    pnt = entries.blkentries_val;
    for (i=0; i &lt; entries.blkentries_len; i++, pnt++)
            display_vldbentry(pnt);
    /* Make sure you free the allocated space */
    free((char *)entries.blkentries_val);
}
```

Sample SERVER code:

```
VL_ListAttributes(attributes, entries)
{
    vldbentry *singleentry = entries-&gt;blkentries_val;
    entries-&gt;blkentries_len = 0;


    while (copy_to_vldbentry(&vlentry, singleentry))
        singleentry++, vldbentries-&gt;entries_len++;
}
```

Although this method for variable-size arrays works fine, there are some major drawbacks. The array parameter (i.e. vldbentries above) must be declared as INOUT since we need to pass the max length of the expected returned array; more importantly, a big (depending on the value of `_len`) chunk of junk code is going to be transferred to the server as result of the IN(out) side-effect of the array. It's an easy and convenient method if the returned array size can be predicted from the start and when the size is quite high. This method is included as an example of erroneous use (and abuse) of rxgen and should not be used.

Array example II (Desirable method)

Procedure declaration in the interface configuration (using Example I above):

```
proc ListAttributes (IN vldblistbyattributes *attributes,
    OUT blkentries *vldbentries) = VL_LISTATTRIBUTES;
```

Sample CLIENT code:

```
blkentries entries, *pnt;


code = VL_ListAttributes(&attributes, &entries);
if (!code) {
    pnt = entries.blkentries_val;
    for (i=0; i &lt; entries.blkentries_len; i++, pnt++)
            display_vldbentry(pnt);
    /* Make sure you free the allocated space (by rxgen) */
    free((char *)entries.blkentries_val);
}
```

Sample SERVER code:

```
VL_ListAttributes(attributes, entries)
{
    vldbentry *singleentry;
    entries-&gt;blkentries_len = 0;
    singleentry = entries-&gt;blkentries_val
        = (vldbentry *)malloc(MAXENTRIES * sizeof(vldbentry));


    while (copy_to_vldbentry(&vlentry, singleentry))
            singleentry++, vldbentries-&gt;entries_len++;
}
```

This is the best (and simplest) way of using variable-size arrays as an output parameter. It is the responsibility of the server-side stub to malloc() the adequate space which is automatically freed by the rxgen stub; the client side should free the space allocated by the rxgen-calling stub.

Array example III (Linked Lists)

Considering the following 3 declarations (could have applied some optimizations) in the configuration file:

```
typedef struct single_vldbentry *vldblist;
struct single_vldbentry {
    vldbentry vlentry;
    vldblist  next_vldb;
};
```

```
struct vldb_list {
    vldblist node;
};
```

and the rxgen procedure declaration:

```
LinkedList (IN vldblistbyattributes *attributes,
    OUT vldb_list *linkedentries) = VL_LINKEDLIST;
```

Sample CLIENT code:

```
vldb_list       linkedvldbs;
vldblist        vllist, vllist1;


bzero(&linkedvldbs, sizeof(vldb_list));
code = VL_LinkedList(&attributes, &nentries, &linkedvldbs);
if (!code) {
    printf("We got %d vldb entries\n", nentries);
    for (vllist = linkedvldbs.node; vllist; vllist = vllist1) {
        vllist1 = vllist-&gt;next_vldb;
        display_entry(&vllist-&gt;vlentry);
        free((char *)vllist);
    }
}
```

Sample SERVER code:

```
VL_LinkedList(rxcall, attributes, nentries, linkedvldbs);
{
    vldblist vllist, *vllistptr = &linkedvldbs-&gt;node;
    while (...) {
        vllist = *vllistptr
            = (single_vldbentry *)malloc (sizeof (single_vldbentry));
        copy_to_vldbentry(&tentry, &vllist-&gt;vlentry);
        nentries++;
        vllistptr = &vllist-&gt;next_vldb;
    };
    *vllistptr = NULL;
}
```

Using a linked list offers many advantages: Nothing is passed to the server (the parameter is OUT), no additional overhead is involved, and the caller doesn't have to explicitly prepare for an arbitrary return size. A drawback is that the caller has the responsibility of malloc() (on the server) and free (on the client) of each entry (to avoid unwanted core-leaks). Another drawback is that since it's a recursive call, the C stack will grow linearly with respect to the number of nodes in the list (so it's wise to increase the Rx LWP stack if huge amounts of data are expected back -- default stack size is 4K). The advantages should outweigh the disadvantages here.

It's important to pay attention to the comments of the three array examples above particularly when they're references to when the user should allocate / free space for the variable length arrays. The mechanism is very similar to the handling of strings thus you might need to review the strings section above; note that the linked lists are handled somewhat differently...

Miscellaneous examples

Below is an abbreviated version of a random interface file which shows some of the common cases.

```
/* Declaration of all structures used by the R.xg script interface */


struct AFSFid {
    unsigned long Volume;
    unsigned long Vnode;
    unsigned long Unique;
};


typedef long ViceDataType;


/* Note that TEST would be equivalent to "HEADER" only during the
   processing of the header, *.h, file */


#ifdef RPC_HDR
#define TEST "HEADER"
#else
#define TEST "REST"
#endif


/* This is the standard *.xg specification file */


package AFS_
splitprefix IN=BEFORE_ OUT=AFTER_;
Prefix Test


proc Remove(IN struct AFSFid *Did, IN string volname&lt;64&gt;,
    OUT struct AFSStatus *Status) = AFS_REMOVE;


DisconnectFS AUX_disconnectFS() = AFS_DISCONNECTFS;


proc GetVolumeInfo(IN string Vid,
    OUT struct VolumeInfo *Info) = AFS_GETVOLUMEINFO;
```

```
/* You could have more than an interface per configuration */


package VOTE_


/* Using the "multi" feature; thus VOTE_Beacon can be called as an
   multi-Rx call or as a regular call */


Beacon (IN long state, long voteStart,
    net_version *version, net_tid *tid)
    multi = VOTE_BEACON;


package DISK_


/* Using the "split" feature */


SendFile (IN long file, long offset,
    long length, net_version *version)
    split = DISK_SENDFILE;
```

Output of an actual interface configuration

We'll demonstrate some of the actual output generated by rxgen by following an abbreviated actual interface configuration.

Configuration file

Contents of the interface configuration file (*`vldbint.xg`*):

```
package VL_
#include "vl_opcodes.h"   /* The opcodes are included here */
%#include "vl_opcodes.h"  /* directly to other places */


/* Current limitations on parameters that affect other packages
   (i.e. volume) */


const   MAXNAMELEN      =       65;
const   MAXNSERVERS     =       8;
const   MAXTYPES        =       3;


/* External (visible) representation of an individual vldb entry */


struct vldbentry {
```

```
    char    name[MAXNAMELEN];
    long    volumeType;
    long    nServers;
    long    serverNumber[MAXNSERVERS];
    long    serverPartition[MAXNSERVERS];
    long    serverFlags[MAXNSERVERS];
    u_long  volumeId[MAXTYPES];
    long    flags;
};


typedef struct single_vldbentry  *vldblist;
struct single_vldbentry {
    vldbentry VldbEntry;
    vldblist next_vldb;
};


struct vldb_list {
    vldblist node;
};


/* vldb interface calls */


CreateEntry     (IN long Volid,
                 vldbentry *newentry) = VLCREATEENTRY;


GetEntryByName  (IN string volumename&lt;MAXNAMELEN&gt;,
                 OUT vldbentry *entry) = VLGETENTRYBYNAME;


GetNewVolumeId  (IN long bumpcount,
                 OUT long *newvolumid) = VLGETNEWVOLUMEID;


ReplaceEntry    (IN long Volid,
                 long voltype,
                 vldbentry *newentry,
                 long ReleaseType) multi = VLREPLACEENTRY;


ListAttributes  (IN VldbListByAttributes *attributes,
                 OUT long *nentries,
                 OUT vldbentry bulkentries&lt;MAXVLDBLEN&gt;)
                 = VLLISTATTRIBUTES;


LinkedList      (IN VldbListByAttributes *attributes,
                 OUT long *nentries,
                 OUT vldb_list *linkedentries) = VLLINKEDLIST;
```

We'll concentrate only on the Rx generated code since the R generated code (-R option) will soon be obsolete. For a detailed description on the Rx-related calls inside the generated stubs (i.e., rx_NewCall(), rx_EndCall()), along with details on what happens inside certain calls (like xdrrx_create()) please refer to the Rx documentation. Typing `rxgen vldbint.xg` will result in the creation of four files: *`vldbint.h`*, *`vldbint.xdr.c`*, *`vldbint.cs.c`* and *`vldbint.ss.c`*. A closer look at these files follows.

Header file (`vldbint.h`)

```
/* Machine generated file -- Do NOT edit */


#include "vl_opcodes.h"  /* directly to other places */
#define MAXNAMELEN 65
#define MAXNSERVERS 8
#define MAXTYPES 3


struct vldbentry {
    char name[MAXNAMELEN];
    long volumeType;
    long nServers;
    long serverNumber[MAXNSERVERS];
    long serverPartition[MAXNSERVERS];
    long serverFlags[MAXNSERVERS];
    u_long volumeId[MAXTYPES];
    long flags;
};
typedef struct vldbentry vldbentry;
bool_t xdr_vldbentry();


typedef struct single_vldbentry *vldblist;
bool_t xdr_vldblist();


struct single_vldbentry {
    vldbentry VldbEntry;
    vldblist next_vldb;
};
typedef struct single_vldbentry single_vldbentry;
bool_t xdr_single_vldbentry();


struct vldb_list {
    vldblist node;
};
typedef struct vldb_list vldb_list;
bool_t xdr_vldb_list();


#include &lt;rx/rx_multi.h&gt;
```

```
#define multi_VL_ReplaceEntry(Volid, voltype, newentry, ReleaseType) \
    multi_Body(StartVL_ReplaceEntry(multi_call, Volid, voltype,
               newentry, ReleaseType), EndVL_ReplaceEntry(multi_call))


typedef struct bulkentries {
    u_int bulkentries_len;
    vldbentry *bulkentries_val;
} bulkentries;
bool_t xdr_bulkentries();


/* Opcode-related useful stats for package: VL_ */
#define VL_LOWEST_OPCODE        501
#define VL_HIGHEST_OPCODE       506
#define VL_NUMBER_OPCODES       6
```

Notice that all structures are automatically typedef'ed and all consts are converted to #defines. Some data structures, such as bulkentries, are taken from procedure params (from ListAttributes proc). Thus, this should be kept in mind when creating stubs piecemeal with rxgen (i.e., using the -c, -h, -C, or -S flags). Also, one of the side effects of the multi option (in ReplaceEntry proc) is the generation of the multi_VL_ReplaceEntry above.

XDR routines for structures (vldbint.xdr.c)

```
/* Machine generated file -- Do NOT edit */


#include &lt;rx/xdr.h&gt;
#include "vldbint.h"


#include "vl_opcodes.h"  /* directly to other places */


bool_t
xdr_vldbentry(xdrs, objp)
    XDR *xdrs;
    vldbentry *objp;
{
    if (!xdr_vector(xdrs, (char *)objp-&gt;name, MAXNAMELEN,
                    sizeof(char), xdr_char))
        return (FALSE);
    if (!xdr_long(xdrs, &objp-&gt;volumeType))
        return (FALSE);
    if (!xdr_long(xdrs, &objp-&gt;nServers))
        return (FALSE);
    if (!xdr_vector(xdrs, (char *)objp-&gt;serverNumber, MAXNSERVERS,
                    sizeof(long), xdr_long))
        return (FALSE);
    if (!xdr_vector(xdrs, (char *)objp-&gt;serverPartition,
```

```
                    MAXNSERVERS, sizeof(long), xdr_long))
        return (FALSE);
    if (!xdr_vector(xdrs, (char *)objp-&gt;serverFlags, MAXNSERVERS,
                    sizeof(long), xdr_long))
        return (FALSE);
    if (!xdr_vector(xdrs, (char *)objp-&gt;volumeId, MAXTYPES,
                    sizeof(u_long), xdr_u_long))
        return (FALSE);
    if (!xdr_long(xdrs, &objp-&gt;flags))
        return (FALSE);
    return (TRUE);
}


bool_t
xdr_vldblist(xdrs, objp)
    XDR *xdrs;
    vldblist *objp;
{
    if (!xdr_pointer(xdrs, (char **)objp,
                     sizeof(struct single_vldbentry),
                     xdr_single_vldbentry))
        return (FALSE);
    return (TRUE);
}


bool_t
xdr_single_vldbentry(xdrs, objp)
    XDR *xdrs;
    single_vldbentry *objp;
{
    if (!xdr_vldbentry(xdrs, &objp-&gt;VldbEntry))
        return (FALSE);
    if (!xdr_vldblist(xdrs, &objp-&gt;next_vldb))
        return (FALSE);
    return (TRUE);
}


bool_t
xdr_vldb_list(xdrs, objp)
    XDR *xdrs;
    vldb_list *objp;
{
    if (!xdr_vldblist(xdrs, &objp-&gt;node))
        return (FALSE);
    return (TRUE);
}


bool_t
xdr_bulkentries(xdrs, objp)
    XDR *xdrs;
```

```
    bulkentries *objp;
{
    if (!xdr_array(xdrs, (char **)&objp-&gt;bulkentries_val,
                   (u_int *)&objp-&gt;bulkentries_len, MAXVLDBLEN,
                   sizeof(vldbentry), xdr_vldbentry))
        return (FALSE);
    return (TRUE);
}
```

Note that the xdr_bulkentries() is automatically generated as a side effect of a procedure parameter declaration. Thus, if identical multiple type parameter declarations are used, then multiply-defined xdr_* stubs will be created! We felt this was a better alternative to having the rxgen programmer deal with types such as bulkentries_1, bulkentries_2...

Client-Side stub routines (vldbint.cs.c)

```
/* Machine generated file -- Do NOT edit */


#include &lt;rx/xdr.h&gt;
#include &lt;rx/rx.h&gt;
#include &lt;afs/rxgen_consts.h&gt;
#include "vldbint.h"


#include "vl_opcodes.h"  /* directly to other places */


int VL_CreateEntry(z_conn, Volid, newentry)
    register struct rx_connection *z_conn;
    long Volid;
    vldbentry * newentry;
{
    struct rx_call *z_call = rx_NewCall(z_conn);
    static int z_op = 501;
    int z_result;
    XDR z_xdrs;


    xdrrx_create(&z_xdrs, z_call, XDR_ENCODE);


    /* Marshal the arguments */
    if ((!xdr_int(&z_xdrs, &z_op))
         || (!xdr_long(&z_xdrs, &Volid))
         || (!xdr_vldbentry(&z_xdrs, newentry))) {
            z_result = RXGEN_CC_MARSHAL;
            goto fail;
    }
```

```
    z_result = RXGEN_SUCCESS;
fail:
    return rx_EndCall(z_call, z_result);
}


int VL_GetEntryByName(z_conn, volumename, entry)
    register struct rx_connection *z_conn;
    char * volumename;
    vldbentry * entry;
{
    struct rx_call *z_call = rx_NewCall(z_conn);
    static int z_op = 504;
    int z_result;
    XDR z_xdrs;


    xdrrx_create(&z_xdrs, z_call, XDR_ENCODE);


    /* Marshal the arguments */
    if ((!xdr_int(&z_xdrs, &z_op))
         || (!xdr_string(&z_xdrs, &volumename, 65))) {
            z_result = RXGEN_CC_MARSHAL;
            goto fail;
    }


    /* Un-marshal the reply arguments */
    z_xdrs.x_op = XDR_DECODE;
    if ((!xdr_vldbentry(&z_xdrs, entry))) {
            z_result = RXGEN_CC_UNMARSHAL;
            goto fail;
    }


    z_result = RXGEN_SUCCESS;
fail:
    return rx_EndCall(z_call, z_result);
}


int VL_GetNewVolumeId(z_conn, bumpcount, newvolumid)
    register struct rx_connection *z_conn;
    long bumpcount;
    long * newvolumid;
{
    struct rx_call *z_call = rx_NewCall(z_conn);
    static int z_op = 505;
    int z_result;
    XDR z_xdrs;


    xdrrx_create(&z_xdrs, z_call, XDR_ENCODE);
```

```
    /* Marshal the arguments */
    if ((!xdr_int(&z_xdrs, &z_op))
         || (!xdr_long(&z_xdrs, &bumpcount))) {
            z_result = RXGEN_CC_MARSHAL;
            goto fail;
    }


    /* Un-marshal the reply arguments */
    z_xdrs.x_op = XDR_DECODE;
    if ((!xdr_long(&z_xdrs, newvolumid))) {
            z_result = RXGEN_CC_UNMARSHAL;
            goto fail;
    }


    z_result = RXGEN_SUCCESS;
fail:
    return rx_EndCall(z_call, z_result);
}


int VL_ReplaceEntry(z_conn, Volid, voltype, newentry, ReleaseType)
    register struct rx_connection *z_conn;
    long Volid, voltype, ReleaseType;
    vldbentry * newentry;
{
    struct rx_call *z_call = rx_NewCall(z_conn);
    static int z_op = 506;
    int z_result;
    XDR z_xdrs;


    xdrrx_create(&z_xdrs, z_call, XDR_ENCODE);


    /* Marshal the arguments */
    if ((!xdr_int(&z_xdrs, &z_op))
         || (!xdr_long(&z_xdrs, &Volid))
         || (!xdr_long(&z_xdrs, &voltype))
         || (!xdr_vldbentry(&z_xdrs, newentry))
         || (!xdr_long(&z_xdrs, &ReleaseType))) {
            z_result = RXGEN_CC_MARSHAL;
            goto fail;
    }


    z_result = RXGEN_SUCCESS;
fail:
    return rx_EndCall(z_call, z_result);
}
```

```
int StartVL_ReplaceEntry(z_call, Volid, voltype, newentry, ReleaseType)
    register struct rx_call *z_call;
    long Volid, voltype, ReleaseType;
    vldbentry * newentry;
{
    static int z_op = 506;
    int z_result;
    XDR z_xdrs;


    xdrrx_create(&z_xdrs, z_call, XDR_ENCODE);


    /* Marshal the arguments */
    if ((!xdr_int(&z_xdrs, &z_op))
         || (!xdr_long(&z_xdrs, &Volid))
         || (!xdr_long(&z_xdrs, &voltype))
         || (!xdr_vldbentry(&z_xdrs, newentry))
         || (!xdr_long(&z_xdrs, &ReleaseType))) {
            z_result = RXGEN_CC_MARSHAL;
            goto fail;
    }


    z_result = RXGEN_SUCCESS;
fail:
    return z_result;
}


int EndVL_ReplaceEntry(z_call)
    register struct rx_call *z_call;
{
    int z_result;
    XDR z_xdrs;


    z_result = RXGEN_SUCCESS;
fail:
    return z_result;
}


int VL_ListAttributes(z_conn, attributes, nentries, bulkentries_1)
    register struct rx_connection *z_conn;
    VldbListByAttributes * attributes;
    long * nentries;
    bulkentries * bulkentries_1;
{
    struct rx_call *z_call = rx_NewCall(z_conn);
    static int z_op = 511;
    int z_result;
    XDR z_xdrs;
```

```
    xdrrx_create(&z_xdrs, z_call, XDR_ENCODE);


    /* Marshal the arguments */
    if ((!xdr_int(&z_xdrs, &z_op))
         || (!xdr_VldbListByAttributes(&z_xdrs, attributes))) {
            z_result = RXGEN_CC_MARSHAL;
            goto fail;
    }


    /* Un-marshal the reply arguments */
    z_xdrs.x_op = XDR_DECODE;
    if ((!xdr_long(&z_xdrs, nentries))
         || (!xdr_bulkentries(&z_xdrs, bulkentries_1))) {
            z_result = RXGEN_CC_UNMARSHAL;
            goto fail;
    }


    z_result = RXGEN_SUCCESS;
fail:
    return rx_EndCall(z_call, z_result);
}


int VL_LinkedList(z_conn, attributes, nentries, linkedentries)
    register struct rx_connection *z_conn;
    VldbListByAttributes * attributes;
    long * nentries;
    vldb_list * linkedentries;
{
    struct rx_call *z_call = rx_NewCall(z_conn);
    static int z_op = 512;
    int z_result;
    XDR z_xdrs;


    xdrrx_create(&z_xdrs, z_call, XDR_ENCODE);


    /* Marshal the arguments */
    if ((!xdr_int(&z_xdrs, &z_op))
         || (!xdr_VldbListByAttributes(&z_xdrs, attributes))) {
            z_result = RXGEN_CC_MARSHAL;
            goto fail;
    }


    /* Un-marshal the reply arguments */
    z_xdrs.x_op = XDR_DECODE;
    if ((!xdr_long(&z_xdrs, nentries))
         || (!xdr_vldb_list(&z_xdrs, linkedentries))) {
            z_result = RXGEN_CC_UNMARSHAL;
```

```
            goto fail;
    }


    z_result = RXGEN_SUCCESS;
fail:
    return rx_EndCall(z_call, z_result);
}
```

Notice the side effect of the `multi` feature (three different modules for `ReplaceEntry` proc).

Server-Side stub routines (vldbint.ss.c)

```
/* Machine generated file -- Do NOT edit */


#include &lt;rx/xdr.h&gt;
#include &lt;rx/rx.h&gt;
#include &lt;afs/rxgen_consts.h&gt;
#include "vldbint.h"


#include "vl_opcodes.h"  /* directly to other places */


long _VL_CreateEntry(z_call, z_xdrs)
    struct rx_call *z_call;
    XDR *z_xdrs;
{
    long z_result;
    long Volid;
    vldbentry newentry;


    if ((!xdr_long(z_xdrs, &Volid))
         || (!xdr_vldbentry(z_xdrs, &newentry))) {
            z_result = RXGEN_SS_UNMARSHAL;
            goto fail;
    }


    z_result = VL_CreateEntry(z_call, Volid, &newentry);
fail:
    return z_result;
}


long _VL_GetEntryByName(z_call, z_xdrs)
    struct rx_call *z_call;
    XDR *z_xdrs;
{
    long z_result;
```

```
    char *volumename = (char *)0;
    vldbentry entry;


    if ((!xdr_string(z_xdrs, &volumename, 65))) {
            z_result = RXGEN_SS_UNMARSHAL;
            goto fail;
    }


    z_result = VL_GetEntryByName(z_call, &volumename, &entry);
    z_xdrs-&gt;x_op = XDR_ENCODE;
    if ((!xdr_vldbentry(z_xdrs, &entry)))
            z_result = RXGEN_SS_MARSHAL;
fail:
    z_xdrs-&gt;x_op = XDR_FREE;
    if (!xdr_string(z_xdrs, &volumename, 65)) goto fail1;
    return z_result;
fail1:
    return RXGEN_SS_XDRFREE;
}


long _VL_GetNewVolumeId(z_call, z_xdrs)
    struct rx_call *z_call;
    XDR *z_xdrs;
{
    long z_result;
    long bumpcount;
    long newvolumid;


    if ((!xdr_long(z_xdrs, &bumpcount))) {
            z_result = RXGEN_SS_UNMARSHAL;
            goto fail;
    }


    z_result = VL_GetNewVolumeId(z_call, bumpcount, &newvolumid);
    z_xdrs-&gt;x_op = XDR_ENCODE;
    if ((!xdr_long(z_xdrs, &newvolumid)))
            z_result = RXGEN_SS_MARSHAL;
fail:
    return z_result;
}


long _VL_ReplaceEntry(z_call, z_xdrs)
    struct rx_call *z_call;
    XDR *z_xdrs;
{
    long z_result;
    long Volid, voltype, ReleaseType;
    vldbentry newentry;
```

```
    if ((!xdr_long(z_xdrs, &Volid))
         || (!xdr_long(z_xdrs, &voltype))
         || (!xdr_vldbentry(z_xdrs, &newentry))
         || (!xdr_long(z_xdrs, &ReleaseType))) {
            z_result = RXGEN_SS_UNMARSHAL;
            goto fail;
    }


    z_result = VL_ReplaceEntry(z_call, Volid, voltype, &newentry,
                               ReleaseType);
fail:
    return z_result;
}


long _VL_ListAttributes(z_call, z_xdrs)
    struct rx_call *z_call;
    XDR *z_xdrs;
{
    long z_result;
    VldbListByAttributes attributes;
    long nentries;
    bulkentries bulkentries_1;


    if ((!xdr_VldbListByAttributes(z_xdrs, &attributes))) {
            z_result = RXGEN_SS_UNMARSHAL;
            goto fail;
    }


    z_result = VL_ListAttributes(z_call, &attributes, &nentries,
                                 &bulkentries_1);
    z_xdrs-&gt;x_op = XDR_ENCODE;
    if ((!xdr_long(z_xdrs, &nentries))
         || (!xdr_bulkentries(z_xdrs, &bulkentries_1)))
            z_result = RXGEN_SS_MARSHAL;
fail:
    z_xdrs-&gt;x_op = XDR_FREE;
    if (!xdr_bulkentries(z_xdrs, &bulkentries_1)) goto fail1;
    return z_result;
fail1:
    return RXGEN_SS_XDRFREE;
}


long _VL_LinkedList(z_call, z_xdrs)
    struct rx_call *z_call;
    XDR *z_xdrs;
{
    long z_result;
```

```
    VldbListByAttributes attributes;
    long nentries;
    vldb_list linkedentries;


    if ((!xdr_VldbListByAttributes(z_xdrs, &attributes))) {
            z_result = RXGEN_SS_UNMARSHAL;
            goto fail;
    }


    z_result = VL_LinkedList(z_call, &attributes, &nentries,
                             &linkedentries);
    z_xdrs-&gt;x_op = XDR_ENCODE;
    if ((!xdr_long(z_xdrs, &nentries))
         || (!xdr_vldb_list(z_xdrs, &linkedentries)))
            z_result = RXGEN_SS_MARSHAL;
fail:
    return z_result;
}


long _VL_CreateEntry();
long _VL_GetEntryByName();
long _VL_GetNewVolumeId();
long _VL_ReplaceEntry();
long _VL_ListAttributes();
long _VL_LinkedList();


static long (*StubProcsArray0[])() = {_VL_CreateEntry,
    _VL_GetEntryByName, _VL_GetNewVolumeId, _VL_ReplaceEntry,
    _VL_ListAttributes, _VL_LinkedList};


VL_ExecuteRequest(z_call)
    register struct rx_call *z_call;
{
    int op;
    XDR z_xdrs;
    long z_result;


    xdrrx_create(&z_xdrs, z_call, XDR_DECODE);
    if (!xdr_int(&z_xdrs, &op))
        z_result = RXGEN_DECODE;
    else if (op &lt; VL_LOWEST_OPCODE || op &gt; VL_HIGHEST_OPCODE)
        z_result = RXGEN_OPCODE;
    else
        z_result = (*StubProcsArray0[op - VL_LOWEST_OPCODE])
            (z_call, &z_xdrs);
    return z_result;
}
```

If there were gaps in the procedures' opcode sequence the code for VL_ExecuteRequest() routine would be have been drastically different (it would have been a case statement for each procedure).

Notes

rxgen is implemented from Sun's rpcgen utility. All of the standard rpcgen's functionality is fully maintained. Note that some active rpcgen options that don't apply to rxgen's purpose aren't referenced here (i.e., -s, -l, -m options) and the interested reader should refer to rpcgen(1) for details.

When the `%#include <include file>` feature is used make sure that you don't have any rxgen language features (i.e. %#defines) since you'll get syntax errors during compilations..

Since this is an ongoing project many of the above may change/disappear without a major warning.

See Also

Rxgen Syntax Summary: Summary description of rxgen's grammar.

Rpcgen Programming Guide: Sun's RPC protocol compiler. rxgen was implemented as an extension to that compiler.

External Data Representation: Sun Technical Notes: Detailed examples in using XDR.

RPCL Syntax Summary: Summary of Sun's Remote Procedure Call Language.

Rx: An extended Remote Procedure Call Protocol.

rgen: An earlier version of a similar stub generator used for the R RPC protocol.

Copyright

scout

Name

scout — Monitors the File Server process

Synopsis

scout [initcmd] -server <*servers 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]

scout [i] -s <*servers to monitor*>+ [-b <*base server name*>] [-f <*poll frequency, in seconds*>] [-ho] [-a <*specify attention (highlighting) level*>+] [-d <*turn debugging output on to the named file*>] [-he]

Description

The scout command displays statistics gathered from the File Server process running on each machine specified with the -server argument. "OUTPUT" on page explains the meaning of the statistics and describes how they appear in the command shell, which is preferably a window managed by a window manager program.

Cautions

The scout program must be able to access the curses graphics package, which it uses to display statistics. Most UNIX distributions include curses as a standard utility.

Both dumb terminals and windowing systems that emulate terminals can display the scout program's statistics. The display makes use of reverse video and cursor addressing, so the display environment must support those features for it to look its best (most windowing systems do, most dumb terminals do not). Also, set the TERM environment variable to the correct terminal type, or one with characteristics similar to the actual ones. For machines running the AIX operating system, 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`.

Options

initcmd

Accommodates the command's use of the AFS command parser, and is optional.

-server <*servers to monitor*>+

Specifies each file server machine running a File Server process to monitor. Provide each machine's fully qualified hostname unless the -basename argument is used. In that case, specify only the unique initial part of each machine name, omitting the domain name suffix (the basename) common to all the names. It is also acceptable to use the shortest

abbreviated form of a host name that distinguishes it from other machines, but successful resolution depends on the availability of a name resolution service (such as the Domain Name Service or a local host table) at the time the command is issued.

-basename <*base server name*>

Specifies the basename (domain name) suffix common to all of the file server machine names specified with the -server argument, 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 ABC Corporation cell, the proper value is `abc.com` rather than `.abc.com`.

-frequency <*poll frequency*>

Indicates how often to probe the File Server processes. Specify a number of seconds greater than `0` (zero). The default is 60 seconds.

-host

Displays the name of the machine that is running the scout program, in the banner line of the display screen.

-attention <*attention level*>+

Defines a list of entries, each of which pairs a statistic and a threshold value. When the value of the statistic exceeds the indicated threshold value, it is highlighted (in reverse video) in the display. List the pairs in any order. The acceptable values are the following:

conn <*connections*>

Indicates the number of open connections to client processes at which to highlight the statistic. The statistic returns to regular display when the value goes back below the threshold. There is no default threshold.

An example of an acceptable value is conn 300.

disk <*blocks_free*>

Indicates the number of remaining free kilobyte blocks at which to highlight the statistic. The statistic returns to regular display when the value again exceeds the threshold. There is no default threshold.

An example of an acceptable value is disk 5000.

disk <*percent_full*>%

Indicates the percentage of disk usage at which to highlight the statistic. The statistic returns to regular display when the value goes back below the threshold. The default threshold is 95%. Acceptable values are the integers in the range from `0` to `99`, followed by the percent sign (`%`) to distinguish this type of value from the one described just previously.

An example is disk 90%.

fetch <*fetch RPCs*>

Indicates the cumulative number of fetch RPCs from client processes at which to highlight the statistic. The statistic does not return to regular display until the File Server process restarts, at which time the value returns to zero. There is no default threshold.

Example of a legal value: fetch 6000000

store <*store RPCs*>

Indicates the cumulative number of store RPCs from client processes at which to highlight the statistic. The statistic does not return to regular display until the File Server process restarts, at which time the value returns to zero. There is no default threshold.

Example of an acceptable value: store 200000

ws <*active client machines*>

Indicates the number of client machines with active open connections at which to highlight the statistic. An active connection is defined as one over which the File Server and client have communicated in the last 15 minutes. The statistic returns to regular display when the value goes back below the threshold. There is no default threshold.

Example of an acceptable value: ws 65

-debug <*debugging trace file*>

Specifies the pathname of the file into which to write a debugging trace. Partial pathnames are interpreted relative to the current working directory.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The scout program 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 needs 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 string `Scout` appears in the banner line at the top of the window or screen. Two optional arguments place additional information in the banner line:

- The -host flag displays the name of the machine where the scout program is running. As mentioned previously, this is useful when running the scout program on several machines but displaying the results on a single machine.

  For example, when the -host flag is included and the scout program is running on the machine `client1.abc.com`, the banner line reads as follows:

  ```
  [client1.abc.com] Scout
  ```

- The -basename argument displays the indicated basename on the banner line. For example, including the argument `-basename abc.com` argument results in the following banner line:

  ```
  Scout for abc.com
  ```

The Statistics Display Region

In this region, which occupies the majority of the window, the scout process 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 equals or exceeds the number in the `Ws` column (see the fourth entry below), 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 (`Ws` stands for workstations) that have communicated with the File Server process within the last 15 minutes. Such machines are termed *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.

server name

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 following form:

```
x:&lt;free_blocks&gt;
```

where `x` indicates the partition name. For example, `a:8949` specifies that the */vicepa* partition has 8,949 1-KB blocks free. Available space can be displayed for up to 26 partitions. If the window is not wide enough for all partition entries to appear on a single line, the scout process 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: &gt; 95% used
```

because by default the scout program highlights the entry for any partition that is over 95% full.

For all columns except the fifth (file server machine name), the optional -attention argument sets the value at which entries in the column are highlighted to indicate that a certain value has been exceeded. Only values in the fifth and `Disk attn` columns ever become highlighted by default.

If the scout program is unable to access or otherwise obtain information about a partition, it generates a message similar to the following example:

```
Could not get information on server fs1.abc.com partition /vicepa
```

The Message/Probe Line

The bottom line of the scout screen indicates how many times the scout program has probed the File Server processes for statistics. The statistics gathered in the latest probe appear in the statistics display region. The -frequency argument overrides the default probe frequency of 60 seconds.

Examples

See the chapter on monitoring tools in the *IBM AFS Administration Guide*, which illustrates the displays that result from different combinations of options.

Privilege Required

None

See Also

"afsmonitor(1)" on page 25, "fstrace(8)" on page 783

Copyright

sys

Name

sys — Reports the compile-time CPU / operating system type

Synopsis

sys

Description

The sys command displays the string set at compile time that indicates the local machine's CPU / operating system (OS) type, conventionally called the *sysname*. This string is the default for the value stored in kernel memory. The Cache Manager substitutes this string for the *@sys* variable which can occur in AFS pathnames; the *IBM AFS Quick Beginnings* and *IBM AFS Administration Guide* explain how using *@sys* can simplify cell configuration.

To set a new value in kernel memory, use the fs sysname command. To view the current value set in the kernel, use either fs sysname or livesys.

Cautions

You almost always want to use livesys rather than this command. The sys command displays a single value hard-coded at compile time. It does not query the Cache Manager for the current value and it does not report sysname lists. If you have changed the local system type with fs sysname, or if you run a version of sys compiled differently than the Cache Manager running on the system, the value retured will not match the behavior of the Cache Manager. The only reason to use sys is that livesys wasn't available in older versions of AFS.

Output

The machine's system type appears as a text string:

```
I&lt;system_type&gt;
```

Examples

The following example shows the output produced on a Sun SPARCStation running Solaris 5.7:

```
% sys
sun4x_57
```

Privilege Required

None

See Also

"fs_sysname(1)" on page 145, "livesys(1)" on page 167

IBM AFS Quick Beginnings

IBM AFS Administration Guide

Copyright

tokens

Name

`tokens` — Displays the issuer's tokens

Synopsis

tokens [-help]

tokens [-h]

Description

The tokens command displays all tokens (tickets) cached on the local machine 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.

Options

-help

Prints the online help for this command. All other valid options are ignored.

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.

The output of the Kerberos version of this command, tokens.krb, also reports the following about the Kerberos ticket-granting ticket: the ticket owner, which Kerberos ticket-granting service that issued the ticket (for example, `krbtgt.ABC.COM`), and ticket's expiration date.

The string `--End of list--` appears at the end of the output. If the user is not authenticated in any cell, this line is all that appears.

Examples

The following example shows the output when the issuer is not authenticated in any cell.

```
% tokens
Tokens held by the Cache Manager:
```

```
   --End of list--
```

The following example shows the output when the issuer is authenticated in ABC Corporation cell, where he or she has AFS UID 1000.

```
% tokens
Tokens held by the Cache Manager:


User’s (AFS ID 1000) tokens for afs@abc.com [Expires Jan 2 10:00]
   --End of list--
```

The following example shows the output when the issuer is authenticated in the ABC Corporation cell, the State University cell, and the XYZ Company cell. The user has different AFS UIDs in the three cells. Tokens for last cell are expired:

```
% tokens
Tokens held by the Cache Manager:


User’s (AFS ID 1000) tokens for afs@abc.com [Expires Jan 3 10:00]
User’s (AFS ID 4286) tokens for afs@stateu.edu [Expires Jan 3 1:34]
User’s (AFS ID 22) tokens for afs@xyz.com [&gt;&gt;Expired&lt;]
   --End of list--
```

The following example shows the output when the issuer uses the tokens.krb version of the command after authenticating in the ABC Corporation cell using the klog.krb command.

```
% tokens.krb
Tokens held by the Cache Manager:


User’s (AFS ID 1000) tokens for afs@abc.com [Expires Jan 31 00:09]
User smiths tokens for krbtgt.ABC.COM@abc.com [Expires Jan 31 00:09]
   --End of list--
```

Privilege Required

None

See Also

"klog(1)" on page 153, "unlog(1)" on page 283

Copyright

translate_et

Name

`translate_et` — Translates numbered error codes into text messages

Synopsis

translate_et <*error number*>+

Description

The translate_et command translates each specified error number into a text message.

This command does not use the syntax conventions of the AFS command suites. Provide the command name in full.

Options

<*error number*>+

Specifies each error number to translate.

Examples

The following command translates the error numbers 1 and 4:

```
% translate_et 1 4
1 ().1 = Not owner
4 ().4 = Interrupted system call
```

Privilege Required

None

Copyright

udebug

Name

udebug — Reports Ubik process status for a database server process

Synopsis

udebug -servers <*server machine*> [-port <*IP port*>] [-long] [-help]

udebug -s <*server machine*> [-p <*IP port*>] [-l] [-h]

Description

The udebug command displays the status of the lightweight Ubik process for the database server process identified by the -port argument that is running on the database server machine named by the -servers argument. The output identifies the machines where peer database server processes are running, which of them is the synchronization site (Ubik coordinator), and the status of the connections between them.

Options

-servers <*server machine*>

Names the database server machine that is running the process for which to display status information. Provide the machine's IP address in dotted decimal format, its fully qualified host name (for example, fs1.abc.com), or the shortest abbreviated form of its host name that distinguishes it from other machines. Successful use of an abbreviated form depends on the availability of a name resolution service (such as the Domain Name Service or a local host table) at the time the command is issued.

-port <*IP port*>

Identifies the database server process for which to display status information, either by its process name or port number. Provide one of the following values.

buserver or 7021 for the Backup Server

kaserver or 7004 for the Authentication Server

ptserver or 7002 for the Protection Server

vlserver or 7003 for the Volume Location Server

-long

Reports additional information about each peer of the machine named by the -servers argument. The information appears by default if that machine is the synchronization site.

-help

Prints the online help for this command. All other valid options are ignored.

Output

Several of the messages in the output provide basic status information about the Ubik process on the machine specified by the -servers argument, and the remaining messages are useful mostly for debugging purposes.

To check basic Ubik status, issue the command for each database server machine in turn. In the output for each, one of the following messages appears in the top third of the output.

```
I am sync site . . . (&lt;#_sites&gt; servers)
```

```
I am not sync site
```

For the synchronization site, the following message indicates that all sites have the same version of the database, which implies that Ubik is functioning correctly. See the following for a description of values other than `1f`.

```
Recovery state 1f
```

For correct Ubik operation, the database server machine clocks must agree on the time. The following messages, which are the second and third lines in the output, report the current date and time according to the database server machine's clock and the clock on the machine where the udebug command is issued.

```
Host's &lt;IP_addr&gt; time is &lt;dbserver_date/time&gt;
Local time is &lt;local_date/time&gt; (time differential &lt;skew&gt; secs)
```

The <skew> is the difference between the database server machine clock and the local clock. Its absolute value is not vital for Ubik functioning, but a difference of more than a few seconds between the *skew* values for the database server machines indicates that their clocks are not synchronized and Ubik performance is possibly hampered.

Following is a description of all messages in the output. As noted, it is useful mostly for debugging and most meaningful to someone who understands Ubik's implementation.

The output begins with the following messages. The first message reports the IP addresses that are configured with the operating system on the machine specified by the -servers argument. As previously noted, the second and third messages report the current date and time according to the clocks on the database server machine and the machine where the udebug command is issued, respectively. All subsequent timestamps in the output are expressed in terms of the local clock rather than the database server machine clock.

```
Host’s addresses are: &lt;list_of_IP_addrs&gt;
Host’s &lt;IP_addr&gt; time is &lt;dbserver_date/time&gt;
Local time is &lt;local_date/time&gt; (time differential &lt;skew&gt; secs)
```

If the <skew> is more than about 10 seconds, the following message appears. As noted, it does not necessarily indicate Ubik malfunction: it denotes clock skew between the database server machine and the local machine, rather than among the database server machines.

```
****clock may be bad
```

If the udebug command is issued during the coordinator election process and voting has not yet begun, the following message appears next.

```
Last yes vote not cast yet
```

Otherwise, the output continues with the following messages.

```
Last yes vote for &lt;sync_IP_addr&gt; was &lt;last_vote&gt; secs ago (sync site);
Last vote started &lt;vote_start&gt; secs ago (at &lt;date/time&gt;)
Local db version is &lt;db_version&gt;
```

The first indicates which peer this Ubik process last voted for as coordinator (it can vote for itself) and how long ago it sent the vote. The second message indicates how long ago the Ubik coordinator requested confirming votes from the secondary sites. Usually, the <last_vote> and <vote_start> values are the same; a difference between them can indicate clock skew or a slow network connection between the two database server machines. A small difference is not harmful. The third message reports the current version number <db_version> of the database maintained by this Ubik process. It has two fields separated by a period. The field before the period is based on a timestamp that reflects when the database first changed after the most recent coordinator election, and the field after the period indicates the number of changes since the election.

The output continues with messages that differ depending on whether the Ubik process is the coordinator or not.

- If there is only one database server machine, it is always the coordinator (synchronization site), as indicated by the following message.

  ```
  I am sync site forever (1 server)
  ```

- If there are multiple database sites, and the -servers argument names the coordinator (synchronization site), the output continues with the following two messages.

```
I am sync site until &lt;expiration&gt; secs from now (at &lt;date/time&gt;)
    (&lt;#_sites&gt; servers)
Recovery state &lt;flags&gt;
```

The first message (which is reported on one line) reports how much longer the site remains coordinator even if the next attempt to maintain quorum fails, and how many sites are participating in the quorum. The *flags* field in the second message is a hexadecimal number that indicates the current state of the quorum. A value of `1f` indicates complete database synchronization, whereas a value of `f` means that the coordinator has the correct database but cannot contact all secondary sites to determine if they also have it. Lesser values are acceptable if the udebug command is issued during coordinator election, but they denote a problem if they persist. The individual flags have the following meanings:

0x1

This machine is the coordinator.

0x2

The coordinator has determined which site has the database with the highest version number.

0x4

The coordinator has a copy of the database with the highest version number.

0x8

The database's version number has been updated correctly.

0x10

All sites have the database with the highest version number.

If the udebug command is issued while the coordinator is writing a change into the database, the following additional message appears.

```
I am currently managing write transaction I&lt;identifier&gt;
```

- If the -servers argument names a secondary site, the output continues with the following messages.

```
I am not sync site
Lowest host &lt;lowest_IP_addr&gt; was set &lt;low_time&gt; secs ago
Sync host &lt;sync_IP_addr&gt; was set &lt;sync_time&gt; secs ago
```

The <lowest_IP_addr> is the lowest IP address of any peer from which the Ubik process has received a message recently, whereas the <sync_IP_addr> is the IP address of the current coordinator. If they differ, the machine with the lowest IP address is not currently the coordinator. The Ubik process continues voting for the current coordinator as long as they remain in contact, which provides for maximum stability. However, in the event of another

coordinator election, this Ubik process votes for the <lowest_IP_addr> site instead (assuming they are in contact), because it has a bias to vote in elections for the site with the lowest IP address.

For both the synchronization and secondary sites, the output continues with the following messages. The first message reports the version number of the database at the synchronization site, which needs to match the <db_version> reported by the preceding `Local db version` message. The second message indicates how many VLDB records are currently locked for any operation or for writing in particular. The values are nonzero if the udebug command is issued while an operation is in progress.

```
Sync site’s db version is &lt;db_version&gt;
&lt;locked&gt; locked pages, &lt;writes&gt; of them for write
```

The following messages appear next only if there are any read or write locks on database records:

```
There are read locks held
There are write locks held
```

Similarly, one or more of the following messages appear next only if there are any read or write transactions in progress when the udebug command is issued:

```
There is an active write transaction
There is at least one active read transaction
Transaction tid is &lt;tid&gt;
```

If the machine named by the -servers argument is the coordinator, the next message reports when the current coordinator last updated the database.

```
 Last time a new db version was labelled was:
        &lt;last_restart&gt; secs ago (at &lt;date/time&gt;)
```

If the machine named by the -servers argument is the coordinator, the output concludes with an entry for each secondary site that is participating in the quorum, in the following format.

```
Server (&lt;IP_address&gt;): (db &lt;db_version&gt;)
last vote rcvd &lt;last_vote&gt; secs ago (at &lt;date/time&gt;),
last beacon sent &lt;last_beacon&gt; secs ago (at &lt;date/time&gt;),
    last vote was { yes | no }
dbcurrent={ 0 | 1 }, up={ 0 | 1 } beaconSince={ 0 | 1 }
```

The first line reports the site’s IP address and the version number of the database it is maintaining. The <last_vote> field reports how long ago the coordinator received a vote message from the Ubik process at the site, and the <last_beacon> field how long ago the coordinator last requested a vote message. If the udebug command is issued during the coordinator election process and voting has not yet begun, the following messages appear instead.

```
Last vote never rcvd
Last beacon never sent
```

On the final line of each entry, the fields have the following meaning:

- `dbcurrent` is `1` if the site has the database with the highest version number, `0` if it does not.
- `up` is `1` if the Ubik process at the site is functioning correctly, `0` if it is not.
- `beaconSince` is `1` if the site has responded to the coordinator's last request for votes, `0` if it has not.

Including the -long flag produces peer entries even when the -servers argument names a secondary site, but in that case only the *IP_address* field is guaranteed to be accurate. For example, the value in the <db_version> field is usually `0.0`, because secondary sites do not poll their peers for this information. The values in the *last_vote* and *last_beacon* fields indicate when this site last received or requested a vote as coordinator; they generally indicate the time of the last coordinator election.

Examples

This example checks the status of the Ubik process for the Volume Location Server on the machine `afs1`, which is the synchronization site.

```
% udebug afs1 vlserver
Host's addresses are: 192.12.107.33
Host's 192.12.107.33 time is Wed Oct 27 09:49:50 1999
Local time is Wed Oct 27 09:49:52 1999 (time differential 2 secs)
Last yes vote for 192.12.107.33 was 1 secs ago (sync site);
Last vote started 1 secs ago (at Wed Oct 27 09:49:51 1999)
Local db version is 940902602.674
I am sync site until 58 secs from now (at Wed Oct 27 09:50:50 1999) (3 servers)
Recovery state 1f
Sync site's db version is 940902602.674
0 locked pages, 0 of them for write
Last time a new db version was labelled was:
         129588 secs ago (at Mon Oct 25 21:50:04 1999)


Server( 192.12.107.35 ): (db 940902602.674)
    last vote rcvd 2 secs ago (at Wed Oct 27 09:49:50 1999),
    last beacon sent 1 secs ago (at Wed Oct 27 09:49:51 1999), last vote was yes
    dbcurrent=1, up=1 beaconSince=1


Server( 192.12.107.34 ): (db 940902602.674)
    last vote rcvd 2 secs ago (at Wed Oct 27 09:49:50 1999),
    last beacon sent 1 secs ago (at Wed Oct 27 09:49:51 1999), last vote was yes
    dbcurrent=1, up=1 beaconSince=1
```

This example checks the status of the Authentication Server on the machine with IP address 192.12.107.34, which is a secondary site. The local clock is about 4 minutes behind the database server machine's clock.

```
% udebug 192.12.107.34 7004
Host's addresses are: 192.12.107.34
Host's 192.12.107.34 time is Wed Oct 27 09:54:15 1999
Local time is Wed Oct 27 09:50:08 1999 (time differential -247 secs)
****clock may be bad
Last yes vote for 192.12.107.33 was 6 secs ago (sync site);
Last vote started 6 secs ago (at Wed Oct 27 09:50:02 1999)
Local db version is 940906574.25
I am not sync site
Lowest host 192.12.107.33 was set 6 secs ago
Sync host 192.12.107.33 was set 6 secs ago
Sync site's db version is 940906574.25
0 locked pages, 0 of them for write
```

Privilege Required

None

See Also

"buserver(8)" on page 763, "kaserver(8)" on page 859, "ptserver(8)" on page 875, "vlserver(8)" on page 915

Copyright

unlog

Name

`unlog` — Discards all of the issuer's tokens

Synopsis

unlog [-cell <*cell name*>+] [-help]

unlog [-c <*cell name*>+] [-h]

Description

The unlog command by default discards all tokens that the issuer currently holds. To discard tokens for certain cells only, name them with the -cell argument.

Since a token pertains to one client machine only, destroying tokens on one machine has no effect on tokens on another machine.

Cautions

Specifying one or more cell names can cause a brief authentication outage during which the issuer has no valid tokens in any cell. This is because the command actually discards all tokens and then restores the ones for cells not named by the -cell argument. The outage can sometimes interrupt the operation of jobs that require authentication.

Options

-cell <*cell name*>+

Specifies each cell for to discard the token. If this argument is omitted, the Cache Manager discards all tokens. Provide the fully qualified domain name, or a shortened form, in which case successful resolution depends on the availability of a name resolution service (such as the Domain Name Service or a local host table) at the time the command is issued.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command discards all tokens.

```
% unlog
```

The following command discards only the tokens for the `abc.com` and `stateu.edu` cells.

```
% unlog -cell abc.com stateu
```

Privilege Required

None

See Also

"klog(1)" on page 153, "tokens(1)" on page 269

Copyright

up

Name

up — Recursively copy directories, preserving AFS metadata

Synopsis

up [-v] [-1] [-f] [-r] [-x] <*source directory*> <*destination directory*>

Description

The up command recursively copies the files and subdirectories in a specified source directory to a specified destination directory. The command interpreter changes the destination directory and the files and subdirectories in it in the following ways:

- It copies the source directory's access control list (ACL) to the destination directory and its subdirectories, overwriting any existing ACLs.
- If the issuer is logged on as the local superuser root and has AFS tokens as a member of the group system:administrators, then the source directory's owner (as reported by the `ls -ld` command) becomes the owner of the destination directory and all files and subdirectories in it. Otherwise, the issuer's user name is recorded as the owner.
- If a file or directory exists in both the source and destination directories, the source version overwrites the destination version. The overwrite operation fails if the first (user) `w` (write) mode bit is turned off on the version in the destination directory, unless the -f flag is provided.
- The modification timestamp on a file (as displayed by the `ls -l` command) in the source directory overwrites the timestamp on a file of the same name in the destination directory, but the timestamp on an existing subdirectory in the destination directory remains unchanged. If the command creates a new subdirectory in the destination directory, the new subdirectory's timestamp is set to the time of the copy operation, rather than to the timestamp that the subdirectory has in the source directory.

The up command is idempotent, meaning that if its execution is interrupted by a network, server machine, or process outage, then a subsequent reissue of the same command continues from the interruption point, rather than starting over at the beginning. This saves time and reduces network traffic in comparison to the UNIX commands that provide similar functionality.

The up command returns a status code of `0` (zero) only if it succeeds. Otherwise, it returns a status code of `1` (one).

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Options

-v

Prints a detailed trace to the standard output stream as the command runs.

-1

Copies only the files in the top level source directory to the destination directory, rather than copying recursively through subdirectories. The source directory's ACL still overwrites the destination directory's. (This is the number one, not the letter `l`.)

-f

Overwrites existing directories, subdirectories, and files even if the first (user) `w` (write) mode bit is turned off on the version in the destination directory.

-r

Creates a backup copy of all files overwritten in the destination directory and its subdirectories, by adding a `.old` extension to each filename.

-x

Sets the modification timestamp on each file to the time of the copying operation.

source directory

Names the directory to copy recursively.

destination directory

Names the directory to which to copy. It does not have to exist already.

Examples

The following command copies the contents of the directory *`dir1`* to directory *`dir2`*:

```
% up dir1 dir2
```

Privilege Required

The issuer must have the `a` (administer) permission on the ACL of both the source and destination directories.

Copyright

up

vos

Name

vos — Introduction to the vos command suite

Description

The commands in the vos command suite are the administrative interface to the Volume Server and Volume Location (VL) Server. System administrators use vos commands to create, move, delete, replicate, back up and examine volumes, among other operations. The VL Server automatically records in the Volume Location Database (VLDB) changes in volume status and location that result from vos commands.

The operations invoked by most vos commands are idempotent, meaning that if an operation is interrupted by a network, server machine, or process outage, then a subsequent attempt at the same operation continues from the interruption point, rather than starting over at the beginning of the operation. Before executing a command, the Volume and VL Servers check the current state of the volumes and VLDB records to be altered by the command. If they are already in the desired end state (or a consistent intermediate state), there is no need to repeat the internal steps that brought them there. Idempotency does not apply if the command issuer explicitly interrupts the operation with the Ctrl-C command or another interrupt signal. In that case, the volume is left locked and the administrator must use the vos unlock command to unlock it before proceeding.

It is important that the VLDB accurately indicate the status of the volumes on file server machines at all times. "vldb.DB0(5)" on page 519 and "afs_volume_header(5)" on page 445 describe the information recorded in the VLDB and volume headers, respectively. If a vos command changes volume status, it automatically records the change in the corresponding VLDB entry. The most common cause of discrepancies between the VLDB and volume status on file server machines is interrupted operations; to restore consistency, use the vos syncserv and vos syncvldb commands.

There are several categories of commands in the vos command suite:

- Commands to create, move, and rename volumes: vos backup, vos backupsys, vos create, vos move, and vos rename.
- Commands to remove VLDB volume records or volumes or both: vos delentry, vos remove, and vos zap.
- Commands to edit or display VLDB server entries: vos changeaddr and vos listaddrs.
- Commands to create and restore dump files: vos dump and vos restore.
- Commands to administer replicated volumes: vos addsite, vos release, and vos remsite.
- Commands to display VLDB records, volume headers, or both: vos examine, vos listvldb, and vos listvol.
- Commands to display information about partitions that house volumes: vos listpart and vos partinfo.

- Commands to restore consistency between the VLDB and volume headers: vos syncserv and vos syncvldb.
- Commands to lock and unlock VLDB entries: vos lock, vos unlock, and vos unlockvldb.
- A command to report Volume Server status: vos status.
- Commands to obtain help: vos apropos and vos help.

Options

The following arguments and flags are available on many commands in the bos suite. The reference page for each command also lists them, but they are described here in greater detail.

-cell <*cell name*>

Names the cell in which to run the command. It is acceptable to abbreviate the cell name to the shortest form that distinguishes it from the other entries in the */usr/vice/etc/CellServDB* file on the local machine. If the -cell argument is omitted, the command interpreter determines the name of the local cell by reading the following in order:

- The value of the AFSCELL environment variable.
- The local */usr/vice/etc/ThisCell* file.

Do not combine the -cell and -localauth options. A command on which the -localauth flag is included always runs in the local cell (as defined in the server machine's local */usr/afs/etc/ThisCell* file), whereas a command on which the -cell argument is included runs in the specified foreign cell.

-help

Prints a command's online help message on the standard output stream. Do not combine this flag with any of the command's other options; when it is provided, the command interpreter ignores all other options, and only prints the help message.

-localauth

Constructs a server ticket using the server encryption key with the highest key version number in the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents the ticket, which never expires, to the Volume Server and VL Server during mutual authentication.

Use this flag only when issuing a command on a server machine; client machines do not usually have a */usr/afs/etc/KeyFile* file. The issuer of a command that includes this flag must be logged on to the server machine as the local superuser `root`. The flag is useful for commands invoked by an unattended application program, such as a process controlled by the UNIX cron utility or by a cron entry in the machine's */usr/afs/local/BosConfig* file. It is also useful if an administrator is unable to authenticate to AFS but is logged in as the local superuser root.

Do not combine the -cell and -localauth options. A command on which the -localauth flag is included always runs in the local cell (as defined in the server machine's local

`/usr/afs/etc/ThisCell` file), whereas a command on which the -cell argument is included runs in the specified foreign cell. Also, do not combine the -localauth and -noauth flags.

-noauth

Establishes an unauthenticated connection to the Volume Server and VL Server, in which the servers treat the issuer as the unprivileged user `anonymous`. It is useful only when authorization checking is disabled on the server machine (during the installation of a file server machine or when the bos setauth command has been used during other unusual circumstances). In normal circumstances, the servers allow only privileged users to issue commands that change the status of a volume or VLDB record, and refuses to perform such an action even if the -noauth flag is provided. Do not combine the -noauth and -localauth flags.

-partition <*partition name*>

Identifies the AFS server partition on a file server machine that houses, or is to house, the volumes of interest, or about which to list information. The vos command interpreter accepts any of the following four name formats:

```
/vicepa     =     vicepa      =      a      =      0
/vicepb     =     vicepb      =      b      =      1
```

After /vicepz (for which the index is 25) comes

```
/vicepaa    =     vicepaa     =      aa     =      26
/vicepab    =     vicepab     =      ab     =      27
```

and so on through

```
/vicepiv    =     vicepiv     =      iv     =      255
```

The -frompartition and -topartition arguments to the vos move command also accept this notation.

-server <*machine name*>

Identifies the file server machine that houses, or is to house, the volumes or AFS server partitions of interest. Provide the machine's IP address in dotted decimal format, its fully qualified host name (for example, `fs1.abc.com`), or the shortest abbreviated form of its host name that distinguishes it from other machines. Successful use of an abbreviated form depends on the availability of a name resolution service (such as the Domain Name Service or a local host table) at the time the command is issued.

The -fromserver and -toserver arguments to the vos move command also accept these name formats.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

Privilege Required

To issue most vos commands, the issuer must be listed in the */usr/afs/etc/UserList* file on each server machine that houses or is to house an affected volume, and on each database server machine. The most predictable performance results if all database server and file server machines in the cell share a common *UserList* file. Alternatively, if the -localauth flag is included, the issuer must be logged on to a server machine as the local superuser `root`.

To issue a vos command that only displays information, no privilege is required.

See Also

"CellServDB(5)" on page 415, "UserList(5)" on page 437, "vos_addsite(1)" on page 293, "vos_apropos(1)" on page 297, "vos_backup(1)" on page 299, "vos_backupsys(1)" on page 301, "vos_changeaddr(1)" on page 307, "vos_create(1)" on page 311, "vos_delentry(1)" on page 315, "vos_dump(1)" on page 319, "vos_examine(1)" on page 323, "vos_help(1)" on page 329, "vos_listaddrs(1)" on page 331, "vos_listpart(1)" on page 333, "vos_listvldb(1)" on page 335, "vos_listvol(1)" on page 339, "vos_lock(1)" on page 345, "vos_move(1)" on page 347, "vos_partinfo(1)" on page 351, "vos_release(1)" on page 355, "vos_remove(1)" on page 359, "vos_remsite(1)" on page 363, "vos_rename(1)" on page 367, "vos_restore(1)" on page 369, "vos_status(1)" on page 373, "vos_syncserv(1)" on page 377, "vos_syncvldb(1)" on page 379, "vos_unlock(1)" on page 383, "vos_unlockvldb(1)" on page 385, "vos_zap(1)" on page 389

Copyright

vos addsite

Name

`vos addsite` — Adds a read-only site definition to a volume's VLDB entry

Synopsis

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 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

The vos addsite command defines a new read-only site (partition on a file server machine, specified by the -server and -partition arguments) in the Volume Location Database (VLDB) entry of the read / write volume named by the -id argument. When the vos release command is next issued against the read / write volume, a read-only copy of it is distributed to all of the read-only sites, including the newly defined one.

Cautions

A volume's VLDB entry accommodates a maximum number of site definitions, as defined in the *IBM AFS Release Notes*. The site housing the read / write and backup versions of the volume counts as one site, and each read-only site counts as an additional site (even the read-only site defined on the same file server machine and partition as the read / write site counts as a separate site). The limit in the VLDB entry effectively determines the maximum number of copies of the volume that are available to AFS clients.

Attempts to create additional sites by using this command fail with an error.

Options

-server <*machine name*>

Identifies the file server machine where the read-only volume is to reside. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies the partition where the read-only volume is to reside, on the file server machine named by the -server argument. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-id <*volume name or ID*>

Specifies either the complete name or volume ID number of the read / write source volume.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example, appropriate in the State University cell, defines a read-only site for the cell's `root.afs` volume.

```
% vos addsite -server sv7.stateu.edu -partition /vicepb -id root.afs
```

Privilege Required

The issuer must be listed in the `/usr/afs/etc/UserList` file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_examine(1)" on page 323, "vos_release(1)" on page 355

Copyright

vos apropos

Name

`vos apropos` — Displays each help entry containing a keyword string

Synopsis

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

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

Description

The vos apropos command displays the first line of the online help entry for any vos command that has in its name or short description the string specified by the -topic argument.

To display the syntax for a command, use the vos help command.

Options

-topic <*help string*>

Specifies the keyword string to match. Use lowercase letters only, except for the acronym `VLDB`. If the string is more than a single word, surround it with double quotes ("") or other delimiters.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of a command's online help entry names it and briefly describes its function. This command displays the first line for any vos command where the string specified with the -topic argument is part of the command name or first line.

Examples

The following command displays all vos commands that include the word lock in their names or short descriptions:

```
% vos apropos 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

See Also

"vos(1)" on page 289, "vos_help(1)" on page 329

Copyright

vos backup

Name

`vos backup` — Creates a backup volume for a single read / write volume

Synopsis

vos backup -id <*volume name or ID*> [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos backup -i <*volume name or ID*> [-c<*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos backup command clones the indicated read / write volume to create a backup version, placing it at the same site as the read / write version. The backup volume's name is the same as the read / write source's with the addition of the `.backup` extension. Its volume ID number is the one allocated for it in the Volume Location Database (VLDB) when the read / write source was created with the vos create command. If a backup version already exists, the new clone replaces it.

To create a backup version of multiple volumes, use the vos backupsys command.

Options

-id <*volume name or ID*>

Specifies either the complete name or volume ID number of the read / write source volume.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The following message confirms that the command succeeded:

```
Created backup volume for I&lt;volume name&gt;
```

Examples

The following example creates a backup version of the volume `user.smith`.

```
% vos backup user.smith
Created backup volume for user.smith
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_backupsys(1)" on page 301

Copyright

vos backupsys

Name

vos backupsys — Creates a backup volume for several read / write volumes

Synopsis

vos backupsys [-prefix <*common prefix on volume(s)*>+] [-server <*machine name*>] [-partition <*partition name*>] [-exclude] [-xprefix <*negative prefix on volume(s)*>+] [-dryrun] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos backups [-pr <*common prefix on volume(s)*>+] [-s <*machine name*>] [-pa <*partition name*>] [-e] [-x <*negative prefix on volume(s)*>+] [-d] [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos backupsys command clones each indicated read / write volume to create a backup version, placing each clone at the same site as its read / write source version. It assigns each clone the same name as the read / write source, adding a .backup extension. It assigns the volume ID number already allocated for the backup version in the Volume Location Database (VLDB). If a backup version already exists for a given volume, the new clone replaces it.

To clone every read / write volume listed in the VLDB, omit all of the command's options. Otherwise, combine the command's options to clone various groups of volumes. The options use one of two basic criteria to select volumes: location (the -server and -partition arguments) or presence in the volume name of one of a set of specified character strings (the -prefix, -exclude, and -xprefix options).

To clone only volumes that reside on one file server machine, include the -server argument. To clone only volumes that reside on one partition, combine the -server and -partition arguments. The -partition argument can also be used alone to clone volumes that reside on the indicated partition on every file server machine. These arguments can be combined with those that select volumes based on their names.

Combine the -prefix, -exclude, and -xprefix options (with or without the -server and -partition arguments) in the indicated ways to select volumes based on character strings contained in their names:

- To clone every read / write volume at the specified location whose name includes one of a set of specified character strings (for example, begins with user. or includes the string afs), use the -prefix argument or combine the -xprefix and -exclude options.
- To clone every read / write volume at the specified location except those whose name includes one of a set of specified character strings, use the -xprefix argument or combine the -prefix and -exclude options.
- To clone every read / write volume at the specified location whose name includes one of one of a set of specified character strings, except those whose names include one of a different set of specified character strings, combine the -prefix and -xprefix arguments. The command creates a list of all volumes that match the -prefix argument and then removes from the list

the volumes that match the -xprefix argument. For effective results, the strings specified by the -xprefix argument must designate a subset of the volumes specified by the -prefix argument.

If the -exclude flag is combined with the -prefix and -xprefix arguments, the command creates a list of all volumes that do not match the -prefix argument and then adds to the list any volumes that match the -xprefix argument. As when the -exclude flag is not used, the result is effective only if the strings specified by the -xprefix argument designate a subset of the volumes specified by the -prefix argument.

The -prefix and -xprefix arguments both accept multiple values, which can be used to define disjoint groups of volumes. Each value can be one of two types:

- A simple character string, which matches volumes whose name begin with the string. All characters are interpreted literally (that is, characters that potentially have special meaning to the command shell, such as the period, have only their literal meaning).
- A regular expression, which matches volumes whose names contain the expressions. Place a caret (^) at the beginning of the expression, and enclose the entire string in single quotes (″). Explaining regular expressions is outside the scope of this reference page; see the UNIX manual page for regexp(5) or (for a brief introduction) "backup_addvolentry(8)" on page 541. As an example, the following expression matches volumes that have the string `aix` anywhere in their names:

  ```
  -prefix  '^.*aix'
  ```

To display a list of the volumes to be cloned, without actually cloning them, include the -dryrun flag. To display a statement that summarizes the criteria being used to select volume, include the -verbose flag.

This command can be used to clone a single read / write volume; specify its complete name as the -prefix argument. However, it is more efficient to use the vos backup command, which employs a more streamlined technique for finding a single volume.

Options

-prefix <*common prefix*>

Specifies one or more simple character strings or regular expressions of any length; a volume whose name includes the string is placed on the set of volumes to be cloned. Include field separators (such as periods) if appropriate. This argument can be combined with any combination of the -server, -partition, -exclude, and -xprefix options.

-server <*machine name*>

Identifies the file server machine where each read / write source volume resides. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

This argument can be combined with any combination of the -prefix, -partition, -exclude, and -xprefix options.

-partition <*partition name*>

Identifies the partition where each read / write source volume resides. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

This argument can be combined with any combination of the -prefix, -server, -exclude, and -xprefix options.

-exclude

Reverses the meaning of the -prefix or -xprefix argument. This flag can be combined with any combination of the -prefix, -server, -partition, and -xprefix options.

-xprefix <*negative prefix*>

Specifies a simple character string or regular expression of any length; a volume whose name includes the string is removed from the set of volumes to be cloned. Include field separators (such as periods) if appropriate. This argument can be combined with any combination of the -prefix, -server, -partition, and -exclude options.

-dryrun

Displays on the standard output stream a list of the volumes to be cloned, without actually cloning them.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The command generates the following messages on the standard output stream to confirm that the operation was successful:

```
done
Total volumes backed up: &lt;number_cloned&gt;; failed to backup: &lt;failures&gt;
```

If the -dryrun flag is included, a list of the volumes to be backed up precedes the standard confirmation messages.

If the -verbose flag is included but not the -dryrun flag, the following messages appear for each volume. The output concludes with the standard confirmation messages.

```
Creating backup volume for &lt;volume_name&gt; on &lt;date/time&gt;
{Recloning backup volume | Creating a new backup clone} &lt;backup_volumeID&gt; . . .don
```

If both the -dryrun and -verbose flags are included, the output begins with a statement summarizing the criteria being used to select the volumes, followed by a list of the volumes and the standard confirmation messages. The format of the criteria summary statement depends on which other options are provided:

- If only the -prefix argument is provided, or the -xprefix and -exclude options are combined:

  ```
  Would have backed up volumes which are prefixed with &lt;string&gt; [or &lt;string&gt;
  ```

- If only the -xprefix argument is provided, or the -prefix and -exclude options are combined:

  ```
  Would have backed up volumes which are not prefixed with &lt;string&gt; [nor &lt;string
  ```

- If the -prefix and -xprefix arguments are combined:

  ```
  Would have backed up volumes which are prefixed with &lt;string&gt; [or &lt;string&gt;
     removing those which are prefixed with &lt;x_string&gt; [or &lt;x_string&gt;] . .
  ```

- If the -prefix, -xprefix, and -exclude options are provided:

  ```
  Would have backed up volumes which are not prefixed with &lt;string&gt; [nor &lt;string
     adding those which are prefixed with &lt;x_string&gt; [or &lt;x_string&gt;] . .
  ```

Examples

The following example creates a backup version of every read / write volume listed in the cell's VLDB whose name begins with the string user.

```
% vos backupsys -prefix user
```

The following example, appropriate in the ABC Corporation cell, creates a backup version of every read / write volume on the file server machine `fs3.abc.com`.

```
% vos backupsys -server fs3.abc.com
```

The following example, appropriate in the State University cell, creates a backup version of every read / write volume on the file server machine `db1.stateu.edu` except those whose name includes the string `temp`.

```
% vos backupsys  -server db1.stateu.edu -prefix '^.*temp'
```

The following example creates a backup version of every volume listed in the cell's VLDB, excluding those whose names contain the string `source`, but including those whose names contain the string `source.current`.

```
% vos backupsys  -prefix '^.*source' -exclude -xprefix '^.*source\.current'
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"backup_addvolentry(8)" on page 541, "vos(1)" on page 289, "vos_backup(1)" on page 299

UNIX manual page for regexp(5)

Copyright

vos changeaddr

Name

vos changeaddr — Changes or removes a file server machine's entry in the VLDB

Synopsis

vos changeaddr -oldaddr <*original IP address*> [-newaddr <*new IP address*>] [-remove] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos ch -o <*original IP address*> [-ne <*new IP address*>] [-r] [-c <*cell name*>] [-no] [-l] [-v] [-h]

Description

The vos changeaddr command removes a server entry from the Volume Location Database (VLDB) when the -remove flag is combined with the -oldaddr argument. There must be no VLDB entries that list the machine as a site for any version of a volume (if necessary, use the vos move or vos remove command to more or remove volumes). It is appropriate to remove a VLDB server entry when removing the corresponding file server machine from service; this is the only recommended use of the command.

To display all VLDB server entries, use the vos listaddrs command.

Cautions

Combining the command's -oldaddr and -newaddr arguments is no longer the appropriate way to change the IP address registered for a file server machine. Furthermore, if a machine is multihomed and its server entry includes several addresses, then the address specified with the -newaddr argument replaces all of the addresses currently listed in the server entry that includes the address specified by the -oldaddr argument. This effectively makes the machine single-homed with respect to AFS operations, which is probably not the desired result.

The recommended method for changing the IP addresses in a server entry is instead to restart the `fs` process group (which includes the File Server) after using the utilities provided by the operating system to reconfigure the machine's network interfaces. For a description of how the File Server constructs and registers a list of its network interfaces in the VLDB, see "sysid(5)" on page 491.

If, counter to recommended usage, the command is used to change the IP address in a server entry, it does not also change the names of machine entries in the Protection Database. Operations fail when they refer to a protection group that has an obsolete IP address in it. Use the pts rename command to change the names of machine entries that correspond to the addresses changed with this command. Changing the address of a database server machine also requires updating the client and server versions of the *`CellServDB`* file on every machine.

Options

-oldaddr <*original IP address*>

Specifies the IP address currently registered for the file server machine in the VLDB server entry. If there are multiple addresses registered for a multihomed machine, use any of them to identify the server entry.

-newaddr <*new IP address*>

Specifies the new IP address that replaces all currently registered addresses.

-remove

Removes from the VLDB the server entry that includes the address specified by the -oldaddr argument.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command removes the VLDB server entry that includes the IP address `192.12.107.214`.

```
% vos changeaddr -oldaddr 192.12.107.214 -remove
```

Privilege Required

Issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -oldaddr argument and on each database server machine.

See Also

"CellServDB(5)" on page 415, "UserList(5)" on page 437, "sysid(5)" on page 491, "fileserver(8)" on page 771, "pts_rename(1)" on page 217, "vos(1)" on page 289, "vos_listaddrs(1)" on page 331

Copyright

vos create

Name

`vos create` — Creates a read / write volume and associated VLDB entry

Synopsis

vos create -server <*machine name*> -partition <*partition name*> -name <*volume name*> [-maxquota <*initial quota (KB)*>] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos cr -s <*machine name*> -p <*partition name*> -na <*volume name*> [-m <*initial quota (KB)*>] [-c <*cell name*>] [-no] [-l] [-v] [-h]

Description

The vos create command creates a read / write volume with the name specified by the -name argument at the site specified by the -server and -partition arguments. In addition, the command allocates or sets the following:

- Volume ID numbers for the read / write volume and its associated read-only and backup volumes (this command does not actually create the latter two types of volume). A volume ID number is an identification number guaranteed to be unique within a cell.
- An access control list (ACL) associated with the volume's root directory, which takes the same name as volume's mount point when the volume is mounted with the fs mkmount command. An entry that grants all seven permissions to the members of the system:administrators group is automatically placed on the ACL. (In addition, the File Server by default always implicitly grants the `l` (lookup) and `a` (administer) permissions on every ACL to members of the system:administrators group, even when the group does not appear on an ACL; use the -implicit argument to the fileserver initialization command to alter the set of rights on a server-by-server basis if desired.)
- The volume's space quota, set to 5000 kilobyte blocks by default. Use the -maxquota argument to specify a different quota, or use the fs setquota command to change the volume's quota after mounting the volume with the fs mkmount command.

The volume is empty when created. To access it via the Cache Manager, mount it in the file space by using the fs mkmount command.

Options

-server <*server name*>

Identifies the file server machine on which to create the read / write volume. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies the partition on which to create the read / write volume, on the file server machine specified by the -server argument. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-name <*volume name*>

Specifies a name for the read / write volume. The maximum length is 22 characters, which can include any alphanumeric or punctuation character. By convention, periods separate the fields in a name. Do not apply the `.backup` or `.readonly` extension to a read / write volume name; they are reserved for the Volume Server to add to the read / write name when creating those backup and read-only volumes respectively.

-maxquota <*volume quota*>

Specifies the maximum amount of disk space the volume can use, as a number of kilobyte blocks (a value of `1024` is one megabyte). The value `0` (zero) grants an unlimited quota, but the size of the disk partition that houses the volume places an absolute limit on its size. If this argument is omitted, the default value is `5000`.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The Volume Server produces the following message to confirm that it created the volume:

```
Volume &lt;volume_ID&gt; created on partition &lt;partition_name&gt; of &lt;machine_name
```

Examples

The following command creates the read/write volume `user.pat` on the */vicepf* partition of the file server machine `fs4.abc.com`.

```
% vos create -server fs4.abc.com -partition /vicepf -name user.pat
Volume user.pat created on partition /vicepf of fs4.abc.com
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289

Copyright

vos delentry

Name

`vos delentry` — Removes a volume entry from the VLDB.

Synopsis

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]

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

The vos delentry command removes the Volume Location Database (VLDB) entry for each specified volume. A specified volume can be any of the three types (read / write, read-only, 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 a volume removal operation did not update the VLDB (perhaps because the vos zap command was used), but the system administrator does not feel it is necessary to use the vos syncserv and vos syncvldb commands to synchronize an entire file server machine.

To remove the VLDB entry for a single volume, use the -id argument. To remove groups of volumes, combine the -prefix, -server, and -partition arguments. The following list describes how to remove the VLDB entry for the indicated group of volumes:

- For every volume whose name begins with a certain character string (for example, `sys.` or `user.`): use the -prefix argument.
- Every volume for which the VLDB lists a site on a certain file server machine: specify the file server name with the -server argument.
- Every volume for which the VLDB lists a site on a partition of the same name (for instance, on the */vicepa* partition on any file server machine): specify the partition name with the -partition argument.
- Every volume for which the VLDB lists a site one a specific partition of a file server machine: specify both the -server and -partition arguments.
- Every volume whose name begins with a certain prefix and for which the VLDB lists a site on a file server machine: combine the -prefix and -server arguments. Combine the -prefix argument with the -partition argument, or both the -server and -partition arguments, to remove a more specific group of volumes.

Cautions

Do not use this command to remove a volume in normal circumstances; it does not remove a volume from the file server machine, and so is likely to make the VLDB inconsistent with state of the volumes on server machines. Use the vos remove command to remove both the volume and its VLDB entry.

Options

-id <*volume name or ID*>+

Specifies the complete name or the volume ID number of each volume for which to remove the VLDB entry. The entire entry is removed, regardless of whether the read / write, read-only, or backup version is indicated. Provide this argument or some combination of the -prefix, -server, and -partition arguments.

-prefix <*prefix of volume entry*

Specifies a character string of any length; the VLDB entry for a volume whose name begins with the string is removed. Include field separators (such as periods) if appropriate. Combine this argument with the -server argument, -partition argument, or both.

-server <*server name*>

Identifies a file server machine; if a volume's VLDB entry lists a site on the machine, the entry is removed. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

Combine this argument with the -prefix argument, the -partition argument, or both.

-partition <*partition name*>

Identifies a partition; if a volume's VLDB entry lists a site on the partition, the entry is removed. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

Combine this argument with the -prefix argument, the -server argument, or both.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during

mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The following message confirms the success of the command by indicating how many VLDB entries were removed.

```
Deleted &lt;number&gt; VLDB entries
```

Examples

The following command removes the VLDB entry for the volume `user.temp`.

```
% vos delentry user.temp
```

The following command removes the VLDB entry for every volume whose name begins with the string `test` and for which the VLDB lists a site on the file server machine `fs3.abc.com`.

```
% vos delentry -prefix test -server fs3.abc.com
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_remove(1)" on page 359, "vos_syncserv(1)" on page 377, "vos_syncvldb(1)" on page 379, "vos_zap(1)" on page 389

Copyright

vos dump

Name

`vos dump` — Converts a volume into ASCII format and writes it to a file

Synopsis

vos dump -id <*volume name or ID*> [-time <*dump from time*>] [-file <*dump file*>] [-server <*server*>] [-partition <*partition*>] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos du -i <*volume name or ID*> [-t <*dump from time*>] [-f <*dump file*>] [-s <*server*>] [-p <*partition*>] [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos dump command converts the contents of the indicated volume, which can be read / write, read-only or backup, into ASCII format. The Volume Server writes the converted contents to the file named by the -file argument, or to the standard output stream. In the latter case, the output can be directed to a named pipe, which enables interoperation with third-party backup utilities.

To dump the complete contents of a volume (create a *full dump*), omit the -time argument or specify the value `0` (zero) for it. To create an *incremental dump*, which includes only the files and directories in the volume that have modification timestamps later than a certain time, specify a date and time as the value for the -time argument.

By default, the vos command interpreter consults the Volume Location Database (VLDB) to learn the volume's location, so the -server and -partition arguments are not required. If the -id argument identifies a read-only volume that resides at multiple sites, the command dumps the version from just one of them (normally, the one listed first in the volume's VLDB entry as reported by the vos examine or vos listvldb command). To dump the read-only volume from a particular site, use the -server and -partition arguments to specify the site. To bypass the VLDB lookup entirely, provide a volume ID number (rather than a volume name) as the value for the -id argument, together with the -server and -partition arguments. This makes it possible to dump a volume for which there is no VLDB entry.

During the dump operation, the volume is inaccessible both to Cache Managers and to other volume operations. Dumping a volume does not otherwise affect its status on the partition or its VLDB entry.

To restore a dumped volume back into AFS, use the vos restore command.

Cautions

Support for incremental dumps is provided to facilitate interoperation with third-party backup utilities. The vos dump command does not provide any of the administrative facilities of an actual backup system, so the administrator must keep manual records of dump times and the relationship between full and incremental dumps of a volume. For a volume's contents to be consistent after restoration of incremental dumps, there must be no gap between the time at

which a prior dump of the volume was created and the value of the -time argument to the vos dump command that creates the incremental dump. More specifically, for a read / write volume, the -time argument must specify the time that the prior dump was performed, and for a read-only or backup volume it must specify the time that the volume was last released (using the vos release command) or cloned (using the vos backup or vos backupsys command) prior to dumping it. The parent dump can be either a full dump or another incremental dump.

Options

-id <*volume name or ID*>

Specifies either the complete name or volume ID number of the read / write, read-only, or backup volume to dump.

-time <*dump from time*>

Specifies whether the dump is full or incremental. Omit this argument to create a full dump, or provide one of three acceptable values:

- The value `0` (zero) to create a full dump.
- A date in the format *mm*/*dd*/*yyyy* (month, day and year) to create an incremental dump that includes only files and directories with modification timestamps later than midnight (12:00 a.m.) on the indicated date. Valid values for the year range from `1970` to `2037`; higher values are not valid because the latest possible date in the standard UNIX representation is in 2038. The command interpreter automatically reduces later dates to the maximum value. An example is `01/13/1999`.
- A date and time in the format "*mm*/*dd*/*yyyy hh*:*MM*" to create an incremental dump that includes only files and directories with modification timestamps later than the specified date and time. The date format is the same as for a date alone. Express the time as hours and minutes (*hh*:*MM*) in 24-hour format (for example, 20:30 is 8:30 p.m.). Surround the entire expression with double quotes ("") because it contains a space. An example is `"01/13/1999 22:30"`.

-file <*dump file*>

Specifies the pathname of the file to which to write the dump. The file can be in AFS, but not in the volume being dumped. A partial pathname is interpreted relative to the current working directory. If this argument is omitted, the dump is directed to the standard output stream.

-server <*server name*>

Specifies the file server machine on which the volume resides. Provide the -partition argument along with this one.

-partition <*partition name*>

Specifies the partition on which the volume resides. Provide the -server argument along with this one.

-cell <*cell name*

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command writes a full dump of the volume `user.terry` to the file */afs/abc.com/common/dumps/terry.dump*.

```
% vos dump -id user.terry -time 0 -file /afs/abc.com/common/dumps/terry.dump
```

The following command writes an incremental dump of the volume `user.smith` to the file `smith.990131.dump` in the current working directory. Only those files in the volume with modification time stamps later than 6:00 p.m. on 31 January 1999 are included in the dump.

```
% vos dump -id user.smith -time "01/31/1999 18:00" -file smith.990131.dump
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

If the -file argument is included, the issuer must also have permission to insert and write in the directory that houses the file.

See Also

"vos(1)" on page 289, "vos_examine(1)" on page 323, "vos_listvldb(1)" on page 335, "vos_restore(1)" on page 369

Copyright

vos examine

Name

`vos examine` — Shows volume header and VLDB entry information for a volume

Synopsis

vos examine -id <*volume name or ID*> [-extended] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

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]

vos v -i <*volume name or ID*> [-e] [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos examine command formats and displays information from the Volume Location Database (VLDB) entry and the volume header of the volume specified by the -id argument.

To display the volume header only, use the vos listvol command. To display information from the VLDB only, use the vos listvldb command.

Options

-id <*volume name or ID*>

Specifies either the complete name or volume ID number of the volume, which can be read / write, read-only, or backup.

-extended

Display statistics about read and write operations on files and directories in the volume.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first seven lines of the output show information from the volume header and the remaining lines come from the VLDB. Each item in the following list corresponds to a line of output derived from the volume header.

- Basic information about the specified volume (displayed on a single line):
  - Name
  - Volume ID number
  - Type (the flag is `RW` for read / write, `RO` for read-only, `BK` for backup)
  - Size in kilobytes (`1024` equals a megabyte)
  - Number of files in the volume, if the -extended flag is provided
  - Status on the file server machine, which is one of the following:

    On-line

    The volume is completely accessible to Cache Managers.

    Off-line

    The volume is not accessible to Cache Managers, but does not seem to be corrupted. This status appears while a volume is being dumped, for example.

    Off-line**needs salvage**

    The volume is not accessible to Cache Managers, because it seems to be corrupted. Use the bos salvage or salvager command to repair the corruption.

- The file server machine and partition that house the volume, as determined by the command interpreter as the command runs, rather than derived from the VLDB or the volume header.
- The volume ID numbers associated with the various versions of the volume: read / write (`RWrite`), read-only (`ROnly`), backup (`Backup`), and ReleaseClone (`RClone`). One of them matches the volume ID number that appears on the first line of the volume's output. If the value in the `RWrite`, `ROnly`, or `Backup` field is `0` (zero), there is no volume of that type. If there is currently no ReleaseClone, the `RClone` field does not appear at all.

- The maximum space quota allotted to the read / write copy of the volume, expressed in kilobyte blocks in the `MaxQuota` field.
- The date and time the volume was created, in the `Creation` field. If the volume has been restored with the backup diskrestore, backup volrestore, or vos restore command, this is the restore time.
- The date and time when the contents of the volume last changed, in the `Last Update` field. For read-only and backup volumes, it matches the timestamp in the `Creation` field.
- The number of times the volume has been accessed for a fetch or store operation 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

When the -extended flag is included, two tables appear next:

- The table labeled `Raw Read/Write Stats` contains information on the number of reads (fetches) and writes (stores) made on the specified volume.
- The table labeled `Writes Affecting Authorship` contains information on writes made to files and directories in the specified volume.

If the following message appears instead of the previously listed information, it indicates that a volume is not accessible to Cache Managers or the vos command interpreter, for example because a clone is being created.

```
**** Volume &lt;volume_ID&gt; is busy ****
```

If the following message appears instead of the previously listed information, it indicates that the File Server is unable to attach the volume, perhaps because it is seriously corrupted. The *`FileLog`* and *`VolserLog`* log files in the */usr/afs/logs* directory on the file server machine possibly provide additional information; use the bos getlog command to display them.

```
**** Could not attach volume &lt;volume_ID&gt; ****
```

Following a blank line, information from the VLDB entry appears. Each item in this list corresponds to a separate line in the output:

- The base (read / write) volume name. The read-only and backup versions have the same name with a `.readonly` and `.backup` extension, respectively.
- The volume ID numbers allocated to the versions of the volume that actually exist, in fields labeled `RWrite` for the read / write, `ROnly` for the read-only, `Backup` for the backup, and `RClone` for the ReleaseClone. (If a field does not appear, the corresponding version of the volume does not exist.) The appearance of the `RClone` field normally indicates that a release operation did not complete successfully; the `Old release` and `New release` flags often also appear on one or more of the site definition lines described just following.

- The number of sites that house a read / write or read-only copy of the volume, following the string `number of sites ->`.
- A line for each site that houses a read / write or read-only copy of the volume, specifying the file server machine, partition, and type of volume (`RW` for read / write or `RO` for read-only). If a backup version exists, it is understood to share the read / write site. Several flags can appear with a site definition:

  Not released

  Indicates that the vos release command has not been issued since the vos addsite command was used to define the read-only site.

  Old release

  Indicates that a vos release command did not complete successfully, leaving the previous, obsolete version of the volume at this site.

  New release

  Indicates that a vos release command did not complete successfully, but that this site did receive the correct new version of the volume.

- If the VLDB entry is locked, the string `Volume is currently LOCKED`.

For further discussion of the `New release` and `Old release` flags, see "vos_release(1)" on page 355.

Examples

The following example shows output for the ABC Corporation volume called `usr` with two read-only replication sites (this volume is mounted at the */afs/abc.com/usr* directory). For the sake of illustration, the output shows the volume as locked.

```
% vos examine usr
usr                          536870981 RW   3459 K On-line
     fs2.abc.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)
     RWrite: 5360870981   ROnly: 536870982   Backup: 536870983
     number of sites -&gt; 3
        server fs1.abc.com partition /vicepa RO Site
        server fs3.abc.com partition /vicepa RO Site
        server fs2.abc.com partition /vicepb RW Site
     Volume is currently LOCKED
```

The following example shows the output for the volume `user.terry` using the -extended flag. The volume has no read-only replication sites.

```
% vos examine -id user.terry -extended
user.terry         354287190 RW    2302 K used 119 files On-line
    fs4.abc.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 |
&gt; 1wk     |        0 |        0 |        0 |        0 |
           |-----------------------------------------|
    RWrite: 354287190    Backup: 354287192
    number of sites -&gt; 1
       server fs4.abc.com partition /vicepc RW Site
```

Privilege Required

None

See Also

"backup_diskrestore(8)" on page 565, "backup_volrestore(8)" on page 647, "bos_getlog(8)" on page 691, "bos_salvage(8)" on page 727, "salvager(8)" on page 879, "vos(1)" on page 289, "vos_listvol(1)" on page 339, "vos_listvldb(1)" on page 335, "vos_release(1)" on page 355, "vos_restore(1)" on page 369

Copyright

vos help

Name

`vos help` — Displays help for vos commands

Synopsis

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

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

Description

The vos help command displays the complete online help entry (short description and syntax statement) for each command operation code specified by the -topic argument. If the -topic argument is omitted, the output includes the first line (name and short description) of the online help entry for every vos command.

To list every vos command whose name or short description includes a specified keyword, use the vos apropos command.

Options

-topic <*help string*>+

Identifies each command for which to display the complete online help entry. Omit the vos part of the command name, providing only the operation code (for example, specify create, not vos create). If this argument is omitted, the output briefly describes every vos command.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The first line names the command and briefly describes its function.
- The second line lists aliases for the command, if any.
- The final line, which begins with the string `Usage`, lists the command's options in the prescribed order. Online help entries use the same symbols (for example, brackets) as the reference pages in this document.

Examples

The following command displays the online help entry for the vos create command:

```
% vos help create
vos create: create a new volume
Usage: vos create -server &lt;machine name&gt; -partition &lt;partition name&gt;
-name &lt;volume name&gt; [-cell &lt;cell name&gt;] [-noauth] [-localauth]
[-verbose] [-help]
```

Privilege Required

None

See Also

"vos(1)" on page 289, "vos_apropos(1)" on page 297

Copyright

vos listaddrs

Name

`vos listaddrs` — Displays all VLDB server entries

Synopsis

vos listaddrs [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos lista [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos listaddrs command displays all of the server entries from the Volume Location Database (VLDB). An entry is created as the File Server initializes and registers the contents of its */usr/afs/local/sysid* file in the VLDB.

Options

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output displays all server entries from the VLDB, each on its own line. If a file server machine is multihomed, all of its registered addresses appear on the line. The first one is the

one reported as a volume's site in the output from the vos examine and vos listvldb commands.

The VLDB records IP addresses, and the command interpreter has the local name service (either a process like the Domain Name Service or a local host table) translate them to hostnames before displaying them. If an IP address appears in the output, it is not possible to translate it.

The existence of an entry does not necessarily indicate that the machine that is still an active file server machine. To remove obsolete server entries, use the vos changeaddr command with the -remove argument.

Examples

The following command displays the VLDB server entries in the ABC Corporation cell:

```
% vos listaddrs
sv5.abc.com
sv1.abc.com
sv2.abc.com  afs2.abc.com
sv6.abc.com
```

Privilege Required

None

See Also

"sysid(5)" on page 491, "vos(1)" on page 289, "vos_changeaddr(1)" on page 307, "vos_examine(1)" on page 323, "vos_listvldb(1)" on page 335

Copyright

vos listpart

Name

`vos listpart` — Displays all AFS partitions on a file server machine

Synopsis

vos listpart -server <*machine name*> [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos listp -s <*machine name*> [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos listpart command displays all of the valid AFS partitions on the indicated file server machine, without consulting the Volume Location Database (VLDB). The vos partinfo command reports the size of a partition and the available space on that partition.

Options

-server <*server name*>

Identifies the file server machine for which to list the partitions. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local /usr/afs/etc/KeyFile file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output consists of a list of partition names of the form */vicepxx*, following the header:

```
The partitions on the server are:
```

The last line of the output reports the total number of partitions.

Examples

The following command displays the partitions on `fs1.abc.com`:

```
% vos listpart fs1.abc.com
The partitions on the server are:
    /vicepa     /vicepb     /vicepc     /vicepd
Total:  4
```

Privilege Required

None

See Also

"vos(1)" on page 289, "vos_partinfo(1)" on page 351

Copyright

vos listvldb

Name

`vos listvldb` — Displays a volume's VLDB entry

Synopsis

vos listvldb [-name <*volume name or ID*>] [-server <*machine name*>] [-partition <*partition name*>] [-locked] [-quiet] [-nosort] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos listvl [-na <*volume name or ID*>] [-s <*machine name*>] [-p <*partition name*>] [-lock] [-q] [-nos] [-c <*cell name*>] [-noa] [-loca] [-v] [-h]

Description

The vos listvldb command formats and displays information from the Volume Location Database (VLDB) entry for each volume specified. The output depends on the combination of options supplied on the command line. Combine options as indicated to display the desired type of VLDB entries:

- Every entry in the VLDB: provide no options.
- Every VLDB entry that mentions a certain file server machine as the site for a volume: specify the machine's name as the -server argument.
- Every VLDB entry that mentions a certain partition on any file server machine as the site for a volume: specify the partition name as the -partition argument.
- Every VLDB entry that mentions a certain partition on a certain file server machine as the site for a volume: combine the -server and -partition arguments.
- A single VLDB entry: specify a volume name or ID number with the -name argument.
- 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.

Options

-name <*volume name or ID*>

Specifies either the complete name or volume ID number of a volume of any of the three types.

-server <*server name*>

Identifies the file server machine listed as a site in each VLDB entry to display. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

This argument can be combined with the -partition argument, the -locked flag, or both.

-partition <*partition name*>

Identifies the partition (on the file server machine specified by the -server argument) listed as a site in each VLDB entry to display. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

This argument can be combined with the -server argument, the -locked flag, or both.

-locked

Displays only locked VLDB entries. This flag can be combined with the -server argument, the -partition argument, or both.

-quiet

Suppresses the lines that summarize the number of volumes listed and their status, which otherwise appear at the beginning and end of the output when the output includes more than one volume.

-nosort

Suppresses the default sorting of volume entries alphabetically by volume name.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If the output includes more than one VLDB entry, by default the first line reports which file server machine, partition, or both, houses the volumes. The final line of output reports the total number of entries displayed. Including the -quiet flag suppresses these lines.

By default, volumes are sorted alphabetically by volume name. Including the -nosort flag skips the sorting step, which can speed up the production of output if there are a large number of entries.

The VLDB entry for each volume includes the following information:

- The base (read / write) volume name. The read-only and backup versions have the same name with a `.readonly` and `.backup` extension, respectively.
- The volume ID numbers allocated to the versions of the volume that actually exist, in fields labeled `RWrite` for the read / write, `ROnly` for the read-only, `Backup` for the backup, and `RClone` for the ReleaseClone. (If a field does not appear, the corresponding version of the volume does not exist.) The appearance of the `RClone` field normally indicates that a release operation did not complete successfully; the `Old release` and `New release` flags often also appear on one or more of the site definition lines described just following.
- The number of sites that house a read / write or read-only copy of the volume, following the string `number of sites ->`.
- A line for each site that houses a read / write or read-only copy of the volume, specifying the file server machine, partition, and type of volume (`RW` for read / write or `RO` for read-only). If a backup version exists, it is understood to share the read / write site. Several flags can appear with a site definition:

  Not released

  Indicates that the vos release command has not been issued since the vos addsite command was used to define the read-only site.

  Old release

  Indicates that a vos release command did not complete successfully, leaving the previous, obsolete version of the volume at this site.

  New release

  Indicates that a vos release command did not complete successfully, but that this site did receive the correct new version of the volume.

- If the VLDB entry is locked, the string `Volume is currently LOCKED`.

For further discussion of the `New release` and `Old release` flags, see "vos_release(1)" on page 355.

Examples

The following command displays VLDB information for the ABC Corporation volume called `usr`, which has two read-only replication sites:

```
% vos listvldb -name usr
usr
 RWrite: 5360870981   ROnly: 536870982   Backup: 536870983
```

```
 number of sites -&gt; 3
    server fs1.abc.com partition /vicepa RO Site
    server fs3.abc.com partition /vicepa RO Site
    server fs2.abc.com partition /vicepb RW Site
```

The following example shows entries for two of the volumes that reside on the file server machine `fs4.abc.com`. The first VLDB entry is currently locked. There are 508 entries that mention the machine as a volume site.

```
% vos listvldb -server fs4.abc.com
VLDB entries for server fs4.abc.com
    .       .           .        .
    .       .           .        .
user.smith
 RWrite: 278541326   ROnly: 278541327   Backup: 278542328
 number of sites -&gt; 1
   server fs4.abc.com partition /vicepg RW Site
 Volume is currently LOCKED
   user.terry
 RWrite 354287190   ROnly 354287191   Backup 354287192
 number of sites -&gt; 1
   server fs4.abc.com partition /vicepc RW Site
   .       .           .        .
   .       .           .        .
Total entries: 508
```

Privilege Required

None

See Also

"vos(1)" on page 289, "vos_examine(1)" on page 323, "vos_listvol(1)" on page 339, "vos_lock(1)" on page 345, "vos_unlock(1)" on page 383, "vos_unlockvldb(1)" on page 385

Copyright

vos listvol

Name

`vos listvol` — Displays information from a volume header

Synopsis

vos listvol -server <*machine name*> [-partition <*partition name*>] [-fast] [-long] [-quiet] [-extended] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos listvo -s <*machine name*> [-p <*partition name*>] [-f] [-lon] [-q] [-e] [-c <*cell name*>] [-n] [-loc] [-v] [-h]

Description

The vos listvol command formats and displays the following information from the volume header of 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 various numbers of volumes, combine the command's arguments as indicated:

- For every volume on a file server machine, specify the machine's name with the -server argument.
- For every volume at a particular site, combine the -server argument with the -partition argument.

To display the Volume Location Database (VLDB) entry for one or more volumes, use the vos listvldb command. To display both the VLDB entry and the volume header for a single volume, use the vos examine command.

Options

-server <*server name*>

Identifies the file server machine that houses volumes for which to display the header. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

This argument can be combined with the -partition argument, as well as the -fast, -long, or -extended flag.

-partition <*partition name*>

Identifies the partition (on the file server machine specified by the -server argument) that houses volumes for which to display the header. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-fast

Displays only the volume ID numbers of volumes stored at the site specified by the -server, and optionally -partition, argument. Do not combine this flag with the -extended flag.

-long

Displays more detailed information about each volume stored at the site specified by the -server, and optionally -partition, argument. The information includes the volume IDs of all three volume types associated with the volume, and the read / write volume's quota, creation date and update date.

-quiet

Suppresses the lines that summarize the number of volumes listed and their status, which otherwise appear at the beginning and end of the output when the output includes more than one volume.

-extended

Displays extensive statistics about access patterns for each volume stored at the site specified by the -server, and optionally -partition, argument. The statistics include the number of reads and writes to files in the volume, and how recently files and directories have been updated by their owners or other users. Do not combine this flag with the -fast flag.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output is ordered alphabetically by volume name and by default provides the following information on a single line for each volume:

- Name
- Volume ID number
- Type (the flag is `RW` for read / write, `RO` for read-only, `BK` for backup)
- Size in kilobytes (`1024` equals a megabyte)
- Number of files in the volume, if the -extended flag is provided
- Status on the file server machine, which is one of the following:

  On-line

  The volume is completely accessible to Cache Managers.

  Off-line

  The volume is not accessible to Cache Managers, but does not seem to be corrupted. This status appears while a volume is being dumped, for example.

  Off-line**needs salvage**

  The volume is not accessible to Cache Managers, because it seems to be corrupted. Use the bos salvage or salvager command to repair the corruption.

If the following message appears instead of the previously listed information, it indicates that a volume is not accessible to Cache Managers or the vos command interpreter, for example because a clone is being created.

```
**** Volume &lt;volume_ID&gt; is busy ****
```

If the following message appears instead of the previously listed information, it indicates that the File Server is unable to attach the volume, perhaps because it is seriously corrupted. The *`FileLog`* and *`VolserLog`* log files in the */usr/afs/logs* directory on the file server machine possibly provide additional information; use the bos getlog command to display them.

```
**** Could not attach volume &lt;volume_ID&gt; ****
```

The information about individual volumes is bracketed by summary lines. The first line of output specifies the number of volumes in the listing. The last line of output summarizes the number of volumes that are online, offline, and busy. These lines do not appear if the -quiet flag is used.

If the -fast flag is added, the output displays only the volume ID number of each volume, arranged in increasing numerical order. The final line (which summarizes the number of online, offline, and busy volumes) is omitted.

If the -long flag is included, the output for each volume includes all of the information in the default listing plus the following. Each item in this list corresponds to a separate line of output:

- The file server machine and partition that house the volume, as determined by the command interpreter as the command runs, rather than derived from the VLDB or the volume header.
- The volume ID numbers associated with the various versions of the volume: read / write (`RWrite`), read-only (`ROnly`), backup (`Backup`), and ReleaseClone (`RClone`). One of them matches the volume ID number that appears on the first line of the volume's output. If the value in the `RWrite`, `ROnly`, or `Backup` field is `0` (zero), there is no volume of that type. If there is currently no ReleaseClone, the `RClone` field does not appear at all.
- The maximum space quota allotted to the read / write copy of the volume, expressed in kilobyte blocks in the `MaxQuota` field.
- The date and time the volume was created, in the `Creation` field. If the volume has been restored with the backup diskrestore, backup volrestore, or vos restore command, this is the restore time.
- The date and time when the contents of the volume last changed, in the `Last Update` field. For read-only and backup volumes, it matches the timestamp in the `Creation` field.
- The number of times the volume has been accessed for a fetch or store operation 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 -extended flag is included, the output for each volume includes all of the information reported with the -long flag, plus two tables of statistics:

- The table labeled `Raw Read/Write Stats` table summarizes the number of times the volume has been accessed for reading or writing.
- The table labeled `Writes Affecting Authorship` table contains information on writes made to files and directories in the specified volume.

Examples

The following example shows the output for the */vicepb* partition on the file server machine `fs2.abc.com` when no flags are provided:

```
% vos listvol -server fs2.abc.com -partition b
Total number of volumes on server fs2.abc.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 example shows the output when the -fast flag is added:

```
% vos listvol -server fs2.abc.com -partition b -fast
Total number of volumes on server fs2.abc.com partition /vicepb : 66
 1969516782
 1969516784
     .
     .
 1969535796
```

The following example shows two volumes from the output that appears when the -long flag is added:

```
% vos listvol -server fs2.abc.com -partition b -long
Total number of volumes on server fs2.abc.com partition /vicepb: 66
      .                  .      .         .   .    .
      .                  .      .         .   .    .
user.pat             1969534536 RW      17518 K On-line
     fs2.abc.com /vicepb
     RWrite 1969534536 ROnly 0        Backup 1969534538
     MaxQuota      20000 K
     Creation    Mon Jun 12 09:02:25 1989
     Last Update Thu May 20 17:39:34 1999
     1573 accesses in the past day (i.e., vnode references)
user.pat.backup      1969534538 BK      17537 K On-line
     fs2.abc.com /vicepb
     RWrite 1969534536 ROnly 0        Backup 1969534538
     MaxQuota      20000 K
     Creation    Tue Jun 13 04:37:59 1989
     Last Update Wed May 19 06:37:59 1999
     0 accesses in the past day (i.e., vnode references)
       .                  .      .         .   .    .
       .                  .      .         .   .    .
Total volumes onLine 66 ; Total volumes offLine 0 ; Total busy 0
```

Privilege Required

None

See Also

"backup_diskrestore(8)" on page 565, "backup_volrestore(8)" on page 647, "bos_getlog(8)" on page 691, "bos_salvage(8)" on page 727, "salvager(8)" on page 879, "vos(1)" on page 289, "vos_examine(1)" on page 323, "vos_listvldb(1)" on page 335, "vos_restore(1)" on page 369

Copyright

vos lock

Name

`vos lock` — Locks a VLDB volume entry

Synopsis

vos lock -id <*volume name or ID*> [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos lo -i <*volume name or ID*> [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos lock command locks the Volume Location Database (VLDB) entry for the indicated volume, blocking any operation that requires a write to that entry. The lock applies to all of the volume versions associated with the entry, not just the one specified with the -id argument.

To unlock a single VLDB entry, use the vos unlock command. To unlock several entries, or all locked entries in the VLDB, use the vos unlockvldb command.

Cautions

Do not use this command in normal circumstances. It is useful for guaranteeing that the volume stays unchanged when there is reason to believe that volume operations cannot properly lock VLDB volume entries as they normally do to synchronize with one another.

Options

-id <*volume name or ID*>

Specifies either the complete name or volume ID number of a volume of the any of the three types.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command locks the VLDB entry for `user.terry`.

```
% vos lock user.terry
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_unlock(1)" on page 383, "vos_unlockvldb(1)" on page 385

Copyright

vos move

Name

vos move — Moves a read / write volume to another site

Synopsis

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]

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

The vos move command moves the indicated read / write volume from its current site (specified with the -fromserver and -frompartition arguments) to the destination site (specified with the -toserver and -topartition arguments). This command automatically removes the backup copy from the current site, if it exists. To create a new backup volume at the destination site, use the vos backup command.

This command works on read / write volumes only. To move a read-only volume, use the vos addsite and vos release commands to define a new read-only site and release the volume contents to it, and then use the vos remove command to remove the previous read-only volume's definition from the Volume Location Database (VLDB) and data from the partition. To move a backup volume, use this command to move its read / write source and then issue the vos backup command.

Before executing this command, the vos command interpreter initiates a check that the destination partition contains enough space to house the volume being moved. If there is not enough space, the move operation is not attempted and the following message appears:

```
vos: no space on target partition &lt;dest_part&gt; to move volume &lt;volume&gt;
```

Cautions

Unless there is a compelling reason, do not interrupt a vos move command in progress. Interrupting a move can result in one or more of the following inconsistent states:

- There are two versions of the volume, one at the source site and one at the destination site. (If this happens, retain the version identified by the VLDB and use the vos zap command to remove the other version.)
- The backup version of the volume is stranded at the old site. (If this happens, use the vos zap command to remove it.)

- The volume is off-line. (If this happens, run the bos salvage command to bring it back on line.)

If the Ctrl-C interrupt signal is pressed while a vos move operation is executing, the following message warns of the consequences and requests confirmation of the kill signal:

```
SIGINT handler: vos move operation in progress
WARNING: may leave AFS storage and metadata in indeterminate state
enter second control-c to exit
```

To confirm termination of the operation, press Ctrl-C a second time; press any other key to continue the operation.

Options

-id <*volume name or ID*>

Specifies either the complete name or volume ID number of a read / write volume.

-fromserver <*server name*>

Identifies the file server machine where the volume currently resides. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-frompartition <*partition name*>

Names the partition where the volume currently resides. Provide the full partition name (for, example, /vicepa) or one of the abbreviated forms described in "vos(1)" on page 289.

-toserver <*server name*>

Identifies the file server machine to which to move the volume. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-topartition <*partition name*>

Names the partition to which to move the volume. Provide the full partition name (for, example, /vicepa) or one of the abbreviated forms described in "vos(1)" on page 289.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example moves the volume `user.smith` from the */vicepb* partition on the file server machine `fs3.abc.com` to the */vicepg* partition on the file server machine `fs7.abc.com`.

```
% vos move -id user.smith -fromserver fs3.abc.com -frompartition b \
    -toserver fs7.abc.com -topartition g
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machines specified with the -toserver and -fromserver arguments and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_addsite(1)" on page 293, "vos_backup(1)" on page 299, "vos_release(1)" on page 355, "vos_listvol(1)" on page 339, "vos_remove(1)" on page 359

Copyright

vos partinfo

Name

`vos partinfo` — Reports the available and total space on a partition

Synopsis

vos partinfo -server <*machine name*> [-partition <*partition name*>] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos p -s <*machine name*> [-p <*partition name*>] [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos partinfo command reports the amount of space available and total size on either all of the partitions on the indicated file server machine (if the -partition argument is omitted) or the specified partition on that file server machine. The Volume Location Database (VLDB) is not consulted.

Options

-server <*server name*>

Identifies the file server machine for which to display partition information. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies which partition on the file server machine specified by the -server argument for which to display information. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during

mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Cautions

The partition-related statistics in this command's output do not always agree with the corresponding values in the output of the standard UNIX df command. The statistics reported by this command can be up to five minutes old, because the Cache Manager polls the File Server for partition information at that frequency. Also, on some operating systems, the df command's report of partition size includes reserved space not included in this command's calculation, and so is likely to be about 10% larger.

Output

The output reports the amount of space available and total space for each specified partition.

Examples

The following command displays all partitions on the file server machine `fs2.abc.com`.

```
% vos partinfo fs2.abc.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

See Also

"vos(1)" on page 289, "vos_listpart(1)" on page 333

Copyright

vos release

Name

`vos release` — Updates read-only volumes to match the read / write source volume

Synopsis

vos release -id <*volume name or ID*> [-f] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos rel -i <*volume name or ID*> [-f] [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos release command copies the contents of the indicated read / write source volume to each read-only site defined in the source volume's Volume Location Database (VLDB) entry. (Use the vos addsite command to define sites as necessary before issuing this command). Each read-only copy has the same name as read / write source with the addition of a `.readonly` extension.

For users to have a consistent view of the file system, the release of the new volume version must be atomic: either all read-only sites receive the new version, or all sites keep the version they currently have. The vos release command is designed to ensure that all copies of the volume's read-only version match both the read / write source and each other. In cases where problems such as machine or server process outages prevent successful completion of the release operation, AFS uses two mechanisms to alert the administrator.

First, the command interpreter generates an error message on the standard error stream naming each read-only site that did not receive the new volume version. Second, during the release operation the Volume Location (VL) Server marks site definitions in the VLDB entry with flags (`New release` and `Old release`) that indicate whether or not the site has the new volume version. If any flags remain after the operation completes, it was not successful. The Cache Manager refuses to access a read-only site marked with the `Old release` flag, which potentially imposes a greater load on the sites marked with the `New release` flag. It is important to investigate and eliminate the cause of the failure and then to issue the vos release command as many times as necessary to complete the release without errors.

The pattern of site flags remaining in the volume's VLDB entry after a failed release operation can help determine the point at which the operation failed. Use the vos examine or vos listvldb command to display the VLDB entry. The VL Server sets the flags in concert with the Volume Server's operations, as follows:

- Before the operation begins, the VL Server sets the `New release` flag on the read / write site definition in the VLDB entry and the `Old release` flag on read-only site definitions (unless the read-only site has been defined since the last release operation and has no actual volume, in which case its site flag remains `Not released`).
- If necessary, the Volume Server creates a temporary copy (a *clone*) of the read / write source called the ReleaseClone (see the following discussion of when the Volume Server does or

does not create a new ReleaseClone.) It assigns the ReleaseClone its own volume ID number, which the VL Server records in the `RClone` field of the source volume's VLDB entry.

- The Volume Server distributes a copy of the ReleaseClone to each read-only site defined in the VLDB entry. As the site successfully receives the new clone, the VL Server sets the site's flag in the VLDB entry to `New release`.
- When all the read-only copies are successfully released, the VL Server clears all the `New release` site flags. The ReleaseClone is no longer needed, so the Volume Server deletes it and the VL Server erases its ID from the VLDB entry.

By default, the Volume Server determines automatically whether or not it needs to create a new ReleaseClone:

- If there are no flags (`New release`, `Old release`, or `Not released`) on site definitions in the VLDB entry, the previous vos release command completed successfully and all read-only sites currently have the same volume. The Volume Server infers that the current vos release command was issued because the read / write volume has changed. The Volume Server creates a new ReleaseClone and distributes it to all of the read-only sites.
- If any site definition in the VLDB entry is marked with a flag, either the previous release operation did not complete successfully or a new read-only site was defined since the last release. The Volume Server does not create a new ReleaseClone, instead distributing the existing ReleaseClone to sites marked with the `Old release` or `Not released` flag. As previously noted, the VL Server marks each VLDB site definition with the `New release` flag as the site receives the ReleaseClone, and clears all flags after all sites successfully receive it.

To override the default behavior, forcing the Volume Server to create and release a new ReleaseClone to the read-only sites, include the -f flag. This is appropriate if, for example, the data at the read / write site has changed since the existing ReleaseClone was created during the previous release operation.

Options

-id <*volume name or id*>

Specifies either the complete name or volume ID number of a read / write volume.

-f

Creates a new ReleaseClone and distributes it all read-only sites regardless of whether or not any site definitions in the VLDB entry are marked with a flag.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command clones the read/write volume usr and releases it to the read-only sites defined in its VLDB entry.

```
% vos release usr
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_addsite(1)" on page 293, "vos_examine(1)" on page 323, "vos_listvldb(1)" on page 335

Copyright

vos remove

Name

`vos remove` — Removes a volume from a site

Synopsis

vos remove [-server <*machine name*>] [-partition <*partition name*>] -id <*volume name or ID*> [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos remo [-s <*machine name*>] [-p <*partition name*>] -i <*volume name or ID*> [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos remove command removes the indicated volume from the partition on which it resides. The Volume Location Database (VLDB) record is altered appropriately, as described in the following paragraphs. Use this command to remove any of the three types of volumes; the effect depends on the type.

- If the -id argument names the read / write volume (that is, specifies the volume's base name), both it and the associated backup volume are removed from the partition that houses them. The -server and -partition arguments are optional, because there can be only one read / write site. When the volume is removed, the site information is also removed from the VLDB entry. The read / write and backup volume ID numbers no longer appear in the output from the vos listvldb or vos examine commands, but they are preserved internally. Read-only sites, if any, are not affected, but cannot be changed unless a read / write site is again defined. The site count reported by the vos examine and vos listvldb commands as `number of sites` decrements by one. The entire VLDB entry is removed if there are no read-only sites.
- If the -id argument names a read-only volume, it is removed from the partition that houses it, and the corresponding site information is removed from the VLDB entry. The site count reported by the vos examine and vos listvldb commands as `number of sites` decrements by one for each volume you remove. If there is more than one read-only site, the -server argument (and optionally -partition argument) must be used to specify the site from which to remove the volume. If there is only one read-only site, the -id argument is sufficient; if there is also no read / write volume in this case, the entire VLDB entry is removed.
- If the -id argument names a backup volume, it is removed from the partition that houses it. The -server and -partition arguments are optional, because there can be only one backup site. The backup volume ID number no longer appears in the output from the vos listvldb command or in the corresponding portion of the output from the vos examine command, but is preserved internally.

This command is the most appropriate one for removing volumes in almost all cases. Other commands that remove only volumes or only VLDB entries (such as the vos delentry, vos remsite and vos zap commands) by definition can put the volumes and VLDB out of sync. Use them only in the special circumstances mentioned on their reference pages. Like the vos

delentry command, this command can remove a VLDB entry when no corresponding volumes exist on the file server machine. Like the vos zap command, this command can remove a volume that does not have a VLDB entry, as long as the volume is online, -server and -partition arguments are provided, and the -id argument specifies the volume's ID number.

Options

-server <*server name*>

Identifies the file server machine that houses the volume to remove. It is necessary only when the -id argument names a read-only volume that exists at multiple sites. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies the partition (on the file server machine specified by the -server argument) that houses the volume to remove. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

Including this argument is necessary only when the -id argument names a read-only volume that exists at multiple sites. Provide the -server argument along with this one.

-id <*volume name or id*>

Identifies the volume to remove, either by its complete name or volume ID number. If identifying a read-only or backup volume by name, include the appropriate extension (`.readonly` or `.backup`).

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example removes the read/write volume `user.terry` and its backup version, if any.

```
% vos remove  -id user.terry
```

The following example removes the read-only volume `root.afs.readonly` from one of its sites, the */vicepa* partition on the file server machine `fs1.abc.com`.

```
% vos remove fs1.abc.com  a  root.afs.readonly
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_delentry(1)" on page 315, "vos_remsite(1)" on page 363, "vos_zap(1)" on page 389

Copyright

vos remsite

Name

`vos remsite` — Removes a read-only site definition from a VLDB entry

Synopsis

vos remsite -server <*machine name*> -partition <*partition name*> -id <*volume name or ID*> [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos rems -s <*machine name*> -p <*partition name*> -i <*volume name or ID*> [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos remsite command removes the read-only replication site specified by the -machine and -partition arguments from the Volume Location Database (VLDB) entry for the indicated volume, which is read / write.

This command is useful for removing read-only sites that were mistakenly created with the vos addsite command, before the vos release command actually releases them. If a read-only copy already exists at the site, it is not affected. However, if this read-only site was the last site housing any version of the volume, then the entire VLDB entry is removed, even if a copy of the read-only version still actually exists at the site. The VL Server does not correct the discrepancy until the vos syncserv and vos syncvldb commands are run.

Cautions

Do not use this command as the standard way to remove a read-only volume, because it can create a discrepancy between the VLDB and the volumes on file server machines. Use the vos remove command instead.

Options

-server <*server name*>

Specifies the file server machine portion of the site definition to remove. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Specifies the partition name portion of the site definition to remove. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-id <*volume name or id*>

Specifies either the complete name or volume ID number of the read / write volume to remove.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command removes the mistakenly defined read-only site */viceph* on the file server machine `fs5.abc.com` from the VLDB entry for the volume `root.cell`.

```
% vos remsite -server fs5.abc.com -partition h -id root.cell
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_delentry(1)" on page 315, "vos_remove(1)" on page 359, "vos_zap(1)" on page 389

Copyright

vos rename

Name

`vos rename` — Renames a volume

Synopsis

vos rename -oldname <*old volume name*> -newname <*new volume name*> [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos ren -o <*old volume name*> -ne <*new volume name*> [-c <*cell name*>] [-no] [-l] [-v] [-h]

Description

The vos rename command changes the name of the read / write volume specified with the -oldname argument to the name specified with the -newname argument. The names of the read / write's read-only 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 the fs rmmount command and creating a new one with the fs mkmount command.

Options

-oldname <*old volume name*>

Is the current name of the read / write volume.

-newname <*new volume name*>

Is the desired new name for the volume.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The vos rename command produces no output if the command succeeds.

If the volume named by the -oldname argument does not exist, the following message appears:

```
vos: Could not find entry for volume &lt;old volume name&gt;.
```

Examples

The following example changes the mistaken volume name `sun4x_56.afsws` to the correct alternative `sun4x_56.usr.afsws`.

```
% vos rename -oldname sun4x_56.afsws -newname sun4x_56.usr.afsws
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289

Copyright

vos restore

Name

`vos restore` — Converts an ASCII dump file into an AFS volume

Synopsis

vos restore -server <*machine name*> -partition <*partition name*> -name <*name of volume to be restored*> [-file <*dump file*>] [-id <*volume ID*>] [-overwrite (abort | full | incremental)] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos res -s <*machine name*> -p <*partition name*> -na <*name of volume to be restored*> [-f <*dump file*>] [-i <*volume ID*>] [-o (a | f | i)] [-c <*cell name*>] [-no] [-l] [-v] [-h]

Description

The vos restore command converts a volume dump file previously created with the vos dump command from ASCII into the volume format appropriate for the machine type indicated by the -server argument, and restores it as a read / write volume to the partition named by the -partition argument on that machine. The Volume Server assigns the volume name indicated with the -name argument, and resets the volume's creation timestamp to the time at which the restore operation begins (the creation timestamp is stored in the volume header and reported in the `Creation` field in the output from the vos examine and vos listvol commands.)

Use the -file argument to name the dump file, or omit the argument to provide the file via the standard input stream, presumably through a pipe. The pipe can be named, which enables interoperation with third-party backup utilities.

As described in the following list, the command can create a completely new volume or overwrite an existing volume. In all cases, the full dump of the volume must be restored before any incremental dumps. If there are multiple incremental dump files, they must be restored in the order they were created.

- To create a new read / write volume, use the -name argument to specify a volume name that does not already exist in the Volume Location Database (VLDB), and the -server and -partition arguments to specify the new volume's site. It is best to omit the -id argument so that the Volume Location (VL) Server allocates a volume ID automatically. Do not include the -overwrite argument, because there is no existing volume to overwrite.
- To overwrite an existing volume at its current site, specify its name and site with the -name, -server, and -partition arguments. The volume retains its current volume ID number unless the -id argument is provided. Specify the value `f` or `i` for the -overwrite argument to indicate whether the dump file is full or incremental, respectively.
- To overwrite an existing volume and move it to a new site, specify its name and the new site with the -name, -server, and -partition arguments. The volume retains its current volume ID number unless the -id argument is provided. The volume is removed from its original site. Specify the value `f` for the -overwrite argument to indicate that the dump file is a full dump (it is not possible to restore an incremental dump and move the volume at the same time).

If the volume named by the -name argument already exists and the -overwrite argument is omitted, the command interpreter produces the following prompt:

```
Do you want to do a full/incremental restore or abort? [fia](a):
```

Respond by entering one of the following values:

- `f` if restoring a full dump file
- `i` if restoring an incremental dump file
- `a` or Return to cancel the restore operation

Cautions

If the -file argument is omitted, the issuer must provide all other necessary arguments, because the standard input stream is unavailable for responding to the command interpreter's prompts for missing information. In particular, the issuer must provide the -overwrite argument if overwriting an existing volume.

Options

-server <*server name*>

Identifies the file server machine onto which to restore the volume. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies the partition (on the file server machine specified by the -server argument) onto which to restore the volume. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-name <*name of volume*>

Specifies the name under which to restore the volume. It can be up to 22 characters long, but cannot end with a `.readonly` or `.backup` extension. If the volume already exists, it is overwritten subject to the value of the -overwrite argument.

-file <*dump file*>

Names the dump file to restore. Incomplete pathnames are interpreted relative to the current working directory. Omit this argument to provide the dump file via the standard input stream.

-id <*volume ID*>

Specifies the volume ID number to assign to the restored volume.

-overwrite (a | f | i)

Specifies which type of dump file is being restored when overwriting an existing volume. Provide one of the following values:

- `a` to terminate the restore operation.
- `f` if restoring a full dump file.
- `i` if restoring an incremental dump file. This value is not acceptable if the -server and -partition arguments do not indicate the volume's current site.

This argument is mandatory if the -file argument is not provided.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command restores the contents of the dump file */afs/abc.com/common/dumps/terry.dump* to the */vicepc* partition on the file server machine `fs3.abc.com`. The restored volume is named `user.terry`.

```
% cd /afs/abc.com/common/dumps
% vos restore -file terry.dump -server fs3.abc.com -partition c \
    -name user.terry
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_dump(1)" on page 319, "vos_examine(1)" on page 323, "vos_listvol(1)" on page 339

Copyright

vos status

Name

`vos status` — Reports a Volume Server's status

Synopsis

vos status -server <*machine name*> [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos st -s <*machine name*> [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos status command 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 following message appears:

```
No active transactions on &lt;machine_name&gt;
```

This command is useful mainly if there is concern that the Volume Server is not performing requested actions.

Options

-server <*server name*>

Identifies the file server machine running the Volume Server for which to display status information. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

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 I&lt;machine name&gt;
```

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 can 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 the vos status command is issued repeatedly during a restore, `readNext` increases 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 the vos status command is issued repeatedly during a dump, `transmitNext` increases 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 example illustrates the kind of output that sometimes appears when the Volume Server on `fs1.abc.com` is executing a dump at the time this command is issued.

```
% vos status fs1.abc.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

See Also

"vos(1)" on page 289

Copyright

vos syncserv

Name

`vos syncserv` — Verifies VLDB entries that mention a specified site

Synopsis

vos syncserv -server <*machine name*> [-partition <*partition name*>] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos syncs -s <*machine name*> [-p <*partition name*>] [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos syncserv command verifies that each volume mentioned in a VLDB entry actually exists at the site indicated in the entry. It checks all VLDB entries that mention a read / write, read-only, or backup site either on any partition on the file server machine specified by the -server argument, or on the one partition specified by the -server and -partition arguments. Note that the command can end up inspecting sites other than those specified by the -server and -partition arguments, if there are versions of the volume at sites other than the one specified.

The command alters any incorrect information in the VLDB, unless there is an irreconcilable conflict with other VLDB entries. In that case, it writes a message to the standard error stream instead. The command never removes volumes from file server machines.

To achieve complete VLDB consistency, first run the vos syncvldb command on all file server machines in the cell, then run this command on all file server machines in the cell.

Options

-server <*server name*>

Identifies the file server machine mentioned in each VLDB entry to check. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies the partition mentioned in each VLDB entry to check. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example verifies the VLDB entries in which a site definition mentions the file server machine `fs3.abc.com`.

```
% vos syncserv -server fs3.abc.com
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_syncvldb(1)" on page 379

Copyright

vos syncvldb

Name

vos syncvldb — Verifies VLDB entries for volumes residing at specified site

Synopsis

vos syncvldb [-server <*machine name*>] [-partition <*partition name*>] [-volume <*volume name or ID*>] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos syncv [-s <*machine name*>] [-p <*partition name*>] [-vo <*volume name or ID*>] [-c <*cell name*>] [-n] [-l] [-ve] [-h]

Description

The vos syncvldb command verifies that the status of the volumes housed either on all partitions on the file server machine specified by the -server argument, or on the single partition specified by the -server and -partition arguments, is recorded correctly in the VLDB. If the -volume argument is included to indicate a single volume, the command compares only its status on the file server machine with its VLDB entry.

If the -volume argument is not included, the command interpreter obtains from the Volume Server a list of the volumes that reside on each partition, then changes information in the VLDB as necessary to reflect their state on the partition. For example, it creates or updates a VLDB entry when it finds a volume for which the VLDB entry is missing or incomplete. However, if there is already a VLDB entry that defines a different location for the volume, or there are irreconcilable conflicts with other VLDB entries, it instead writes a message about the conflict to the standard error stream. The command never removes volumes from the file server machine.

To achieve complete VLDB consistency, run this command on all file server machines in the cell, and then run the vos syncserv command on all file server machines in the cell.

Using the -volume argument basically combines the effects of this command with those of the vos syncserv command, for a single volume. The command not only verifies that the VLDB entry is correct for the specified volume type (read / write, backup, or read-only), but also checks that any related volume types mentioned in the VLDB entry actually exist at the site listed in the entry. It is not necessary to provide the -server argument (and optionally, -partition argument); if one or both is provided, the results are reliable only if they specify the actual location of the volume indicated by the -volume argument.

Options

-server <*server name*>

Identifies the file server machine housing the volumes for which to verify VLDB entries. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies the partition housing the volumes for which to verify VLDB entries. Provide the -server argument along with this one. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-volume <*volume name or ID*>

Specifies the name or volume ID number of a single volume for which to verify the VLDB entry. This argument can be combined with the -server (and optionally, the -partition) argument.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example command verifies the VLDB entry for each volume stored on the file server machine `fs4.abc.com`.

```
% vos syncvldb fs4.abc.com
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_syncserv(1)" on page 377

Copyright

vos unlock

Name

`vos unlock` — Unlocks a single VLDB entry

Synopsis

vos unlock -id <*volume name or ID*> [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos unlock -i <*volume name or ID*> [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos unlock command releases the lock on the Volume Location Database (VLDB) entry for the indicated volume.

Cautions

Do not user this command 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 command unlocks several VLDB entries at once, or even the entire VLDB. The vos lock command locks a VLDB entry so that no one else can perform an action that requires writing the VLDB.

Options

-id <*volume name or ID*>

Specifies either the complete name or volume ID number of a volume of any of the three types.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The vos command interpreter presents it to the Volume Server and Volume Location Server during

mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example unlocks the VLDB entry for the volume `user.terry`.

```
% vos unlock user.terry
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_lock(1)" on page 345, "vos_unlockvldb(1)" on page 385

Copyright

vos unlockvldb

Name

`vos unlockvldb` — Unlocks several locked VLDB entries

Synopsis

vos unlockvldb [-server <*machine name*>] [-partition <*partition name*>] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos unlockv [-s <*machine name*>] [-p <*partition name*>] [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos unlockvldb command releases the lock on the Volume Location Database (VLDB) entries indicated by the combination of arguments provided:

- To unlock all entries in the VLDB, provide no arguments.
- To unlock all entries that mention a file server machine in a site definition, provide its name with the -server argument.
- To unlock all entries that mention a partition on any file server machine in a site definition, provide the partition name with the -partition argument.
- To unlock all entries that mention a specific site, provide both the -server and -partition arguments.

To unlock a single volume, use the vos unlock command instead.

Cautions

Do not use this command 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 can perform an action that requires writing the VLDB.

Options

-server <*server name*>

Identifies the file server machine for which to unlock VLDB entries. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies the partition (on the file server machine specified by the -server argument) for which to unlock VLDB entries. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command unlocks all locked entries in the VLDB.

```
% vos unlockvldb
```

The following command unlocks all locked VLDB entries that mention the */vicepa* partition in a site definition.

```
% vos unlockvldb -partition a
```

The following command unlocks all locked VLDB entries that refer to volumes on the */vicepc* partition of the file server machine `fs3.abc.com`.

```
% vos unlockvldb -server fs3.abc.com -partition c
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_lock(1)" on page 345, "vos_unlock(1)" on page 383

Copyright

vos zap

Name

vos zap — Removes a volume from its site without writing to the VLDB

Synopsis

vos zap -server <*machine name*> -partition <*partition name*> -id <*volume ID*> [-force] [-backup] [-cell <*cell name*>] [-noauth] [-localauth] [-verbose] [-help]

vos z -s <*machine name*> -p <*partition name*> -i <*volume ID*> [-f] [-b] [-c <*cell name*>] [-n] [-l] [-v] [-h]

Description

The vos zap command removes the volume with the specified *volume ID* from the site defined by the -server and -partition arguments, without attempting to change the corresponding Volume Location Database (VLDB) entry. If removing the volume can possibly 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 "CAUTIONS" on page .

To remove the specified read / write volume's backup version at the same time, include the -backup flag.

Cautions

Do not use this command 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 the vos remove command instead.

This command is useful in situations where it is important to delete the volume, but for some reason the VLDB is unreachable -- for example, because s the Volume Location Server is unavailable. The issuer can remove the VLDB entry later with the vos remove or vos delentry command, or it is removed automatically when the vos syncserv and vos syncvldb commands run.

To remove a read-only site defined in the VLDB by mistake, before a copy actually exists at the site, use the vos remsite command. To remove an entire VLDB entry without affecting volumes at their sites, use the vos delentry command.

Do not use the -force flag if the volume is online, but only when attempts to remove the volume with the vos remove or the vos zap command have failed, or the volume definitely cannot be attached. After using the -force flag, make sure that the volume's VLDB entry is also removed (issue the vos delentry command 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.

Options

-server <*server name*>

Identifies the file server machine from which to remove the volume. Provide the machine's IP address or its host name (either fully qualified or using an unambiguous abbreviation). For details, see "vos(1)" on page 289.

-partition <*partition name*>

Identifies the partition (on the file server machine specified by the -server argument) from which to remove the volume. Provide the partition's complete name with preceding slash (for example, `/vicepa`) or use one of the three acceptable abbreviated forms. For details, see "vos(1)" on page 289.

-id <*volume ID*>

Specifies the volume ID number of the volume to remove, which can be of any of the three types. The volume name is not acceptable.

-force

Removes the volume even though it cannot be attached (brought online). Use only after the failure of previous attempts to remove the volume by using the vos remove command or the vos zap command without this flag.

-backup

Removes the backup version of the read / write volume specified by the -id argument. Do not use this flag if the -id argument identifies a read-only or backup volume.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "vos(1)" on page 289.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "vos(1)" on page 289.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The vos command interpreter presents it to the Volume Server and Volume Location Server during mutual authentication. Do not combine this flag with the -cell argument or -noauth flag. For more details, see "vos(1)" on page 289.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example removes the volume with volume ID 536870988 from the */vicepf* partition of the file server machine `fs6.abc.com`, without noting the change in the VLDB.

```
% vos zap -server fs6.abc.com -partition f -id 536870988
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine specified with the -server argument and on each database server machine. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"vos(1)" on page 289, "vos_delentry(1)" on page 315, "vos_remove(1)" on page 359, "vos_remsite(1)" on page 363

Copyright

xstat_cm_test

Name

`xstat_cm_test` — Displays data collections from the Cache Manager

Synopsis

xstat_cm_test [*initcmd*] -cmname <*cache manager name(s) to monitor*>+ -collID <*collection(s) to fetch*>+ [-onceonly] [-frequency <*poll frequency, in seconds*>] [-period <*data collection time, in minutes*>] [-debug] [-help]

xstat_cm_test [*i*] -cm <*cache manager name(s) to monitor*>+ -co <*collection(s) to fetch*>+ [-o] [-f <*poll frequency, in seconds*>] [-p <*data collection time, in minutes*>] [-d] [-h]

Description

The xstat_cm_test command tests the routines in the *`libxstat_cm.a`* library and displays the data collections associated with the Cache Manager. The command executes in the foreground.

The command produces a large volume of output; to save it for later analysis, direct it to a file.

Options

initcmd

Accommodates the command's use of the AFS command parser, and is optional.

-cmname <*cache manager name to monitor*>+

Specifies the fully qualified hostname of each client machine for which to monitor the Cache Manager.

-collID <*collection to fetch*>+

Specifies each data collection to return, which defines the type and amount of data the command interpreter gathers about the Cache Manager. Data is returned in a predefined data structure.

There are three acceptable values:

0

Provides profiling information about the numbers of times different internal Cache Manager routines were called since the Cache Manager started.

1

Reports various internal performance statistics related to the Cache Manager (for example, statistics about how effectively the cache is being used and the quantity of intracell and intercell data access).

2

Reports all of the internal performance statistics provided by the `1` setting, plus some additional, detailed performance figures (for example, statistics about the number of RPCs sent by the Cache Manager and how long they take to complete, and statistics regarding authentication, access, and PAG information associated with data access).

-onceonly

Gathers statistics just one time. Omit this flag to have the command continue to probe the Cache Manager for statistics at the frequency specified by the -frequency argument; in this case press Ctrl-C to stop the probes.

-frequency <*poll frequency*>

Sets the frequency in seconds at which the program initiates probes to the Cache Manager. The default is 30 seconds.

-period <*data collection time*>

Sets the number of minutes the program runs; at the end of this period of time, the program exits. The default is 10 minutes.

-debug

Displays a trace on the standard output stream as the command runs.

-help

Prints the online help for this command. All other valid options are ignored.

See Also

"xstat_fs_test(1)" on page 395

Copyright

xstat_fs_test

Name

`xstat_fs_test` — Displays data collections from the File Server process

Synopsis

xstat_fs_test [*initcmd*] -fsname <*file server name(s) to monitor*>+ -collID <*collection(s) to fetch*>+ [-onceonly] [-frequency <*poll frequency, in seconds*>] [-period <*data collection time, in minutes*>] [-debug] [-help]

xstat_fs_test [initcmd] -fs <*File Server name(s) to monitor*>+ -c <*Collection(s) to fetch*>+ [-o] [-fr <*poll frequency, in seconds*>] [-p <*data collection time, in minutes*>] [-d] [-h]

Description

The xstat_fs_test command tests the routines in the *`libxstat_fs.a`* library and displays the data collections associated with the File Server (the `fs` process). The command executes in the foreground.

The command produces a large volume of output; to save it for later analysis, direct it to a file.

Options

initcmd

Accommodates the command's use of the AFS command parser, and is optional.

-fsname <*file server name to monitor*>+

Specifies the fully qualified hostname of each file server machine for which to monitor the File Server process.

-collID <*collection to fetch*>+

Specifies each data collection to return, which defines the type and amount of data the command interpreter gathers about the File Server. Data is returned in a predefined data structure.

There are three acceptable values:

0

Provides profiling information about the numbers of times different internal File Server routines were called since the File Server started. This value is not currently implemented; it returns no data.

1

Reports various internal performance statistics related to the File Server (for example, vnode cache entries and Rx protocol activity).

2

Reports all of the internal performance statistics provided by the `1` setting, plus some additional, detailed performance figures about the File Server (for example, minimum, maximum, and cumulative statistics regarding File Server RPCs, how long they take to complete, and how many succeed).

-onceonly

Gathers statistics just one time. Omit this flag to have the command continue to probe the Cache Manager for statistics at the frequency specified by the -frequency argument; in this case press Ctrl-C to stop the probes.

-frequency <*poll frequency*>

Sets the frequency in seconds at which the program initiates probes to the Cache Manager. The default is 30 seconds.

-period <*data collection time*>

Sets the number of minutes the program runs; at the end of this period of time, the program exits. The default is 10 minutes.

-debug

Displays a trace on the standard output stream as the command runs.

-help

Prints the online help for this command. All other valid options are ignored.

See Also

"xstat_cm_test(1)" on page 393

Copyright

II. AFS System Files

afs

Name

`afs` — Introduction to AFS files

Description

A number of files must reside on the local disk of AFS server and client machines. They belong to the following general categories:

- *Configuration files* define configuration parameters for specific server and kernel processes such as the Backup System Tape Coordinator or the Cache Manager.
- *Administrative files* list information used in administration of server machines, such as a list of privileged users or server encryption keys.
- *Cache-related files* contain cached data or information about cached data, on client machines.
- *Log files* contain tracing messages about the operation of a specific process.
- *Database files* contain database records used to administer the AFS cell.
- *Controller files* control the behavior of a process.
- *Volume header files* represent AFS volumes on server partitions.

For a description of the format and contents of each file, see its reference page.

Note for Windows users: Some files described in this document possibly do not exist on machines that run a Windows operating system. Also, Windows uses a backslash (\) rather than a forward slash (/) to separate the elements in a pathname.

See Also

Configuration files:

"BosConfig(5)" on page 409

"CellServDB(5)" on page 415

"NetInfo(5)" on page 425

"NetRestrict(5)" on page 427

"ThisCell(5)" on page 435

Administrative files:

Cache-related files:

Log files:

Database files:

Controller files:

"salvage.lock(5)" on page 489

Volume header files:

"afs_volume_header(5)" on page 445

Copyright

AuthLog.dir, AuthLog.pag

Name

`AuthLog.dir, AuthLog.pag` — Log of Authentication Server privileged operations

Description

The *`AuthLog.dir`* and *`AuthLog.pag`* files record a trace of privileged operations performed by the Authentication Server (kaserver process) on the local machine. If the files do not exist when the Authentication Server starts, it creates them in the */usr/afs/logs* directory as necessary.

The files are in binary format. To display their contents, use the kdb command, which requires being logged in to the local machine as the local superuser `root`.

Cautions

The Authentication Server is possibly unable to create these files on some operating systems that AFS otherwise supports, making the kdb command inoperative. See the *IBM AFS Release Notes* for details.

See Also

"kaserver(8)" on page 859, "kdb(8)" on page 863

Copyright

AuthLog.dir, AuthLog.pag

AuthLog

Name

AuthLog — Traces Authentication Server operations

Description

The *AuthLog* file records a trace of Authentication Server (kaserver process) operations on the local machine and describes any error conditions it encounters.

If the *AuthLog* file does not exist in the */usr/afs/logs* directory when the Authentication Server starts, the server process creates it and writes initial start-up messages to it. If there is an existing file, the Authentication Server renames it to *AuthLog.old*, overwriting the existing *AuthLog.old* file if it exists.

The file is in ASCII format. Administrators listed in the */usr/afs/etc/UserList* file can use the bos getlog command to display its contents. Alternatively, log onto the server machine and use a text editor or a file display command such as the UNIX cat command. By default, the mode bits on the *AuthLog* file grant the required r (read) permission to all users.

The Authentication Server records operations only as it completes them, and cannot recover from failures by reviewing the file. The log contents are useful for administrative evaluation of process failures and other problems.

See Also

"UserList(5)" on page 437, "bos_getlog(8)" on page 691, "kaserver(8)" on page 859

Copyright

BackupLog

Name

BackupLog — Traces Backup Server operations

Description

The *BackupLog* file records a trace of Backup Server (buserver process) operations on the local machine and describes any error conditions it encounters.

If the *BackupLog* file does not already exist in the */usr/afs/logs* directory when the Backup Server starts, the server process creates it and writes initial start-up messages to it. If there is an existing file, the Backup Server renames it to *BackupLog.old*, overwriting the existing *BackupLog.old* file if it exists.

The file is in ASCII format. Administrators listed in the */usr/afs/etc/UserList* file can use the bos getlog command to display its contents. Alternatively, log on to the machine and use a text editor or a file display command such as the UNIX cat command. By default, the mode bits on the *BackupLog* file grant the required r (read) permission to all users.

The Backup Server records operations only as it completes them, and so cannot recover from failures by reviewing the file. The log contents are useful for administrative evaluation of process failures and other problems.

See Also

"UserList(5)" on page 437, "bos_getlog(8)" on page 691, "buserver(8)" on page 763

Copyright

BosConfig

Name

`BosConfig` — Defines server processes for the BOS Server to monitor

Description

The *`BosConfig`* file lists the processes that the Basic OverSeer (BOS) Server monitors on its server machine, and thus defines which AFS server processes run on the machine. It specifies how the BOS Server reacts when a process fails, and also defines the times at which the BOS Server automatically restarts processes as part of performance maintenance. The file must reside in the */usr/afs/local* directory on each AFS server machine.

A server process entry in the *`BosConfig`* file records the following information:

- The *entry type*, which is one of the following:

  cron

  Designates a server process that runs periodically instead of continuously. The BOS Server starts a cron process only at specified times, not whenever it fails. All standard AFS process entries except `fs` are simple (there are no standard cron processes).

  fs

  Designates a group of interdependent server processes. If one of the processes fails, the BOS Server must coordinate its restart with the restart of the other processes in the group, possibly by stopping them first.

  There is only one standard entry of this type, for which the conventional name is `fs`. It combines three server processes: the File Server (fileserver process), the Volume Server (volserver process), and the Salvager (salvager process). These processes all operate on the same data--the AFS data stored on an AFS server machine's */vicep* partitions and mounted in the AFS filespace--but in different ways. Grouping the processes prevents them from attempting to access the same data simultaneously, which can cause corruption.

  During normal operation, the Salvager process is not active. If the File Server process fails, however, the BOS Server stops the Volume Server process and runs the Salvager process to correct any corruption that resulted from the failure. (The administrator can also issue the bos salvage command to invoke the Salvager process.) If the Volume Server fails, the BOS Server can restart it without stopping the File Server or running the Salvager.

  simple

  Designates a server process that runs independently of any other on the server machine. If a simple process fails, the BOS Server does not have to coordinate its restart with any other process.

- The *entry name*. The conventional name for an entry in the `BosConfig` file and the associated process matches the binary filename. When issuing any bos command that takes the -instance argument, identify each process by the name used in the `BosConfig` file. For a list of the names, see the bos create reference page.
- The process's *status flag*, which determines whether the BOS Server attempts to start the process in two cases: each time the BOS Server itself restarts, and when the process fails. The `BosConfig` file currently uses a binary notation to indicate whether the BOS Server attempts to restart the process as necessary or does not monitor it at all. For the sake of clarity, the AFS documentation refers to the flags as `Run` and `NotRun` instead. Only a system administrator, not the BOS Server, can change the flag.
- One or more *command parameters* which the BOS Server invokes to start the process or processes associated with the entry:
  - A `cron` entry has two command parameters, the first the complete pathname to the program, and the second the time at which the BOS Server invokes the program.
  - The `fs` entry has three command parameters, each the complete pathname to the fileserver, volserver, and salvager programs, in that order.
  - A `simple` entry has only one command parameter, the complete pathname to the program.

In addition to server process entries, the `BosConfig` file specifies the times at which the BOS Server performs two types of automatic process restarts:

- The *general restart* time at which the BOS Server restarts itself and then each process for which the entry in the `BosConfig` file has status flag `Run`. The default setting is Sunday at 4:00 a.m.
- The *binary restart* time at which the BOS Server restarts any server process for which the time stamp on the binary file in the `/usr/afs/bin` directory is later than the last restart time for the process. The default is 5:00 a.m.

Although the `BosConfig` file is in ASCII format, do not use a text editor to alter it. Its format is subject to change and incorrectly formatted entries can prevent server startup in ways that are difficult to diagnose. Instead always use the appropriate commands from the bos command suite:

- The bos create command to create an entry in the file and start the associated process.
- The bos delete command to remove an entry from the file after the bos stop command is used to stop the associated process.
- The bos getrestart command to display the times at which the BOS Server performs automatic restarts.
- The bos setrestart command to set the times at which the BOS Server performs automatic process restarts.
- The bos start command to change an entry's status flag to `Run` and start the associated process.
- The bos status command to display all processes listed in the file.

- The bos stop command to change an entry's status flag to `NotRun` and stop the associated process.

There are also bos commands that start and stop processes without changing entries in the *`BosConfig`* file. The BOS Server reads the *`BosConfig`* file only when it starts, transferring the information into its memory. Thus a process's status as represented in the BOS Server's memory can diverge from its status in the *`BosConfig`* file. The following commands change a process's status in the BOS Server's memory only:

- The bos restart command restarts a specified set of processes, all processes, or all processes other than the BOS Server.
- The bos shutdown command stops a process.
- The bos startup command starts a process.

See Also

"bos_create(8)" on page 677, "bos_delete(8)" on page 685, "bos_getrestart(8)" on page 695, "bos_restart(8)" on page 723, "bos_setrestart(8)" on page 739, "bos_shutdown(8)" on page 743, "bos_start(8)" on page 747, "bos_startup(8)" on page 749, "bos_status(8)" on page 751, "bos_stop(8)" on page 755, "bos_salvage(8)" on page 727, "fileserver(8)" on page 771, "salvager(8)" on page 879, "volserver(8)" on page 925

Copyright

BosLog

Name

BosLog — Traces BOS Server operations

Description

The *BosLog* file records a trace of Basic OverSeer (BOS) Server (bosserver process) operations on the local machine and describes any error conditions it encounters.

If the *BosLog* file does not already exist in the */usr/afs/logs* directory when the BOS Server starts, the server process creates it and writes initial start-up messages to it. If there is an existing file, the BOS server renames it to *BosLog.old*, overwriting the existing *BosLog.old* file if it exists.

The file is in ASCII format. Administrators listed in the */usr/afs/etc/UserList* file can use the bos getlog command to display its contents. Alternatively, log onto the server machine and use a text editor or a file display command such as the UNIX cat command. By default, the mode bits on the *BosLog* file grant the required r (read) permission to all users.

The BOS Server records operations only as it completes them, and cannot recover from failures by reviewing the file. The log contents are useful for administrative evaluation of process failures and other problems.

See Also

"UserList(5)" on page 437, "bos_getlog(8)" on page 691, "bosserver(8)" on page 759

Copyright

CellServDB

Name

`CellServDB` — Lists the database server machines in AFS cells

Description

There are two versions of the *`CellServDB`* file, both of which have the same format. One version is used by an AFS client and lists all of the database server machines in the local cell and any foreign cell that is to be accessible from the local client machine. The other version is used on servers and lists only the database servers in the local cell.

Client CellServDB

The client version of the CellServDB file lists the database server machines in the local cell and any foreign cell that is to be accessible from the local client machine. Database server machines run the Authentication Server, Backup Server, Protection Server, and Volume Location (VL) Server (the kaserver, buserver, ptserver, and vlserver) processes, which maintain the cell's administrative AFS databases.

The Cache Manager and other processes running on a client machine use the list of a cell's database server machines when performing several common functions, including:

- Fetching files. The Cache Manager contacts the VL Server to learn the location of the volume containing a requested file or directory.
- Authenticating users. Client-side authentication programs (such as an AFS-modified login utility or the klog command interpreter) contact the Authentication Server to obtain a server ticket, which the AFS server processes accept as proof that the user is authenticated.
- Creating protection groups. The pts command interpreter contacts the Protection Server when users create protection groups or request information from the Protection Database.

The Cache Manager reads the CellServDB file into kernel memory as it initializes, and not again until the machine next reboots. To enable users on the local machine to continue accessing the cell correctly, update the file whenever a database server machine is added to or removed from a cell. To update the kernel-resident list of database server machines without rebooting, use the fs newcell command.

The *`CellServDB`* file is in ASCII format and must reside in the */usr/vice/etc* directory on each AFS client machine. Use a text editor to create and maintain it.

The client version of the *`CellServDB`* file is distinct from the server version, which resides in the */usr/afs/etc* directory on each AFS server machine. The client version lists the database server machines in every AFS cell that the cell administrator wants the machine's users to be able to access, whereas the server version lists only the local cell's database server machines.

Server CellServDB

The server version of the `CellServDB` file lists the local cell's database server machines. These machines run the Authentication Server, Backup Server, Protection Server, and Volume Location (VL) Server (the kaserver, buserver, ptserver, and vlserver) processes, which maintain the cell's administrative AFS databases. The initial version of the file is created with the bos setcellname command during the installation of the cell's server machine, which is automatically recorded as the cell's first database server machine. When adding or removing database server machines, be sure to update this file appropriately. It must reside in the `/usr/afs/etc` directory on each AFS server machine.

The database server processes consult the `CellServDB` file to learn about their peers, with which they must maintain constant connections in order to coordinate replication of changes across the multiple copies of each database. The other AFS server processes consult the file to learn which machines to contact for information from the databases when they need it.

Although the server `CellServDB` file is in ASCII format, do not use a text editor to alter it. Instead always use the appropriate commands from the bos command suite:

- The bos addhost command to add a machine to the file.
- The bos listhosts command to display the list of machines from the file.
- The bos removehost command to remove a machine from the file.

In cells that use the Update Server to distribute the contents of the `/usr/afs/etc` directory, it is customary to edit only the copy of the file stored on the system control machine. Otherwise, edit the file on each server machine individually. For instructions on adding and removing database server machine, see the *IBM AFS Quick Beginnings* chapter on installing additional server machines.

CellServDB Format

Both `CellServDB` files have the same format:

- The first line begins at the left margin with the greater-than character (>), followed immediately by the cell's name without an intervening space. Optionally, a comment can follow any number of spaces and a number sign (#), perhaps to identify the organization associated with the cell.
- Each subsequent line in the entry identifies one of the cell's database server machines, with the indicated information in order:
  - The database server machine's IP address in dotted-decimal format.
  - One or more spaces.
  - A number sign (#), followed by the machine's fully qualified hostname without an intervening space. This number sign does not indicate that the hostname is a comment. It is a required field.

No extra blank lines or newline characters are allowed in the file, even after the last entry. Their presence can prevent the Cache Manager from reading the file into kernel memory, resulting in an error message.

grand.central.org maintains a list of the database server machines in all cells that have registered themselves as receptive to access from foreign cells. When a cell's administrators change its database server machines, it is customary to register the change with grand.central.org for inclusion in this file. The file conforms to the required `CellServDB` format, and so is a suitable basis for the `CellServDB` file on a client machine. You can download this file from http://grand.central.org/.

Examples

The following example shows entries for two cells in a client `CellServDB` file and illustrates the required format.

```
&gt;abc.com        # ABC Corporation
192.12.105.2        #db1.abc.com
192.12.105.3        #db2.abc.com
192.12.107.3        #db3.abc.com
&gt;test.abc.com   # ABC Corporation Test Cell
192.12.108.57       #testdb1.abc.com
192.12.108.55       #testdb2.abc.com
```

See Also

"bos_addhost(8)" on page 665, "bos_listhosts(8)" on page 705, "bos_removehost(8)" on page 717, "bos_setcellname(8)" on page 735, "buserver(8)" on page 763, "fs_newcell(1)" on page 109, "kaserver(8)" on page 859, "klog(1)" on page 153, "ptserver(8)" on page 875, "vlserver(8)" on page 915, "upclient(8)" on page 885, "upserver(8)" on page 889

IBM AFS Quick Beginnings

Copyright

FORCESALVAGE

Name

FORCESALVAGE — Forces salvage of entire partition

Description

The *FORCESALVAGE* file, if present on an AFS server partition (that is, in a */vicep* directory), signals that the Salvager must salvage the entire partition. The AFS-modified version of the fsck program creates the empty (zero-length) file when it discovers corruption on the partition. The Salvager removes the file when it completes the salvage operation.

When the File Server detects the presence of the file on a partition on which it is attaching volumes, it stops, detaches any volumes that are already attached, and exits after recording a message in the */usr/afs/logs/FileLog* file. The Bos Server then invokes the Salvager to salvage the partition.

See Also

"FileLog(5)" on page 421, "bosserver(8)" on page 759, "fileserver(8)" on page 771, "salvager(8)" on page 879

Copyright

FileLog

Name

`FileLog` — Traces File Server operations

Description

The *`FileLog`* file records a trace of File Server (fileserver process) operations on the local machine and describes any error conditions it encounters.

If the *`FileLog`* file does not already exist in the */usr/afs/logs* directory when the File Server starts, the server process creates it and writes initial start-up messages to it. If there is an existing file, the File Server renames it to *`FileLog.old`*, overwriting the existing *`FileLog.old`* file if it exists.

The file is in ASCII format. Administrators listed in the */usr/afs/etc/UserList* file can use the bos getlog command to display its contents. Alternatively, log onto the file server machine and use a text editor or a file display command such as the UNIX cat command. By default, the mode bits on the *`FileLog`* file grant the required `r` (read) permission to all users.

The File Server records operations only as it completes them, and cannot recover from failures by reviewing the file. The log contents are useful for administrative evaluation of process failures and other problems.

See Also

"UserList(5)" on page 437, "bos_getlog(8)" on page 691, "fileserver(8)" on page 771

Copyright

KeyFile

Name

KeyFile — Defines AFS server encryption keys

Description

The *KeyFile* file defines the server encryption keys that the AFS server processes running on the machine use to decrypt the tickets presented by clients during the mutual authentication process. AFS server processes perform privileged actions only for clients that possess a ticket encrypted with one of the keys from the file. The file must reside in the */usr/afs/etc* directory on every server machine. For more detailed information on mutual authentication and server encryption keys, see the *IBM AFS Administration Guide*.

Each key has a corresponding a key version number that distinguishes it from the other keys. The tickets that clients present are also marked with a key version number to tell the server process which key to use to decrypt it. The *KeyFile* file must always include a key with the same key version number and contents as the key currently listed for the afs entry in the Authentication Database.

The *KeyFile* file is in binary format, so always use the appropriate commands from the bos command suite to administer it:

- The bos addkey command to define a new key.
- The bos listkeys command to display the keys.
- The bos removekey command to remove a key from the file.

In cells that use the Update Server to distribute the contents of the */usr/afs/etc* directory, it is customary to edit only the copy of the file stored on the system control machine. Otherwise, edit the file on each server machine individually.

See Also

"bos_addkey(8)" on page 669, "bos_listkeys(8)" on page 707, "bos_removekey(8)" on page 719, "kas_setpassword(8)" on page 847, "upclient(8)" on page 885, "upserver(8)" on page 889

IBM AFS Administration Guide

Copyright

NetInfo

Name

`NetInfo` — Defines machine interfaces to register with AFS servers

Description

There are two *`NetInfo`* files, one for an AFS client and one for an AFS File Server or database server. The AFS client *`NetInfo`* file specifies the IP addresses that the client should register with the File Servers it connects to. The server *`NetInfo`* file specifies what interfaces should be registered with AFS Database Servers or used to talk to other database servers.

Client NetInfo

The client *`NetInfo`* file lists the IP addresses of one or more of the local machine's network interfaces. If it exists in the */usr/vice/etc* directory when the Cache Manager initializes, the Cache Manager uses its contents as the basis for a list of local interfaces. Otherwise, the Cache Manager uses the list of interfaces configured with the operating system. It then removes from the list any addresses that appear in the */usr/vice/etc/NetRestrict* file, if it exists. The Cache Manager records the resulting list in kernel memory. The first time it establishes a connection to a File Server, it registers the list with the File Server.

The File Server uses the addresses when it initiates a remote procedure call (RPC) to the Cache Manager (as opposed to responding to an RPC sent by the Cache Manager). There are two common circumstances in which the File Server initiates RPCs: when it breaks callbacks and when it pings the client machine to verify that the Cache Manager is still accessible.

The *`NetInfo`* file is in ASCII format. One of the machine's IP addresses appears on each line, in dotted decimal format. The File Server initially uses the address that appears first in the list. The order of the remaining addresses is not significant: if an RPC to the first interface fails, the File Server simultaneously sends RPCs to all of the other interfaces in the list. Whichever interface replies first is the one to which the File Server then sends pings and RPCs to break callbacks.

To prohibit the Cache Manager absolutely from using one or more addresses, list them in the *`NetRestrict`* file. To display the addresses the Cache Manager is currently registering with File Servers, use the fs getclientaddrs command. To replace the current list of interfaces with a new one between reboots of the client machine, use the fs setclientaddrs command.

Server NetInfo

The server *`NetInfo`* file, if present in the */usr/afs/local* directory, defines the following:

- On a file server machine, the local interfaces that the File Server (fileserver process) can register in the Volume Location Database (VLDB) at initialization time.
- On a database server machine, the local interfaces that the Ubik database synchronization library uses when communicating with the database server processes running on other database server machines.

If the *NetInfo* file exists when the File Server initializes, the File Server uses its contents as the basis for a list of interfaces to register in the VLDB. Otherwise, it uses the list of network interfaces configured with the operating system. It then removes from the list any addresses that appear in the */usr/afs/local/NetRestrict* file, if it exists. The File Server records the resulting list in the */usr/afs/local/sysid* file and registers the interfaces in the VLDB. The database server processes use a similar procedure when initializing, to determine which interfaces to use for communication with the peer processes on other database machines in the cell.

The *NetInfo* file is in ASCII format. One of the machine's IP addresses appears on each line, in dotted decimal format. The order of the addresses is not significant.

To display the File Server interface addresses registered in the VLDB, use the vos listaddrs command.

See Also

"sysid(5)" on page 491, "vldb.DB0(5)" on page 519, "fileserver(8)" on page 771, "fs_getclientaddrs(1)" on page 83, "fs_setclientaddrs(1)" on page 127, "vos_listaddrs(1)" on page 331

Copyright

NetRestrict

Name

NetRestrict — Defines interfaces not to register with AFS servers

Description

There are two *NetRestrict* files, one for an AFS client and one for an AFS File Server or database server. The AFS client *NetRestrict* file specifies the IP addresses that the client should not register with the File Servers it connects to. The server *NetInfo* file specifies what interfaces should not be registered with AFS Database Servers or used to talk to other database servers.

Client NetRestrict

The *NetRestrict* file, if present in a client machine's */usr/vice/etc* directory, defines the IP addresses of the interfaces that the local Cache Manager does not register with a File Server when first establishing a connection to it. For an explanation of how the File Server uses the registered interfaces, see "NetInfo(5)" on page 425.

As it initializes, the Cache Manager constructs a list of interfaces to register, from the */usr/vice/etc/NetInfo* file if it exists, or from the list of interfaces configured with the operating system otherwise. The Cache Manager then removes from the list any addresses that appear in the *NetRestrict* file, if it exists. The Cache Manager records the resulting list in kernel memory.

The *NetRestrict* file is in ASCII format. One IP address appears on each line, in dotted decimal format. The order of the addresses is not significant. The value 255 is a wildcard that represents all possible addresses in that field. For example, the value 192.12.105.255 indicates that the Cache Manager does not register any of the addresses in the 192.12.105 subnet.

To display the addresses the Cache Manager is currently registering with File Servers, use the fs getclientaddrs command.

Server NetRestrict

The *NetRestrict* file, if present in the */usr/afs/local* directory, defines the following:

- On a file server machine, the local interfaces that the File Server (fileserver process) does not register in the Volume Location Database (VLDB) at initialization time.
- On a database server machine, the local interfaces that the Ubik synchronization library does not use when communicating with the database server processes running on other database server machines.

As it initializes, the File Server constructs a list of interfaces to register, from the */usr/afs/local/NetInfo* file if it exists, or from the list of interfaces configured with the operating system otherwise. The File Server then removes from the list any addresses that appear in the *NetRestrict* file, if it exists. The File Server records the resulting list in the

/usr/afs/local/sysid file and registers the interfaces in the VLDB. The database server processes use a similar procedure when initializing, to determine which interfaces to use for communication with the peer processes on other database machines in the cell.

The *NetRestrict* file is in ASCII format. One IP address appears on each line, in dotted decimal format. The order of the addresses is not significant. The value `255` is a wildcard that represents all possible addresses in that field. For example, the value `192.12.105.255` indicates that the File Server or database server processes do not register or use any of the addresses in the `192.12.105` subnet.

To display the File Server interface addresses registered in the VLDB, use the vos listaddrs command.

See Also

"sysid(5)" on page 491, "vldb.DB0(5)" on page 519, "fileserver(8)" on page 771, "fs_getclientaddrs(1)" on page 83 "vos_listaddrs(1)" on page 331

Copyright

NoAuth

Name

NoAuth — Disables authorization checking

Description

The *NoAuth* file, if present in a server machine's */usr/afs/local* directory, indicates to the AFS server processes running on the machine that it is not necessary to perform authorization checking. They perform any action for any user who logs into the machine's local file system or issues a remote command that affects the machine's AFS server functioning, such as commands from the AFS command suites. Because failure to check authorization exposes the machine's AFS server functionality to attack, there are normally only two circumstances in which the file is present:

- During installation of the machine, as instructed in the *IBM AFS Quick Beginnings*.
- During correction of a server encryption key emergency, as discussed in the *IBM AFS Administration Guide*.

In all other circumstances, the absence of the file means that the AFS server processes perform authorization checking, verifying that the issuer of a command has the required privilege.

Create the file in one of the following ways:

- By issuing the bosserver initialization command with the -noauth flag, if the Basic OverSeer (BOS) Server is not already running.
- By issuing the bos setauth command with off as the value for the -authrequired argument, if the BOS Server is already running.

To remove the file, issue the bos setauth command with on as the value for the -authrequired argument.

The file's contents, if any, are ignored; an empty (zero-length) file is effective.

See Also

"bos_setauth(8)" on page 733, "bosserver(8)" on page 759

Copyright

SALVAGE.fs

Name

SALVAGE.fs — Triggers salvaging of AFS server partitions

Description

The *SALVAGE.fs* file, if present in a file server machine's */usr/afs/local* directory, indicates to the Basic OverSeer (BOS) Server (bosserver process) that it must invoke the Salvager (salvager process) during recovery from a failure of the File Server (fileserver process).

The BOS Server creates the zero-length file each time it starts or restarts the **fs** process. When the File Server exits normally (for example, in response to the bos shutdown or bos stop command), the BOS Server removes the file. However, if the File Server exits unexpectedly, the file remains in the */usr/afs/local* directory as a signal that the BOS Server must invoke the Salvager process to repair any file system inconsistencies possibly introduced during the failure, before restarting the File Server and Volume Server processes.

Do not create or remove this file. To invoke the Salvager process directly, use the bos salvage command or log onto the file server machine as the local superuser **root** and issue the salvager command.

See Also

"bos_salvage(8)" on page 727, "bosserver(8)" on page 759, "fileserver(8)" on page 771, "salvager(8)" on page 879, "volserver(8)" on page 925

Copyright

SalvageLog

Name

SalvageLog — Traces Salvager operations

Description

The *SalvageLog* file records a trace of Salvager (salvager process) operations on the local machine and describes any error conditions it encounters.

If the *SalvageLog* file does not already exist in the */usr/afs/logs* directory when the Salvager starts, the process creates it and writes initial start-up messages to it. If there is an existing file, the Salvager renames is to *SalvageLog.old*, overwriting the existing *SalvageLog.old* file if it exists.

The file is in ASCII format. Administrators listed in the */usr/afs/etc/UserList* file can use the bos getlog command to display its contents. Alternatively, log onto the file server machine and use a text editor or a file display command such as the UNIX cat command. By default, the mode bits on the *SalvageLog* file grant the required r (read) permission to all users.

The Salvager records operations only as it completes them, and cannot recover from failures by reviewing the file. The log contents are useful for administrative evaluation of process failures and other problems.

See Also

"UserList(5)" on page 437, "bos_getlog(8)" on page 691, "salvager(8)" on page 879

Copyright

ThisCell

Name

`ThisCell` — Defines the local cell name

Description

The *`ThisCell`* file defines the local cell name. There are two versions of this file, one for a AFS client and one for an AFS server.

Client ThisCell

The client version of the *`ThisCell`* file defines the complete Internet domain-style name (for example, `abc.com`) of the cell to which the local client machine belongs. It must reside in the */usr/vice/etc* directory on every AFS client machine. To change a client machine's cell membership, edit the file and reboot the machine.

The file is in ASCII format and contains a character string on a single line. The *IBM AFS Quick Beginnings* instructs the administrator to create it during the installation of each client machine.

The client machine's cell membership determines three defaults important to its functioning:

- The cell in which the machine's users authenticate by default. The effect is two-fold:
  - The AFS-modified login utilities and the klog command interpreter contact an Authentication Server in the cell named in the *`ThisCell`* file (unless -cell argument to the klog command specifies an alternate cell).
  - The command interpreters combine the cell name with the password that the user provides, generating an encryption key from the combination. For authentication to succeed, both the cell name and password must match the ones used to generate the user's encryption key stored in the Authentication Database.

- The cell the Cache Manager considers its local, or home, cell. By default, the Cache Manager allows programs that reside in its home cell to run with setuid permission, but not programs from foreign cells. For more details, see the fs getcellstatus and fs setcell reference pages.
- Which AFS server processes the local AFS command interpreters contact by default as they execute commands issued on the machine.

The client version of the *`ThisCell`* file is distinct from the server version, which resides in the */usr/afs/etc* directory on each AFS server machine. If a server machine also runs as a client, it is acceptable for the server and client versions of the file on the same machine to name different cells. However, the behavior that results from this configuration can be more confusing than useful.

Server ThisCell

The server version of the *ThisCell* file defines the complete Internet domain-style name (for example, abc.com) of the cell to which the server machine belongs. It must reside in the */usr/afs/etc* directory on every AFS server machine.

The file is in ASCII format and contains a character string on a single line. The initial version of the file is created with the bos setcellname command during the installation of the cell's first file server machine, and the *IBM AFS Quick Beginnings* includes instructions for copying it over to additional server machine during their installation.

The only reason to edit the file is as part of changing the cell's name, which is strongly discouraged because of the large number of configuration changes involved. In particular, changing the cell name requires rebuilding the entire Authentication Database, because the Authentication Server combines the cell name it finds in this file with each user and server password and converts the combination into an encryption key before recording it in the Database.

See Also

"bos_setcellname(8)" on page 735, "fs_getcellstatus(1)" on page 81, "fs_setcell(1)" on page 125

Copyright

UserList

Name

`UserList` — Defines privileged administrators

Description

The *`UserList`* file lists the AFS usernames of the system administrators authorized to issue privileged bos, vos, and backup commands that affect the local server machine or the volumes housed on it. It must reside in the */usr/afs/etc* directory on every server machine.

Although the *`UserList`* file is in ASCII format, do not use a text editor to alter it. Instead always use the appropriate commands from the bos command suite:

- The bos adduser command to add a user to the file.
- The bos listusers command to display the contents of the file.
- The bos removeuser command to remove a user from the file.

Although it is theoretically possible to list different administrators in the *`UserList`* files on different server machines, doing so can cause unanticipated authorization failures and is not recommended. In cells that use the Update Server to distribute the contents of the */usr/afs/etc* directory, it is customary to edit only the copy of the file stored on the system control machine. Otherwise, edit the file on each server machine individually.

See Also

"bos_adduser(8)" on page 673, "bos_listusers(8)" on page 711, "bos_removeuser(8)" on page 721, "upclient(8)" on page 885, "upserver(8)" on page 889

Copyright

VLLog

Name

VLLog — Traces Volume Location Server operations

Description

The *VLLog* file records a trace of Volume Location (VL) Server (vlserver process) operations on the local machine and describes any error conditions it encounters.

If the *VLLog* file does not already exist in the */usr/afs/logs* directory when the VL Server starts, the server process creates it and writes initial start-up messages to it. If there is an existing file, the VL Server renames it to *VLLog.old*, overwriting the existing *VLLog.old* file if it exists.

The file is in ASCII format. Administrators listed in the */usr/afs/etc/UserList* file can use the bos getlog command to display its contents. Alternatively, log onto the server machine and use a text editor or a file display command such as the UNIX cat command. By default, the mode bits on the *VLLog* file grant the required `r` (read) permission to all users.

The VL Server records operations only as it completes them, and cannot recover from failures by reviewing the file. The log contents are useful for administrative evaluation of process failures and other problems.

The VL Server can record messages at three levels of detail. By default, it records only very rudimentary messages. To increase logging to the first level of detail, issue the following command while logged onto the database server machine as the local superuser `root`.

```
# kill -TSTP &lt;vlserver_pid&gt;
```

where <vlserver_pid> is the process ID of the vlserver process, as reported in the output from the standard UNIX ps command. To increase to the second and third levels of detail, repeat the command.

To disable logging, issue the following command.

```
# kill -HUP &lt;vlserver_pid&gt;
```

To decrease the level of logging, first completely disable it and then issue the `kill -TSTP` command as many times as necessary to reach the desired level.

See Also

"UserList(5)" on page 437, "bos_getlog(8)" on page 691, "vlserver(8)" on page 915

Copyright

VolserLog

Name

VolserLog — Traces Volume Server operations

Description

The *VolserLog* file records a trace of Volume Server (volserver process) operations on the local machine and describes any error conditions it encounters.

If the VolserLog file does not already exist in the */usr/afs/logs* directory when the Volume Server starts, the server process creates it and writes initial start-up messages to it. If there is an existing file, the Volume Server renames it to *VolserLog.old*, overwriting the existing *VolserLog.old* file if it exists.

The file is in ASCII format. Administrators listed in the */usr/afs/etc/UserList* file can use the bos getlog command to display its contents. Alternatively, log onto the file server machine and use a text editor or a file display command such as the UNIX cat command. By default, the mode bits on the *VolserLog* file grant the required r (read) permission to all users.

The Volume Server records operations only as it completes them, and so cannot recover from failures by reviewing the file. The log contents are useful for administrative evaluation of process failures and other problems.

See Also

"UserList(5)" on page 437, "bos_getlog(8)" on page 691, "volserver(8)" on page 925

Copyright

afs_cache

Name

afs_cache — Format of data stored in an AFS client disk cache

Description

The disk cache on a client machine is composed of multiple *Vn* files that contain the data, a *CacheItems* file that records index information for all of the *Vn* files, and a *VolumeItems* file that records the mapping between volume name and mount point for volumes.

When it initializes, the Cache Manager creates the cache files in the configured cache location. The standard directory name is */usr/vice/cache*, but it is acceptable to use a directory on a partition with more available space. To designate a different directory, change the value in the second field of the */usr/vice/etc/cacheinfo* file before issuing the afsd command, or include the -cachedir argument to the afsd command.

CacheItems

The CacheItems file records information about each file in the disk cache on a client machine (each *Vn* file). The information includes the file ID number and associated volume version number of the AFS file currently stored in the V*n* file, which enables the Cache Manager to determine which *Vn* file contains the AFS data it needs to present to an application.

As it initializes, the Cache Manager creates the binary-format *CacheItems* file in the same local disk cache directory as the *Vn* files that the *CacheItems* file describes, and it must always remain there.

VolumeItems

The *VolumeItems* file records the mapping between volume name and mount point for each volume that the Cache Manager has accessed since it initialized on a client machine using a disk cache. The Cache Manager uses the mappings to respond correctly to queries about the current working directory, which can come from the operating system or commands such as the UNIX pwd command.

As it initializes, the Cache Manager creates the binary-format *VolumeItems* file in the local disk cache directory, and it must always remain there.

Vn

A *Vn* file can store a chunk of cached AFS data on a client machine that is using a disk cache. As the Cache Manager initializes, it verifies that the local disk cache directory houses a number of *Vn* files equal to the largest of the following:

- 100

- One and a half times the result of dividing the cache size by the chunk size (cachesize / chunksize * 1.5).
- The result of dividing the cache size by 10 MB (10,240).

The Cache Manager determines the cache size from the -blocks argument to the afsd command, or if the argument is not included, from the third field of the `/usr/vice/etc/cacheinfo` file. The default chunk size is 64 KB; use the -chunksize argument to the afsd command to override it. To override the default number of chunks resulting from the calculation, include the -files argument to the afsd command. "afsd(8)" on page 523 describes the restrictions on acceptable values for each of the arguments.

If the disk cache directory houses fewer `Vn` files than necessary, the Cache Manager creates new ones, assigning each a unique integer *n* that distinguishes it from the other files; the integers start with 1 and increment by one for each `Vn` file created. The Cache Manager removes files if there are more than necessary. The Cache Manager also adds and removes `Vn` files in response to the fs setcachesize command, which can be used to alter the cache size between reboots.

`Vn` files expand and contract to accommodate the size of the AFS directory listing or file they temporarily house. As mentioned, by default each `Vn` file holds up to 64 KB (65,536 bytes) of a cached AFS element. AFS elements larger than 64 KB are divided among multiple V*n* files. If an element is smaller than 64 KB, the `Vn` file expands only to the required size. A `Vn` file accommodates only a single element, so if there many small cached elements, it is possible to exhaust the available `Vn` files without reaching the maximum cache size.

Cautions

Editing or removing the `CacheItems` or `VolumeItems` files or a `Vn` file can cause a kernel panic. If the contents of `Vn` files seem out of date, clear the files by using the fs flush or fs flushvolume command. If any of the cache files are accidentally modified or deleted, rebooting the machine usually restores normal performance.

To alter cache size (and thus the number of `Vn` files) between reboots, use the fs setcachesize command. Alternatively, alter the value of the -blocks, -files or -chunksize arguments to the afsd command invoked in the machine's AFS initialization file, and reboot. To refresh the contents of one or more `Vn` files, use the fs flush or fs flushvolume command.

See Also

"cacheinfo(5)" on page 467, "afsd(8)" on page 523, "fs_checkvolumes(1)" on page 53, "fs_flush(1)" on page 73, "fs_flushvolume(1)" on page 77, "fs_setcachesize(1)" on page 123

Copyright

afs_volume_header

Name

`afs_volume_header` — Represents an AFS volume

Description

The *Vvol_ID.vol* file is the header file for the AFS volume with volume ID *vol_ID*. There is one such file for each volume stored on an AFS server (*/vicep*) partition. The header file stores information that includes the volume's name, ID number, type (read/write, read-only, or backup), size and status (online, offline, or busy). To display information from the header file, use the vos listvol or vos examine command.

The header file points to, but does not contain, the actual data in the volume. It is not possible to access the AFS data except by mounting the volume in the AFS filespace and reading its contents through the Cache Manager.

Copyright

afsmonitor

Name

`afsmonitor` — Provides instructions for the afsmonitor command

Description

The afsmonitor configuration file determines which machines the afsmonitor command probes for File Server or Cache Manager statistics and which statistics it gathers. Use the -config argument to the afsmonitor command to identify the configuration file to use.

The instructions that can appear in the configuration file are as follows:

cm <*host name*>

Names a client machine for which to display Cache Manager statistics. The order of `cm` lines in the file determines the order in which client machines appear from top to bottom on the `System Overview` and `Cache Managers` output screens.

fs <*host name*>

Names a file server machine for which to display File Server statistics. The order of `fs` lines in the file determines the order in which file server machines appear from top to bottom on the `System Overview` and `File Servers` output screens.

thresh (fs | cm) <*field*> <*thresh*> [<*cmd*>] [<*arg*> ...]

Assigns the threshold value *thresh* to the statistic *field*, for either a File Server statistic (`fs`) or a Cache Manager statistic (`cm`). The optional *cmd* field names a binary or script to execute each time the value of the statistic changes from being below *thresh* to being at or above *thresh*. A change between two values that both exceed *thresh* does not retrigger the binary or script. The optional *arg* fields are additional values that the afsmonitor program passes as arguments to the *cmd* command. If any of them include one or more spaces, enclose the entire field in double quotes.

The afsmonitor program passes the following parameters to the *cmd*:

```
&lt;hostname&gt; (fs|cm) &lt;field&gt; &lt;thresh&gt; &lt;actual&gt; [&lt;arg&gt; ..
```

The parameters `fs`, `cm`, <field>, <thresh>, and <arg> correspond to the values with the same name on the thresh line. The <hostname> parameter identifies the file server or client machine where the statistic has crossed the threshold, and the <actual> parameter is the actual value of <field> that exceeds the threshold value.

Use the `thresh` line to set either a global threshold, which applies to all file server machines listed on `fs` lines or client machines listed on `cm` lines in the configuration file, or a machine-specific threshold, which applies to only one file server or client machine.

- To set a global threshold, place the thresh line before any of the `fs` or `cm` lines in the file.
- To set a machine-specific threshold, place the thresh line below the corresponding `fs` or `cm` line, and above any other `fs` or `cm` lines. A machine-specific threshold value always

overrides the corresponding global threshold, if set. Do not place a `thresh fs` line directly after a `cm` line or a `thresh cm` line directly after a `fs` line.

show (fs | cm) *field/group/section*

Specifies which individual statistic, group of statistics, or section of statistics to display on the `File Servers` screen (`fs`) or `Cache Managers` screen (`cm`) and the order in which to display them. The appendix of afsmonitor statistics in the *IBM AFS Administration Guide* specifies the group and section to which each statistic belongs. Include as many `show` lines as necessary to customize the screen display as desired, and place them anywhere in the file. The top-to-bottom order of the `show` lines in the configuration file determines the left-to-right order in which the statistics appear on the corresponding screen.

If there are no `show` lines in the configuration file, then the screens display all statistics for both Cache Managers and File Servers. Similarly, if there are no `show fs` lines, the `File Servers` screen displays all file server statistics, and if there are no `show cm` lines, the `Cache Managers` screen displays all client statistics.

 # comments

Precedes a line of text that the afsmonitor program ignores because of the initial number (#) sign, which must appear in the very first column of the line.

For a list of the values that can appear in the *field/group/section* field of a `show` instruction, see the afsmonitor statistics appendix to the *IBM AFS Administration Guide*.

See Also

"afsmonitor(1)" on page 25

Copyright

afszcm.cat

Name

`afszcm.cat` — Error message catalog for debugging the Cache Manager

Description

The *`afszcm.cat`* file is a message catalog for the Cache Manager. The fstrace dump command interpreter uses it in conjunction with the standard UNIX catalog utilities to translate Cache Manager operation codes into character strings as it writes traces in the fstrace trace log, which makes the log more readable.

The conventional location for the file is the */usr/vice/etc/C/* directory. It can be placed in another directory if the NLSPATH and LANG environment variables are set appropriately.

See Also

"afsd(8)" on page 523, "fstrace_dump(8)" on page 789

Copyright

bdb.DB0, bdb.DBSYS1

Name

bdb.DB0, bdb.DBSYS1 — Contain the Backup Database and associated log

Description

The *bdb.DB0* file contains the Backup Database, which records configuration information used by the AFS Backup System along with cross-indexed records of the tapes created and volumes dumped using the Backup System commands.

The *bdb.DBSYS1* file is a log file in which the Backup Server (buserver process) logs each database operation before performing it. When an operation is interrupted, the Backup Server replays the log to complete the operation.

Both files are in binary format and reside in the */usr/afs/db* directory on each database server machine that runs the Backup Server. When the Backup Server starts or restarts on a given machine, it establishes a connection with its peers and verifies that its copy of the *bdb.DB0* file matches the copy on the other database server machines. If not, the Backup Servers use AFS's distributed database technology, Ubik, to distribute to all of the machines the copy of the database with the highest version number.

Use the commands in the backup suite to administer the Backup Database. It is advisable to create a backup copy of the *bdb.DB0* file on tape on a regular basis, using the UNIX tar command or another local disk backup utility.

See Also

"backup(8)" on page 531, "backup_savedb(8)" on page 625, "buserver(8)" on page 763

Copyright

butc

Name

`butc` — Defines Tape Coordinator instructions for automated tape devices

Description

The *`CFG_device_name`* file includes instructions that configure a Tape Coordinator (butc) for use with automated backup devices such as tape stackers and jukeboxes, enable the Tape Coordinator to dump and restore data to a *backup data file* on a local disk device, and enable greater automation of other aspects of the backup process.

There is a separate configuration file for each tape device or backup data file. Creating the file is optional, and unnecessary if none of the instructions it can include pertain to a given tape device. The ASCII-format file must reside in the */usr/afs/backup* directory on the Tape Coordinator machine if it exists.

The *`CFG_device_name`* file does not replace the */usr/afs/backup/tapeconfig* file, a single copy of which still must exist on every Tape Coordinator machine.

To enable the Tape Coordinator to locate the configuration file, construct the variable part of the filename, *device_name*, as follows:

- For a tape device, strip off the initial `/dev/` string from the device name, and replace any other slashes in the name with underscores. For example, *`CFG_rmt_4m`* is the appropriate filename for a device called */dev/rmt/4m*.
- For a backup data file, strip off the initial slash (`/`) and replace any other slashes in the name with underscores. For example, *`CFG_var_tmp_FILE`* is the appropriate filename for a backup data file called */var/tmp/FILE*.

The *`CFG_device_name`* file lists one or more of the following instructions, each on its own line. All are optional, and they can appear in any order. A more detailed description of each instruction follows the list:

ASK

Controls whether the Tape Coordinator prompts for guidance when it encounters error conditions.

AUTOQUERY

Controls whether the Tape Coordinator prompts for the first tape.

BUFFERSIZE

Sets the size of the memory buffer the Tape Coordinator uses when transferring data.

FILE

Controls whether the dump is written to a tape device or a file.

MOUNT

Identifies the file that contains routines for inserting tapes into the device's drive.

NAME_CHECK

Controls whether the Tape Coordinator verifies that a tape's AFS tape name matches the dump being written.

UNMOUNT

Identifies the file that contains routines for removing tapes from the device's drive.

The ASK Instruction

The `ASK` instruction takes a boolean value as its argument, in the following format:

```
ASK (YES | NO)
```

When the value is `YES`, the Tape Coordinator generates a prompt in its window, requesting a response to the error cases described in the following list. This is the default behavior if the `ASK` instruction does not appear in the *`CFG_device_name`* file.

When the value is `NO`, the Tape Coordinator does not prompt in error cases, but instead uses the automatic default responses described in the following list. The Tape Coordinator also logs the error in the *`TE_device_name`* file. Suppressing the prompts enables the Tape Coordinator to run unattended, though it still prompts for insertion of tapes unless the `MOUNT` instruction is used.

The error cases controlled by this instruction are the following:

- The Backup System is unable to dump a volume while running the backup dump command. With a `YES` value, the Tape Coordinator prompts to offer three choices: try to dump the volume again immediately, omit the volume from the dump but continue the operation, or terminate the operation. With a `NO` value, the Tape Coordinator omits the volume from the dump and continues the operation.
- The Backup System is unable to restore a volume while running the backup diskrestore, backup volrestore, or backup volsetrestore command. With a `YES` value, the Tape Coordinator prompts to offer two choices: omit the volume and continue restoring the other volumes, or terminate the operation. With a `NO` value, it continues the operation without prompting, omitting the problematic volume but restoring the remaining ones.
- The Backup System cannot determine if the dump set includes any more tapes, while running the backup scantape command (the reference page for that command discusses possible reasons for this problem). With a `YES` value, the Tape Coordinator prompts to ask if there are more tapes to scan. With a `NO` value, it proceeds as though there are more tapes and invokes the routine named by the `MOUNT` instruction in the configuration file, or prompts the operator to insert the next tape.
- The Backup System determines that the tape contains an unexpired dump while running the backup labeltape command. With a `YES` value, the Tape Coordinator prompts to offer two choices: continue or terminate the labeling operation. With a `NO` value, it terminates the operation without relabeling the tape.

The AUTOQUERY Instruction

The AUTOQUERY instruction takes a boolean value as its argument, in the following format:

```
AUTOQUERY (YES | NO)
```

When the value is YES, the Tape Coordinator checks for the MOUNT instruction in the configuration file when it needs to read the first tape involved in an operation. As described for that instruction, it then either prompts for the tape or invokes the specified routine to mount the tape. This is the default behavior if the AUTOQUERY instruction does not appear in the configuration file.

When the value is NO, the Tape Coordinator assumes that the first tape required for an operation is already in the drive. It does not prompt the operator or invoke the MOUNT routine unless there is an error in accessing the first tape. This setting is equivalent in effect to including the -noautoquery flag to the butc command.

Note that the setting of the AUTOQUERY instruction controls the Tape Coordinator's behavior only with respect to the first tape required for an operation. For subsequent tapes, the Tape Coordinator always checks for the MOUNT instruction. It also refers to the MOUNT instruction if it encounters an error while attempting to access the first tape.

The BUFFERSIZE Instruction

The BUFFERSIZE instruction takes an integer value, and optionally units, in the following format:

```
BUFFERSIZE &lt;size&gt;[(k | K | m | M | g | G)]
```

where <size> specifies the amount of memory the Tape Coordinator allocates to use as a buffer during both dump and restore operations. The default unit is bytes, but use k or K to specify kilobytes, m or M for megabytes, and g or G for gigabytes. There is no space between the <size> value and the units letter.

By default, the Tape Coordinator uses a 16 KB buffer during dump operations. As it receives volume data from the Volume Server, the Tape Coordinator gathers 16 KB of data in the buffer before transferring the entire 16 KB to the tape device or backup data file. Similarly, during a restore operation the Tape Coordinator by default buffers 32 KB of data from the tape device or backup data file before transferring the entire 32 KB to the Volume Server for restoration into the file system. Buffering makes the volume of data flowing to and from a tape device more even and so promotes tape streaming, which is the most efficient way for a tape device to operate.

In a normal network configuration, the default buffer sizes are usually large enough to promote tape streaming. If the network between the Tape Coordinator machine and file server machines is slow, it can help to increase the buffer size.

The FILE Instruction

The FILE instruction takes a boolean value as its argument, in the following format:

```
FILE (NO | YES)
```

When the value is NO, the Tape Coordinator writes to a tape device during a dump operation and reads from one during a restore operation. This is the default behavior if the FILE instruction does not appear in the configuration file.

When the value is YES, the Tape Coordinator writes volume data to a backup data file on the local disk during a dump operation and reads volume data from a file during a restore operation. If the file does not exist when the Tape Coordinator attempts to access it to write a dump, the Tape Coordinator creates it. For a restore operation to succeed, the file must exist and contain volume data previously written to it by a backup dump operation.

When the value is YES, the backup data file's complete pathname must appear (instead of a tape drive device name) in the third field of the corresponding port offset entry in the local */usr/afs/backup/tapeconfig* file. If the field instead refers to a tape device, dump operations appear to succeed but are inoperative. It is not possible to restore data that was accidently dumped to a tape device while the FILE instruction was set to YES. (In the same way, if the FILE instruction is set to NO, the *tapeconfig* entry must refer to an actual tape device.)

Rather than put an actual file pathname in the third field of the *tapeconfig* file, however, the recommended configuration is to create a symbolic link in the */dev* directory that points to the actual file pathname, and record the symbolic link in this field. This configuration has a couple of advantages:

- It makes the *device_name* portion of the *CFG_device_name*, *TE_device_name*, and *TL_device_name* names as short as possible. Because the symbolic link is in the */dev* directory as though it were a tape device, the device configuration file's name is constructed by stripping off the entire */dev/* prefix, instead of just the initial slash. If, for example, the symbolic link is called */dev/FILE*, the device configuration file name is *CFG_FILE*, whereas if the actual pathname */var/tmp/FILE* appears in the tapeconfig file, the file's name must be *CFG_var_tmp_FILE*.
- It provides for a more graceful, and potentially automated, recovery if the Tape Coordinator cannot write a complete dump into the backup data file (because the partition housing the backup data file becomes full, for example). The Tape Coordinator's reaction to this problem is to invoke the MOUNT script, or to prompt the operator if the MOUNT instruction does not appear in the configuration file.
  - If there is a MOUNT routine, the operator can prepare for this situation by adding a subroutine that changes the symbolic link to point to another backup data file on a partition where there is space available.
  - If there is no MOUNT instruction, the prompt enables the operator manually to change the symbolic link to point to another backup data file, then press Return to signal that the Tape Coordinator can continue the operation.

If the third field in the *tapeconfig* file names the actual file, there is no way to recover from exhausting the space on the partition that houses the backup data file. It is not possible to change the *tapeconfig* file in the middle of an operation.

When writing to a backup data file, the Tape Coordinator writes data at 16 KB offsets. If a given block of data (such as the marker that signals the beginning or end of a volume) does not fill the entire 16 KB, the Tape Coordinator still skips to the next offset before writing the next block. In the output of a backup dumpinfo command issued with the -id option, the value in the Pos column is the ordinal of the 16-KB offset at which the volume data begins, and so is not generally only one higher than the position number on the previous line, as it is for dumps to tape.

The MOUNT Instruction

The MOUNT instruction takes a pathname as its argument, in the following format:

```
MOUNT &lt;filename&gt;
```

The referenced executable file must reside on the local disk and contain a shell script or program that directs an automated tape device, such as a jukebox or stacker, to mount a tape (insert it into the tape reader). The operator must write the routine to invoke the mount command specified by the device's manufacturer; AFS does not include any scripts, although an example appears in "EXAMPLES" on page . The script or program inherits the Tape Coordinator's AFS authentication status.

When the Tape Coordinator needs to mount a tape, it checks the configuration file for a MOUNT instruction. If there is no MOUNT instruction, the Tape Coordinator prompts the operator to insert a tape before it attempts to open the tape device. If there is a MOUNT instruction, the Tape Coordinator executes the routine in the referenced file. The routine invoked by the MOUNT instruction inherits the local identity (UNIX UID) and AFS tokens of the butc command's issuer.

There is an exception to this sequence: if the AUTOQUERY NO instruction appears in the configuration file, or the -noautoquery flag was included on the butc command, then the Tape Coordinator assumes that the operator has already inserted the first tape needed for a given operation. It attempts to read the tape immediately, and only checks for the MOUNT instruction or prompts the operator if the tape is missing or is not the required one.

When the Tape Coordinator invokes the routine indicated by the MOUNT instruction, it passes the following parameters to the routine in the indicated order:

- The tape device or backup data file's pathname, as recorded in the */usr/afs/backup/tapeconfig* file.
- The tape operation, which (except for the exceptions noted in the following list) matches the backup command operation code used to initiate the operation:
  - appenddump (when a backup dump command includes the -append flag).
  - dump (when a backup dump command does not include the -append flag).
  - labeltape

- `readlabel`
- `restore` (for a backup diskrestore, backup volrestore, or backup volsetrestore command).
- `restoredb`
- `savedb`
- `scantape`

- The number of times the Tape Coordinator has attempted to open the tape device or backup data file. If the open attempt returns an error, the Tape Coordinator increments this value by one and again invokes the `MOUNT` instruction.
- The tape name. For some operations, the Tape Coordinator passes the string `none`, because it does not know the tape name (when running the backup scantape or backup readlabel, for example), or because the tape does not necessarily have a name (when running the backup labeltape command, for example).
- The tape ID recorded in the Backup Database. As with the tape name, the Backup System passes the string `none` for operations where it does not know the tape ID or the tape does not necessarily have an ID.

The routine invoked by the `MOUNT` instruction must return an exit code to the Tape Coordinator:

- Code 0 (zero) indicates that the routine successfully mounted the tape. The Tape Coordinator continues the backup operation. If the routine invoked by the `MOUNT` instruction does not return this exit code, the Tape Coordinator never calls the `UNMOUNT` instruction.
- Code 1 (one) indicates that the routine failed to mount the tape. The Tape Coordinator terminates the operation.
- Any other code indicates that the routine was not able to access the correct tape. The Tape Coordinator prompts the operator to insert the correct tape.

If the backup command was issued in interactive mode and the operator issues the backup kill command while the `MOUNT` routine is running, the Tape Coordinator passes the termination signal to the routine; the entire operation terminates.

The NAME_CHECK Instruction

The `NAME_CHECK` instruction takes a boolean value as its argument, in the following format:

```
NAME_CHECK (YES | NO)
```

When the value is `YES` and the tape does not have a permanent name, the Tape Coordinator checks the AFS tape name when dumping a volume in response to the backup dump command. The AFS tape name must be `<NULL>` or match the tape name that the backup dump operation assigns based on the volume set and dump level names. This is the default behavior if the `NAME_CHECK` instruction does not appear in the configuration file.

When the value is `NO`, the Tape Coordinator does not check the AFS tape name before writing to the tape.

The Tape Coordinator always checks that all dumps on the tape are expired, and refuses to write to a tape that contains unexpired dumps.

The UNMOUNT Instruction

The `UNMOUNT` instruction takes a pathname as its argument, in the following format:

```
UNMOUNT &lt;filename&gt;
```

The referenced executable file must reside on the local disk and contain a shell script or program that directs an automated tape device, such as a jukebox or stacker, to unmount a tape (remove it from the tape reader). The operator must write the routine to invoke the unmount command specified by the device's manufacturer; AFS does not include any scripts, although an example appears in "EXAMPLES" on page . The script or program inherits the Tape Coordinator's AFS authentication status.

After closing a tape device, the Tape Coordinator checks the configuration file for an `UNMOUNT` instruction, whether or not the close operation succeeds. If there is no `UNMOUNT` instruction, the Tape Coordinator takes no action, in which case the operator must take the action necessary to remove the current tape from the drive before another can be inserted. If there is an `UNMOUNT` instruction, the Tape Coordinator executes the referenced file. It invokes the routine only once, passing in the following parameters:

- The tape device pathname (as specified in the */usr/afs/backup/tapeconfig* file).
- The tape operation (always unmount).

Privilege Required

The file is protected by UNIX mode bits. Creating the file requires the `w` (write) and `x` (execute) permissions on the */usr/afs/backup* directory. Editing the file requires the `w` (write) permission on the file.

Examples

The following example configuration files demonstrate one way to structure a configuration file for a stacker or backup dump file. The examples are not necessarily appropriate for a specific cell; if using them as models, be sure to adapt them to the cell's needs and equipment.

Example *CFG_device_name* File for Stackers

In this example, the administrator creates the following entry for a tape stacker called `stacker0.1` in the */usr/afs/backup/tapeconfig* file. It has port offset 0.

```
2G   5K   /dev/stacker0.1   0
```

The administrator includes the following five lines in the */usr/afs/backup/CFG_stacker0.1* file. To review the meaning of each instruction, see "DESCRIPTION" on page .

```
MOUNT /usr/afs/backup/stacker0.1
UNMOUNT /usr/afs/backup/stacker0.1
AUTOQUERY NO
ASK NO
NAME_CHECK NO
```

Finally, the administrator writes the following executable routine in the */usr/afs/backup/stacker0.1* file referenced by the `MOUNT` and `UNMOUNT` instructions in the *CFG_stacker0.1* file.

```
#! /bin/csh -f


set devicefile = $1
set operation = $2
set tries = $3
set tapename = $4
set tapeid = $5


set exit_continue = 0
set exit_abort = 1
set exit_interactive = 2


#--------------------------------------------


if (${tries} &gt; 1) then
   echo "Too many tries"
   exit ${exit_interactive}
endif


if (${operation} == "unmount") then
   echo "UnMount: Will leave tape in drive"
   exit ${exit_continue}
endif


if ((${operation} == "dump")     |\
    (${operation} == "appenddump")     |\
    (${operation} == "savedb"))  then


    stackerCmd_NextTape ${devicefile}
    if (${status} != 0)exit${exit_interactive}
```

```
    echo "Will continue"
    exit ${exit_continue}
endif


if ((${operation} == "labeltape")    |\
    (${operation} == "readlabel")) then
   echo "Will continue"
   exit ${exit_continue}
endif


echo "Prompt for tape"
exit ${exit_interactive}
```

This routine uses two of the parameters passed to it by the Backup System: `tries` and `operation`. It follows the recommended practice of prompting for a tape if the value of the `tries` parameter exceeds one, because that implies that the stacker is out of tapes.

For a backup dump or backup savedb operation, the routine calls the example `stackerCmd_NextTape` function provided by the stacker's manufacturer. Note that the final lines in the file return the exit code that prompts the operator to insert a tape; these lines are invoked when either the stacker cannot load a tape or a the operation being performed is not one of those explicitly mentioned in the file (such as a restore operation).

Example *CFG_device_name* File for Dumping to a Data File

In this example, the administrator creates the following entry for a backup data file called *HSM_device* in the */usr/afs/backup/tapeconfig* file. It has port offset 20.

```
1G   0K   /dev/HSM_device   20
```

The administrator includes the following lines in the */usr/afs/backup/CFG_HSM_device* file. To review the meaning of each instruction, see "DESCRIPTION" on page .

```
MOUNT /usr/afs/backup/file
FILE YES
ASK NO
```

Finally, the administrator writes the following executable routine in the */usr/afs/backup/file* file referenced by the `MOUNT` instruction in the *CFG_HSM_device* file, to control how the Tape Coordinator handles the file.

```
#! /bin/csh -f
set devicefile = $1
set operation = $2
set tries = $3
set tapename = $4
set tapeid = $5
```

```
set exit_continue = 0
set exit_abort = 1
set exit_interactive = 2


#-----------------------------------------


if (${tries} &gt; 1) then
   echo "Too many tries"
   exit ${exit_interactive}
endif


if (${operation} == "labeltape") then
   echo "Won't label a tape/file"
   exit ${exit_abort}
endif


if ((${operation} == "dump")   |\
    (${operation} == "appenddump")   |\
    (${operation} == "restore")   |\
    (${operation} == "savedb")    |\
    (${operation} == "restoredb")) then


   /bin/rm -f ${devicefile}
   /bin/ln -s /hsm/${tapename}_${tapeid} ${devicefile}
   if (${status} != 0) exit ${exit_abort}
endif


exit ${exit_continue}
```

Like the example routine for a tape stacker, this routine uses the `tries` and `operation` parameters passed to it by the Backup System. The `tries` parameter tracks how many times the Tape Coordinator has attempted to access the file. A value greater than one indicates that the Tape Coordinator cannot access it, and the routine returns exit code 2 (`exit_interactive`), which results in a prompt for the operator to load a tape. The operator can use this opportunity to change the name of the backup data file specified in the tapeconfig file.

The primary function of this routine is to establish a link between the device file and the file to be dumped or restored. When the Tape Coordinator is executing a `backup dump`, `backup restore`, `backup savedb`, or `backup restoredb` operation, the routine invokes the UNIX `ln -s` command to create a symbolic link from the backup data file named in the *`tapeconfig`* file to the actual file to use (this is the recommended method). It uses the value of the `tapename` and `tapeid` parameters to construct the file name.

See Also

"tapeconfig(5)" on page 493, "backup_diskrestore(8)" on page 565, "backup_dump(8)" on page 571, "backup_restoredb(8)" on page 621, "backup_savedb(8)" on page 625, "backup_volrestore(8)" on page 647, "backup_volsetrestore(8)" on page 653

Copyright

butc_logs

Name

butc_logs — Message logs from the Tape Coordinator process

Description

The Backup System Tape Coordinator (butc) process generates two log files per device, one for error messages and one for actions.

Error Message Log

The *TE_device_name* file logs error messages generated by the Backup System Tape Coordinator that controls the tape device or backup data file indicated by *device_name*.

As the Tape Coordinator initializes, it creates the file in ASCII format in the */usr/afs/backup* directory. If there is an existing file, the Tape Coordinator renames it to *TE_device_name.old*>, overwriting the existing *TE_device_name.old*> file if it exists.

For a tape device, the Tape Coordinator derives the variable *device_name* portion of the filename from the device pathname listed in the local */usr/afs/backup/tapeconfig* file, by stripping off the initial /dev/ string and replacing any other slashes in the name with underscores. For example, the filename for a device called */dev/rmt/4m* is *TE_rmt_4m*. Similarly, for a backup data file the Tape Coordinator strips off the initial slash (/) and replaces any other slashes in the name with underscores. For example, the filename for a backup data file called */var/tmp/FILE* is *TE_var_tmp_FILE*.

The messages in the file describe the error and warning conditions the Tape Coordinator encounters as it operates. For instance, a message can list the volumes that are inaccessible during a dump operation, or warn that the Tape Coordinator is overwriting a tape or backup data file. The messages also appear in the */usr/afs/backup/TL_device_name* file, which traces most of the Tape Coordinator's actions.

Action Log

The *TL_device_name* file logs the actions performed by the Backup System Tape Coordinator that controls the tape device or backup data file indicated by *device_name*. It also records the same error and warning messages written to the *TE_device_name* file.

As the Tape Coordinator initializes, it creates the file in ASCII format in the */usr/afs/backup* directory. If there is an existing file, the Tape Coordinator renames it to *TL_device_name.old*, overwriting the existing *TL_device_name.old* file if it exists.

For a tape device, the Tape Coordinator derives the variable *device_name* portion of the filename from the device pathname listed in the local */usr/afs/backup/tapeconfig* file, by stripping off the initial /dev/ string and replacing any other slashes in the name with underscores. For example, the filename for a device called */dev/rmt/4m* is *TL_rmt_4m*. Similarly, for a backup data file the Tape Coordinator strips off the initial slash (/) and replaces

any other slashes in the name with underscores. For example, the filename for a backup data file called */var/tmp/FILE* is *TL_var_tmp_FILE*.

See Also

"tapeconfig(5)" on page 493, "butc(8)" on page 767

Copyright

cacheinfo

Name

`cacheinfo` — Defines configuration parameters for the Cache Manager

Description

The *`cacheinfo`* file defines configuration parameters for the Cache Manager, which reads the file as it initializes.

The file contains a single line of ASCII text and must reside in the */usr/vice/etc* directory. Use a text editor to create it during initial configuration of the client machine; the required format is as follows:

```
&lt;mount&gt;:&lt;cache&gt;:&lt;size&gt;
```

where

<mount>

Names the local disk directory at which the Cache Manager mounts the AFS namespace. It must exist before the afsd program runs. The conventional value is */afs*. Using any other value prevents traversal of pathnames that begin with */afs* (such as pathnames to files in foreign cells that do use the conventional name). The -mountdir argument to the afsd command overrides this value.

<cache>

Names the local disk directory to use as a cache. It must exist before the afsd program runs. The standard value is */usr/vice/cache*, but it is acceptable to substitute a directory on a partition with more available space. Although the Cache Manager ignores this field when configuring a memory cache, a value must always appear in it. The -cachedir argument to the afsd command overrides this value.

<size>

Specifies the cache size as a number of 1-kilobyte blocks. Larger caches generally yield better performance, but a disk cache must not exceed 90% of the space available on the cache partition (85% for AIX systems), and a memory cache must use no more than 25% of available machine memory.

The -blocks argument to the afsd command overrides this value. To reset cache size without rebooting on a machine that uses disk caching, use the fs setcachesize command. To display the current size of a disk or memory cache between reboots, use the fs getcacheparms command.

Examples

The following example cacheinfo file mounts the AFS namespace at */afs*, establishes a disk cache in the */usr/vice/cache* directory, and defines cache size as 50,000 1-kilobyte blocks.

```
/afs:/usr/vice/cache:50000
```

See Also

"afsd(8)" on page 523, "fs_getcacheparms(1)" on page 79, "fs_setcachesize(1)" on page 123

Copyright

fms.log

Name

`fms.log` — Records output from the fms command

Description

The *`fms.log`* file records the output generated by the fms command. The output includes two numbers that can appear in a tape device's entry in the */usr/afs/backup/tapeconfig* file on the Tape Coordinator machine to which the tape device is attached:

- The capacity in bytes of the tape in the device.
- The size in bytes of the end-of-file (EOF) marks (often referred to simply as *filemarks*) that the tape device writes.

When transferring the numbers recorded in this file to the *`tapeconfig`* file, adjust them as specified in "tapeconfig(5)" on page 493, to improve Tape Coordinator performance during dump operations.

If the *`fms.log`* file does not already exist in the current working directory, the fms command interpreter creates it. In this case, the directory's mode bits must grant the `rwx` (read, write, and execute) permissions to the issuer of the command. If there is an existing file, the command interpreter overwrites it, so the file's mode bits need to grant only the w permission to the issuer of the fms command. The fms command interpreter also writes similar information to the standard output stream as it runs.

The file is in ASCII format. To display its contents, log onto the client machine and use a text editor or a file display command such as the UNIX cat command. By default, the mode bits on the *`fms.log`* file grant the required `r` permission only to the owner (which is the local superuser `root` by default).

Output

The first few lines of the file provide a simple trace of the fms command interpreter's actions, specifying (for example) how many blocks it wrote on the tape. The final two lines in the file specify tape capacity and filemark size in bytes, using the following format:

```
Tape capacity is &lt;tape_size&gt; bytes
File marks are &lt;filemark_size&gt; bytes
```

Examples

The following example of the fms.log file specifies that the tape used during the execution of the fms command had a capacity of 2,136,604,672 bytes, and that the tape device writes filemarks of size 1,910,220 bytes.

```
fms test started
wrote 130408 blocks
Tape capacity is 2136604672 bytes
File marks are 1910220 bytes
```

See Also

"tapeconfig(5)" on page 493, "fms(8)" on page 779

Copyright

kaserver.DB0, kaserver.DBSYS1

Name

`kaserver.DB0, kaserver.DBSYS1` — The Authentication Database and associated log

Description

The *`kaserver.DB0`* file contains the Authentication Database, which records server encryption keys and an encrypted form of all user passwords. The Authentication Server (kaserver process) uses the information in the database to enable secured communications between AFS server and client processes.

The *`kaserver.DBSYS1`* file is a log file in which the Authentication Server logs each database operation before performing it. When an operation is interrupted, the Authentication Server replays the log to complete the operation.

Both files are in binary format and reside in the */usr/afs/db* directory on each of the cell's database server machines. When the Authentication Server starts or restarts on a given machine, it establishes a connection with its peers and verifies that its copy of the database matches the copy on the other database server machines. If not, the Authentication Servers call on AFS's distributed database technology, Ubik, to distribute to all of the machines the copy of the database with the highest version number.

Always use the commands in the kas suite to administer the Authentication Database. It is advisable to create an archive copy of the database on a regular basis, using a tool such as the UNIX tar command.

See Also

"kadb_check(8)" on page 803, "kas(8)" on page 805, "kaserver(8)" on page 859

Copyright

kaserverauxdb

Name

kaserverauxdb — Records failed authentication attempts

Description

The file *kaserverauxdb* records failed authentication attempts for the local Authentication Server. The server creates it automatically in the */usr/afs/local* directory by default; use the -localfiles argument to the kaserver command to specify an alternate directory.

The *kaserverauxdb* file is an internal database used by the Authentication Server to prevent access by users who have exceeded the limit on failed authentication attempts defined in their Authentication Database entry. The Authentication Server refuses further attempts to authenticate to an account listed in the database until either an AFS system administrator issues the kas unlock command to unlock the account, or the timeout period defined in the user's Authentication Database entry passes.

The *kaserverauxdb* file is in binary format, so its contents are not directly accessible. However, the output from the kas examine command reports an account's maximum number of failed attempts, the lockout time, and whether the account is currently locked.

See Also

"kaserver.DB0(5)" on page 471, "kas_examine(8)" on page 817, "kas_unlock(8)" on page 857, "kaserver(8)" on page 859

Copyright

package

Name

package — Provides instructions for the package command

Description

The package configuration file defines the file system elements that the package command creates or alters on the local disk of an AFS client machine it is configuring. Use the -config or -fullconfig argument to the package command to identify the configuration file to use.

Summary of Configuration File Instructions

The configuration file can include one or more instances of each of the following instructions, each on its own line. A more detailed description of each instruction's syntax follows this list.

B

Defines a block special device, such as a disk, which deals with input in units of multi-byte command blocks.

C

Defines a character special device, such as a terminal or tty, which deals with input in single character units.

D

Creates a directory.

F

Creates or alters a file to match the contents of a specified source file.

L

Creates a symbolic link.

S

Defines a socket, which is a communications device for UDP and TCP/IP connections.

%define

Defines a variable or declares a string as defined.

%ifdef

Specifies an action to perform if a certain string is declared or defined.

%ifndef

Specifies an action to perform if a certain string is not declared or defined.

%include

Includes a library file.

%undef

Declares a string not to be defined, or a variable no longer to have a value.

The B and C Instructions for Defining Special Devices

The `B` instruction in a package configuration file defines a block special device, such as a disk, that deals with input in units of multi-byte command blocks. The `C` instruction defines a character special device, such as a terminal or tty, that deals with input in single character units. They share a common syntax:

```
(B | C) &lt;device&gt; &lt;major&gt; &lt;minor&gt; &lt;owner&gt; &lt;group&gt; &lt;mode&
```

where

B

Indicates the definition of a block special device. It must be a capital letter.

C

Indicates the definition of character special device. It must be a capital letter.

<device>

Names the special device to define. To learn the name format appropriate to the machine's system type, consult the hardware or operating system documentation.

<major>

Specifies the device's major device number in decimal format. To learn the correct value for the machine's system type, consult the hardware or operating system documentation.

<minor>

Specifies the device's minor device number in one of hexadecimal, octal, or decimal format. Precede a hexadecimal number with the string `0x` (zero and the letter `x`) or an octal number with a `0` (zero). A number without either prefix is interpreted as a decimal. To learn the correct value for the machine's system type, consult the hardware or operating system documentation.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the device's owner in the output from the UNIX `ls -l` command.

<group>

Specifies the group name or UNIX group ID (GID) of the group to be designated the device's group in the output from the UNIX `ls -lg` command.

<mode>

Defines the device's UNIX mode bits. Acceptable values are the standard three- or four-digit numbers corresponding to combinations of permissions. Examples: `755` corresponds to `rwxr-xr-x`, and `644` to `rw-r--r--`.

The D Instruction for Creating a Directory

The `D` instruction in a package configuration file creates a directory on the local disk. If a symbolic link, file, or other element on the local disk has the same name, it is replaced with a directory. If the directory already exists, its owner, group, and mode bits are changed if necessary to conform with the instruction. The instruction has the following syntax:

```
D[I&lt;update_code&gt;] &lt;directory&gt; &lt;owner&gt; &lt;group&gt; &lt;mode&gt;
```

where

D

Indicates the creation of a directory. It must be a capital letter.

<update_code>

Modulates the directory creation instruction. It is optional and follows the letter `D` directly, without an intervening space. Choose one of the two acceptable values:

X

Indicates that the directory is a lost+found directory (used by the fsck program).

R

Removes any subdirectory (along its contents) or file that exists in the existing directory on the local disk but for which an instruction does not appear in the configuration file.

<directory>

Specifies the full pathname of the directory to create.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the directory's owner in the output from the UNIX `ls -ld` command.

<group>

Specifies the name or UNIX group ID (GID) of the group to be designated the directory's group in the output from the UNIX `ls -lgd` command.

<mode>

Defines the directory's UNIX mode bits. Acceptable values are the standard three- or four-digit numbers corresponding to combinations of permissions. Examples: `755` corresponds to `drwxr-xr-x`, and `644` to `drw-r--r--`.

The F Instruction for Creating or Updating a File

The `F` instruction in a package configuration file creates or updates a file on the local disk by copying in the contents of the indicated source file, which can reside in AFS or on the local disk. If the package command interpreter cannot access the source file, it exits without executing any instruction in the configuration file.

If a file with the same name already exists on disk, the package command overwrites it with the contents of the source file, unless the `I` update code is used to prevent that. To add a `.old` extension to the current version of the file, include the `O` update code. To have the machine reboot automatically after the package program completes, include the `Q` update code.

If a symbolic link, directory, or other element on the local disk has the same name, it is replaced with the file (a directory's contents are first removed as necessary).

The instruction has the following syntax:

```
F[&lt;update_code&gt;] &lt;file&gt; &lt;source&gt; [&lt;owner&gt; &lt;group&gt; &lt;mode
```

where

F

Indicates the creation or update of a file. It must be a capital letter.

<update_code>

Modulates the file creation instruction. It is optional and follows the letter `F` directly, without an intervening space. Choose one or more of the four acceptable values, and list them in any order:

A

Indicates that the pathname in the <source> field is the complete pathname of the source file, including the filename. If this argument is omitted, the package command appends the pathname in the <file> field to the pathname in the <source> field to derive the source file's full name. This code allows the source and target filenames to differ.

I

Preserves the existing file called <file>, rather than overwriting it.

O

Saves the existing version of the file by appending a `.old` extension to it.

Q

Causes the package command to exit with status code 4 if it overwrites the file. If the standard package-related changes have been made to the machine's AFS initialization file, then status code 4 causes the machine to reboot automatically. Use this code when the machine must reboot if updates to the file are to have any effect (for example, if the operating system file -- */vmunix* or equivalent -- has changed).

<file>

Specifies the complete pathname on the local disk of the file to create or update, including the filename as the final element.

<source>

Specifies the pathname (local or AFS) of the file to copy to the local disk.

If the `A` update code is included, specify the source file's complete pathname. Otherwise, the package command derives the source file's full name by appending the *file* pathname to this pathname. For example, if the `A` update code is not included and the file */afs/abc.com/rs_aix42/bin/grep* is the source file for the */bin/grep* binary, the proper value in this field is */afs/abc.com/rs_aix42*.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the file's owner in the output from the UNIX `ls -l` command.

To copy the source file's owner to the target file, leave this field empty. In this case, the <group> and <mode> fields must also be empty.

<group>

Specifies the name or UNIX group ID (GID) of the group to be designated the file's group in the output from the UNIX `ls -lg` command.

To copy the source file's group to the target file, leave this field empty. In this case, the <owner> and <mode> fields must also be empty.

<mode>

Defines the file's UNIX mode bits. Acceptable values are the standard three- or four-digit numbers corresponding to combinations of permissions. Examples: `755` corresponds to `rwxr-xr-x`, and `644` to `rw-r--r--`.

To copy the source file's mode bits to the target file, leave this field empty. In this case, the <owner> and <group> fields must also be empty.

The L Instruction for Creating a Symbolic Link

The `L` instruction in a package configuration file creates a symbolic link on the local disk to a directory or file that exists either in AFS or elsewhere on the local disk. As with the standard UNIX `ln -s` command, the link is created even if the actual file or directory does not exist.

If a file or directory on the local disk already has the same name, the package command replaces it with a symbolic link.

The instruction has the following syntax:

```
L[I&lt;update_code&gt;] &lt;link&gt; &lt;path&gt; [&lt;owner&gt; &lt;group&gt; &lt;mode&
```

where

L

Indicates the creation of a symbolic link. It must be a capital letter.

<update_code>

Modulates the link creation instruction. It is optional and follows the letter `L` directly, without an intervening space. Choose one or both of the acceptable values, and list them in any order:

A

Indicates that the pathname in the <path> field is the complete pathname of the actual directory or file (including the filename for a file). If this argument is omitted, the package command appends the value in the <link> field to the pathname in the <path> field to derive the actual directory or file's full name. This code allows the name of the symbolic link and actual directory or file to differ.

I

Preserves the existing symbolic link called <link>, rather than overwriting it.

<link>

Specifies the complete local disk pathname of the symbolic link to create.

<path>

Specifies the pathname (local or AFS) of the directory or file to which the link refers. If the `A` update code is included, specify the directory or file's complete pathname. Otherwise, the package command derives the actual directory or file's full name by appending the value in the *link* field to this pathname. For example, if the `A` update code is not included and */etc/ftpd* is a symbolic link to the file */afs/abc.com/sun4x_56/etc/ftpd*, the proper value in this field is */afs/abc.com/sun4x_56*.

The package command interpreter correctly handles pathnames that begin with the `./` (period, slash) or `../` (two periods, slash) notation, interpreting them relative to the current working directory from which the package command is invoked.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the symbolic link's owner in the output from the UNIX `ls -l` command.

To designate the issuer of the package command (usually, the local superuser `root`) as the symbolic link's owner, leave this field empty. In this case, the <group> and <mode> fields must also be empty.

<group>

Specifies the name or UNIX group ID (GID) of the group to be designated the link's group in the output from the UNIX `ls -lg` command.

To have the symbolic link's group match the default group associated with the package command's issuer, leave this field empty. The issuer is usually the local superuser `root` and the default group is designated in the issuer's entry in the local */etc/passwd* file or equivalent. If this field is left empty, the <owner> and <mode> fields must also be empty.

<mode>

Defines the symbolic link's UNIX mode bits. Acceptable values are the standard three- or four-digit numbers corresponding to combinations of permissions. Examples: `755` corresponds to `rwxr-xr-x`, and `644` to `rw-r--r--`.

Leaving this field empty sets the symbolic link's mode bits to `777` (`rwxrwxrwx`). In this case, the <owner> and <group> fields must also be empty.

The S Instruction for Creating a Socket

The `S` instruction in a package configuration file creates a socket (a communications device for UDP or TCP/IP connections) on the local disk. The instruction has the following syntax:

```
S &lt;socket&gt; [&lt;owner&gt; &lt;group&gt; &lt;mode&gt;]
```

where

S

Indicates the creation of a socket. It must be a capital letter.

<socket>

Names the socket. The proper format depends on the local machine's operating system.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the socket's owner in the output from the UNIX `ls -l` command.

To designate the issuer of the package command (usually, the local superuser `root`) as the socket's owner, leave this field empty. In this case, the <group> and <mode> fields must also be empty.

<group>

Specifies the name or UNIX group ID (GID) of the group to be designated the socket's group in the output from the UNIX `ls -lg` command.

To have the symbolic link's group match the default group associated with the package command's issuer, leave this field empty. The issuer is usually the local superuser `root` and the default group is designated in the issuer's entry in the local */etc/passwd* file or equivalent. If this field is left empty, the <owner> and <mode> fields must also be empty.

<mode>

Defines the socket's UNIX mode bits. Acceptable values are the standard three- or four-digit numbers corresponding to combinations of permissions. Examples: `755` corresponds to `rwxr-xr-x`, and `644` to `rw-r--r--`.

Leaving this field empty sets the symbolic link's mode bits to `777` (`rwxrwxrwx`), modulated by the cell's umask. In this case, the <owner> and <group> fields must also be empty.

The %define or %undef Instructions

The `%define` instruction in a package configuration file declares or defines a variable, depending on its number of arguments:

- If followed by a single argument, it declares that argument to be defined. The argument is then available as a controller when mentioned in `%ifdef` and `%ifndef` statements, which evaluate to `true` and `false` respectively.
- If followed by two arguments, it defines the second argument as the value of the first. When the first argument appears later in this prototype or other prototype or library files as a variable -- surrounded by curly braces and preceded by a dollar sign, as in the example `${variable}` -- the package command interpreter substitutes the second argument for it.

The `%undef` statement negates the effect of a previous `%define` statement, declaring its argument to be defined no longer, or to have a value no longer if it is a variable.

The syntax for the two types of instruction are as follows:

```
%define &lt;declaration&gt;
%define &lt;variable&gt; &lt;value&gt;
%undef  &lt;declaration&gt;
%undef  &lt;variable&gt;
```

where

%define

Indicates a definition statement.

%undef

Indicates a statement that negates a definition.

<declaration>

Names the string being declared by a `%define` statement, or negated by an `%undef` statement.

<variable>

Specifies the name of the variable that a `%define` statement is defining, or an `%undef` statement is negating.

<value>

Specifies the value to substitute for the string in the <variable> field when it appears in the appropriate format (surrounded by curly braces and preceded by a dollar sign, as in the example `${variable}`), in this or other prototype and library files. It can include one or more words.

The %ifdef and %ifndef Instructions

The `%ifdef` instruction in a package configuration file specifies one or more actions to perform if the indicated string has been declared by a single-argument `%define` statement, or is a variable for which a value has been defined by a two-argument `%define` statement.

Similarly, the `%ifndef` instruction specifies one or more actions to perform if the indicated string has not been declared or is a variable without a value, either because no `%define` statement has defined it or an `%undef` statement has undefined it.

In both cases, the optional `%else` statement specifies one or more alternate actions to perform if the first statement evaluates to `false`. (For an `%ifdef` statement, the `%else` statement is executed if the indicated string has never been declared or is a variable without a value, or if an `%undef` statement has undefined either one; for an `%ifndef` statement, it is executed if the string has been declared or is a variable with a value.)

It is possible to nest any number of `%ifdef` and `%ifndef` statements.

The two types of statement share a common syntax:

```
(%ifdef | %ifndef) &lt;declaration&gt;
    &lt;action&gt;+
[%else [&lt;declaration&gt;]
    &lt;alternate_action&gt;+]
%endif &lt;declaration&gt;
```

where

%ifdef

Indicates that the statement evaluates as true if the string in the <declaration> field is declared or is a variable with a defined value.

%ifndef

Indicates that the statement evaluates as true if the string in the <declaration> field is not declared or is a variable without a defined value.

<declaration>

Specifies the string that must be declared or the variable name that must have a defined value for an `%ifdef` statement to evaluate as `true`, which results in the specified action being performed. For an `%ifndef` statement, the string must not be declared or the variable must have no defined value for the statement to evaluate as `true`. The first and third occurrences of <declaration> (the latter following the string `%endif`) are required. The second occurrence (following the string `%else`) is optional, serving only to clarify to which `%ifdef` or `%ifndef` statement the `%else` statement belongs.

<action>

Specifies each action to perform if the `%ifdef` or `%ifndef` statement evaluates as `true`. Each action must appear on a separate line. Acceptable types of actions are other statements beginning with a percent sign and definition instructions.

<alternate_action>

Specifies each action to perform if the `%ifdef` or `%ifndef` statement evaluates to `false`. Each action must appear on a separate line. Acceptable types of actions are other statements beginning with a percent sign and definition instructions.

The %include Instruction for Including a Library File

The `%include` instruction in a package configuration file includes the contents of the indicated library file in a configuration file that results from the compilation of the prototype file in which the `%include` instruction appears. It has the following syntax:

```
%include &lt;pathname&gt;
```

where

%include

Indicates a library file include statement.

<pathname>

Specifies the complete pathname of the library file to include. It can be in AFS or on the local disk, and can include one or more variables.

Cautions

The configuration file must be completely correct. If there are any syntax errors or incorrect values, the package command interpreter exits without executing any instruction.

Examples

The following example `B` and `C` instructions define a disk */dev/hd0a* with major and minor device numbers `1` and `0` and mode bits of `-rw-r--r--`, and a tty */dev/ttyp5* with major and minor device numbers `6` and `5` and mode bits of `-rw-rw-rw`. In both cases, the owner is `root` and the owning group `wheel`.

```
B /dev/hd0a 1 0 root wheel 644
C /dev/ttyp5 6 5 root wheel 666
```

The following example `D` instruction creates the local */usr* directory with owner `root` and group `wheel` and mode bits of `drwxr-xr-x`. The `R` update code removes any files and subdirectories that reside in the */usr* directory (if it already exists) but do not appear in the configuration file.

```
DR /usr root wheel 755
```

The following example `F` instruction, appropriate for a machine running AIX 4.2 in the ABC Corporation cell, creates or updates the local disk file */bin/grep*, using */afs/abc.com/rs_aix42/bin/grep* as the source.

```
F /bin/grep /afs/abc.com/rs_aix42 root wheel 755
```

The next example `F` instruction creates the */usr/vice/etc/ThisCell* file and specifies an absolute pathname for the source file, as indicated by the `A` update code. The `Q` code makes the package command return status code 4 as it exits, prompting a reboot of the machine if the standard package-related changes have been made to the machine's AFS initialization file. No values are provided for the owner, group and mode bits, so the file inherits them from the source file.

```
FAQ /usr/vice/etc/ThisCell /afs/abc.com/common/etc/ThisCell
```

The following example `L` instruction, appropriate for a machine running AIX 4.2 in the ABC Corporation cell, creates a symbolic link from */etc/ftpd* on the local disk to the file */afs/abc.com/rs_aix42/etc/ftpd*.

```
L /etc/ftpd /afs/abc.com/rs_aix42 root wheel 644
```

The following example S instruction defines the socket */dev/printer*.

```
S /dev/printer root wheel 777
```

The following example `%define` instruction defines the value for the variable `${diskmode}`. This variable is used elsewhere in the template to fill the <owner>, <group>, and <mode> fields in a `D`, `F`, or `L` instruction.

```
%define diskmode root wheel 644
```

The following example `%undef` instruction declares the string afsd not to be defined.

```
%undef afsd
```

The following example `%ifdef` instruction specifies that if the string `rs_aix42` is currently declared, then when the prototype file containing the instruction is compiled the three indicated library files are included. There is no alternate action defined. There must be `%define` statements earlier in the prototype file to declare `rs_aix42` and to assign a value to the `${wsadmin}` variable.

```
%ifdef rs_aix42
%include ${wsadmin}/lib/rs_aix42.readonly
%include ${wsadmin}/lib/rs_aix42.generic
%include ${wsadmin}/lib/rs_aix42.generic.dev
%endif rs_aix42
```

The following example `%ifndef` instruction, appropriate for the State University cell, defines `stateu.edu` as the value of the `${cell}` variable if it does not already have a value.

```
%ifndef cell
%define cell stateu.edu
%endif cell
```

The following example `%include` instruction includes the library file `base.generic` from the *`lib`* subdirectory of the directory in which package-related files reside. The `${wsadmin}` variable resolves to an actual pathname (such as */`afs/abc.com/wsadmin`*) during compilation.

```
%include ${wsadmin}/lib/base.generic
```

See Also

"package(8)" on page 869

Copyright

prdb.DB0, prdb.DBSYS1

Name

`prdb.DB0, prdb.DBSYS1` — Contain the Protection Database and associated log

Description

The *prdb.DB0* file contains the Protection Database, which maps AFS user, machine, and group names to their respective IDs (AFS UIDs and GIDs) and tracks group memberships. The Protection Server (ptserver process) uses the information in the database to help the File Server grant data access to authorized users.

The *prdb.DBSYS1* file is a log file in which the Protection Server logs each database operation before performing it. When an operation is interrupted, the Protection Server replays the log to complete the operation.

Both files are in binary format and reside in the */usr/afs/db* directory on each of the cell's database server machines. When the Protection Server starts or restarts on a given machine, it establishes a connection with its peers and verifies that its copy of the database matches the copy on the other database server machines. If not, the Protection Servers call on AFS's distributed database technology, Ubik, to distribute to all of the machines the copy of the database with the highest version number.

Always use the commands in the pts suite to administer the Protection Database. It is advisable to create an archive copy of the database on a regular basis, using a tool such as the UNIX tar command.

See Also

"prdb_check(8)" on page 873, "pts(1)" on page 173, "ptserver(8)" on page 875

Copyright

salvage.lock

Name

`salvage.lock` — Prevents multiple simultaneous salvage operations on a partition

Description

The *`salvage.lock`* file guarantees that only one Salvager (salvager process) runs at a time on a file server machine (the single process can fork multiple subprocesses to salvage multiple partitions in parallel). As the Salvager initializes, it creates the empty (zero-length) file in the */usr/afs/local* directory and invokes the flock system call on it. It removes the file when it completes the salvage operation. Because the Salvager must lock the file to run, only one Salvager can run at a time.

See Also

"salvager(8)" on page 879

Copyright

sysid

Name

`sysid` — Lists file server machine interface addresses registered in VLDB

Description

The *`sysid`* file records the network interface addresses that the File Server (fileserver process) registers in the Volume Location Database (VLDB) for the local file server machine.

Each time the File Server restarts, it builds a list of interfaces on the local machine by reading the */usr/afs/local/NetInfo* file, if it exists. If the file does not exist, the File Server uses the list of network interfaces configured with the operating system. It then removes from the list any addresses that appear in the */usr/afs/local/NetRestrict* file, if it exists. The File Server records the resulting list in the binary-format *`sysid`* file and registers the interfaces in the VLDB.

When the Cache Manager requests volume location information, the Volume Location (VL) Server provides all of the interfaces registered for each server machine that houses the volume. This enables the Cache Manager to make use of multiple addresses when accessing AFS data stored on a multihomed file server machine.

Cautions

The *`sysid`* file is unique to each file server machine, and must not be copied from one machine to another. If it is a common practice in the cell to copy the contents of the */usr/afs/local* directory from an existing file server machine to a newly installed one, be sure to remove the *`sysid`* file from the new machine before starting the `fs` trio of processes, which includes the fileserver process.

Some versions of AFS limit how many of a file server machine's interface addresses that can be registered. Consult the *IBM AFS Release Notes*.

See Also

"NetInfo(5)" on page 425, "NetRestrict(5)" on page 427, "vldb.DB0(5)" on page 519, "fileserver(8)" on page 771

Copyright

tapeconfig

Name

`tapeconfig` — Defines parameters for tape devices and backup data files

Description

The *`tapeconfig`* file defines basic configuration parameters for all of the tape devices or backup data files available for backup operations on a Tape Coordinator machine. The file is in ASCII format and must reside in the local */usr/afs/backup* directory. The instruction for each tape device or backup data file appears on its own line and each has the following format:

```
[&lt;capacity&gt; &lt;filemark_size&gt;] &lt;device_name&gt; &lt;port_offset&gt;
```

where

<capacity>

Specifies the capacity of the tapes used with a tape device, or the amount of data to write into a backup data file. The Tape Coordinator refers to this value in two circumstances:

- When the capacity field of a tape or backup data file's label is empty (because the tape has never been labeled). The Tape Coordinator records this value on the label and uses it when determining how much data it can write to the tape or file during a backup dump or backup savedb operation. If there is already a capacity value on the label, the Tape Coordinator uses it instead.
- When the -size argument is omitted the first time the backup labeltape command is used on a given tape or file. The Tape Coordinator copies this value into the label's capacity field.

The Tape Coordinator uses this capacity value or the one on the Backup System tape label to track how much space remains as it writes data to a tape or backup data file. The appropriate value to record for a tape depends on the size of the tapes usually used in the device and whether it has a compression mode; for suggested values, see the *IBM AFS Administration Guide* chapter on configuring the Backup System. If using a value obtained from the fms command, reduce it by 10% to 15% before recording it in the file.

For a backup data file, it is best to provide a value that helps the Tape Coordinator avoid reaching the end-of-file (EOF) unexpectedly. Make it at least somewhat smaller than the amount of space available on the partition housing the file when the dump operation begins, and never larger than the maximum file size allowed by the operating system.

Specify a (positive) integer or decimal value followed by a letter than indicates units, with no intervening space. In a decimal number, the number of digits after the decimal point must not translate to fractions of bytes. The maximum acceptable value is 2048 GB (2 TB). The acceptable units letters are as follows; if the letter is omitted, the default is kilobytes.

- `k` or `K` for kilobytes (KB).
- `m` or `M` for megabytes (MB).

- `g` or `G` for gigabytes (GB).
- `t` or `T` for terabytes (TB).

If this field is omitted, the Tape Coordinator uses the maximum acceptable value (2048 GB or 2 TB). Either leave both this field and the <filemark_size> field empty, or provide a value in both of them.

<filemark_size>

Specifies the size of a tape device's filemarks (also called end-of-file or EOF marks), which is set by the device's manufacturer. In a dump to tape, the Tape Coordinator inserts filemarks at the boundary between the data from each volume, so the filemark size affects how much space is available for actual data.

The appropriate value to record for a tape depends on the size of the tapes usually used in the device and whether it has a compression mode; for suggested values, see the *IBM AFS Administration Guide* chapter on configuring the Backup System. If using a value obtained from the fms command, increase it by 10% to 15% before recording it in the file.

For backup data files, record a value of 0 (zero). The Tape Coordinator actually ignores this field for backup data files, because it does not use filemarks when writing to a file.

Use the same notation as for the <capacity> field, but note that the default units is bytes rather than kilobytes. The maximum acceptable value is 2048 GB.

If this field is empty, the Tape Coordinator uses the value 0 (zero). Either leave both this field and the <capacity> field empty, or provide a value in both of them.

<device_name>

Specifies the complete pathname of the tape device or backup data file. The format of tape device names depends on the operating system, but on UNIX systems device names generally begin with the string `/dev/`. For a backup data file, this field defines the complete pathname; for a discussion of suggested naming conventions see the description of the `FILE` instruction in "butc(5)" on page 453.

<port_offset>

Specifies the port offset number associated with this combination of Tape Coordinator and tape device or backup data file.

Acceptable values are the integers `0` through `58510` (the Backup System can track a maximum of 58,511 port offset numbers). Each value must be unique among the cell's Tape Coordinators, but any number of them can be associated with a single machine. Port offset numbers need not be assigned sequentially, and can appear in any order in the *`tapeconfig`* file. Assign port offset `0` to the Tape Coordinator for the tape device or backup data file used most often for backup operations; doing so will allow the operator to omit the -portoffset argument from the largest possible number of backup commands.

Privilege Required

Creating the file requires UNIX `w` (write) and `x` (execute) permissions on the */usr/afs/backup* directory. Editing the file requires UNIX `w` (write) permission on the file.

Examples

The following example tapeconfig file configures three tape devices and a backup data file. The first device has device name */dev/rmt/0h*, and is assigned port offset `0` because it will be the most frequently used device for all backup operations in the cell. Its default tape capacity is 2 GB and filemark size is 1 MB. The */dev/rmt/3h* drive has half the capacity but a much smaller filemark size; its port offset is `3`. The third device listed, */dev/rmt/4h*, has the same capacity and filemark size as the first device and is assigned port offset `2`. Port offset `4` is assigned to the backup data file */dev/FILE*, which is actually a symbolic link to the actual file located elsewhere on the local disk. The Tape Coordinator writes up to 1.5 GB into the file; as recommended, the filemark size is set to zero.

```
2G 1M /dev/rmt/0h 0
1g 4k /dev/rmt/3h 3
2G 1m /dev/rmt/4h 2
1.5G 0 /dev/FILE 4
```

See Also

"backup_addhost(8)" on page 539, "backup_dump(8)" on page 571, "backup_labeltape(8)" on page 601, "backup_savedb(8)" on page 625, "butc(8)" on page 767, "fms(8)" on page 779

Copyright

uss

Name

uss — Provides instructions for the uss add command

Description

The uss template file defines the components of an AFS user account that the uss add command (or add instruction in a uss bulk input file) creates. Use the -template argument to the uss add or uss bulk command to identify the template file.

Summary of Template File Instructions

The template file can include the following instructions, each on its own line. A more detailed description of each instruction's syntax follows this list.

A

Imposes restrictions on user passwords and authentication attempts.

D

Creates a directory.

E

Creates a single-line file.

F

Creates a file by copying a prototype.

G

Defines a directory that is one of a set of parent directories into which the uss command interpreter evenly distributes newly created home directories.

L

Creates a hard link.

S

Creates a symbolic link.

V

Creates a volume, mounts it in the file space and sets the ACL on the mount point.

X

Executes a command.

If the template file is empty (zero-length), the uss add command or add instruction in a bulk input file only creates an entry in the Protection and Authentication Databases, naming them

according to the name specified with the uss add command's -user argument, or in the bulk input file add instruction's *username* field.

The A Instruction for Setting the Default Treatment of Volumes

The A instruction in a uss template file enhances cell security by imposing the following restrictions on users' password choice and authentication attempts. For further information on these limits, see the *IBM AFS Administration Guide* and the kas setfields reference page.

- Limiting the user's password lifetime. When the lifetime expires, the user can no longer authenticate using that password, and must change it.
- Prohibiting the reuse of the user's 20 most recently used passwords.
- Limiting the number of consecutive times that a user can provide an incorrect password during authentication, and for how long the Authentication Server refuses further authentication attempts after the limit is exceeded (referred to as an *account lockout*). For regular user accounts in most cells, the recommended limit is nine and lockout time is 25 minutes.

The instruction has the following syntax:

```
A &lt;username&gt; &lt;lifetime&gt; &lt;reuse&gt; &lt;failures&gt; &lt;locktime&gt;
```

where

A

Indicates a security-enhancing instruction. It must be a capital letter.

<username>

Names the Authentication Database entry on which to impose security restrictions. Specify the value $USER to read in the username from the uss add command's -user argument, or from the *username* field of an add instruction in a bulk input file.

<lifetime>

Sets the number of days after the user's password is changed that it remains valid. When the password becomes invalid (expires), the user is unable to authenticate, but has 30 more days in which to issue the kpasswd command to change the password (after that, only an administrator can change it).

Specify an integer from the range 1 through 254 to specify the number of days until expiration, the value 0 to indicate that the password never expires, or the value $PWEXPIRES to read in the number of days from the uss add or uss bulk command's -pwexpires argument. If the A instruction does not appear in the template file, the default is for the user's password never to expire.

<reuse>

Determines whether or not the user can change his or her password (using the kpasswd or kas setpassword command) to one that is similar to any of the last twenty passwords. The acceptable values are `reuse` to allow reuse and `noreuse` to prohibit it. If the `A` instruction does not appear in the template file, the default is to allow password reuse.

<failures>

Sets the number of consecutive times the user can provide an incorrect password during authentication (using the klog command or a login utility that grants AFS tokens). When the user exceeds the limit, the Authentication Server rejects further authentication attempts for the amount of time specified in the <locktime> field.

Specify an integer from the range `1` through `254` to specify the number of failures permitted, or the value `0` to indicate that there is no limit to the number of unsuccessful attempts. If the `A` instruction does not appear in the template file, the default is to allow an unlimited number of failures.

<locktime>

Specifies how long the Authentication Server refuses authentication attempts from a user who has exceeded the failure limit set in the <failures> field.

Specify a number of hours and minutes (*hh:mm*) or minutes only (*mm*), from the range `01` (one minute) through `36:00` (36 hours). The Authentication Server automatically reduces any larger value to `36:00` and also rounds up any non-zero value to the next higher multiple of 8.5 minutes. A value of `0` (zero) sets an infinite lockout time; an administrator must always issue the kas unlock command to unlock the account.

The D Instruction for Creating a Directory

The `D` instruction in a uss template file creates a directory. Its intended use is to create a subdirectory in the user home directory created by the `V` instruction in the template file.

Any number of `D` instructions can appear in the template file. If any variables in the instruction take their values from the `V` instruction (notably, the $MTPT variable), the instruction must follow the `V` instruction in the file.

Although it is possible to use the `D` instruction to create a directory on the local disk of the machine where the uss command is issued, it is not recommended. The preferred method for automated creation of directories on a local disk is the package program. Two complications arise if the <pathname> field refers to a local disk directory:

- The uss command prints a warning message because it cannot associate an access control list (ACL) with a local disk directory. It creates the directory nonetheless, and some syntactically correct value must appear in the instruction's <ACL> field.
- To designate any user other than the issuer as the new directory's owner, the issuer must log onto the machine as the local superuser `root`. For local disk directories, only the local superuser `root` is allowed to issue the UNIX chown command that the uss command interpreter invokes to change the owner from the default value (the directory's creator, which in this case is the issuer of the uss command). The issuer must then also use the

-admin argument to the uss add or uss bulk command to authenticate as a privileged AFS administrator, which is required for creating the Authentication Database and Protection Database entries that the uss command interpreter always creates for a new account.

The instruction has the following syntax:

```
D &lt;pathname&gt; &lt;mode&gt; &lt;owner&gt; &lt;ACL&gt;
```

where

D

Indicates a directory creation instruction. It must be a capital letter.

<pathname>

Specifies the directory's full pathname. It can include variables.

Specify the read / write path to the directory, to avoid the failure that results from attempting to create a new directory in a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). For further discussion of the concept of read / write and read-only paths through the filespace, see the reference page for the fs mkmount command.

<mode>

Sets the directory's UNIX mode bits. Acceptable values are the standard three- or four-digit numbers corresponding to combinations of permissions. Examples: `755` corresponds to `rwxr-xr-x`, and `644` to `rw-r--r--`. The first (owner) `x` bit must be turned on to enable access to a directory.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the directory's owner in the output from the UNIX `ls -ld` command. If the directory resides in AFS, place the $UID variable in this field. If the directory resides on the local disk, this field must be the username or UID of the uss command's issuer, unless the issuer is logged in as the local superuser `root`.

<ACL>

Sets the ACL on the new directory. It must appear even if the new directory resides on the local disk rather than in AFS, but is ignored in that case. Provide one or more paired values, each pair consisting of an AFS username or group name and the desired permissions, in that order. Separate the two parts of the pair, and each pair, with a space. The fs setacl reference page describes the available permissions.

For an AFS directory, grant all permissions to the directory's owner at least. Usually that is the new user, in which case the appropriate value is `$USER all`.

It is not possible to grant any permissions to the issuer of the uss command. As the last step in account creation, the uss command interpreter automatically deletes that person from any ACLs set during the creation process.

The E Instruction for Creating a Single-line File

The `E` instruction in a uss template file creates a file by echoing a specified character string into it. Its intended use is to create files in the user home directory created by the `V` instruction in the template file, or in a subdirectory created by a `D` instruction.

Any number of `E` instructions can appear in the template file. If the file resides in a directory created by a `D` instruction, the `E` instruction must follow the `D` instruction in the file.

The `E` and `F` instructions have complementary advantages. The character string echoed into the file by an `E` instruction can be customized for each user, because it can include the standard variables for which the uss command interpreter substitutes the values specified by arguments to the uss add command or fields in a bulk input file add instruction. In contrast, a file created using the `F` instruction cannot include variables and so has the same content for all users. However, a file created by an `E` instruction can be a single line only, because no carriage returns (newline characters) are allowed in the character string.

Although it is possible to use the `E` instruction to create a file on the local disk of the machine where the uss command is issued, it is not recommended. The preferred method for automated creation of files on a local disk is the package program. The main complication is that designating any user other than the issuer as the new file's owner requires logging onto the machine as the local superuser `root`. For local disk files, only the local superuser `root` is allowed to issue the UNIX chown command that the uss command interpreter invokes to change the owner from the default value (the file's creator, which in this case is the issuer of the uss command). The issuer must then also use the -admin argument to the uss add or uss bulk command to authenticate as a privileged AFS administrator, which is required for creating the Authentication Database and Protection Database entries that the uss command interpreter always creates for a new account.

The instruction has the following syntax:

```
E &lt;pathname&gt; &lt;mode&gt; &lt;owner&gt; "&lt;contents&gt;"
```

where

E

Indicates a file creation instruction. It must be a capital letter.

<pathname>

Specifies the file's full pathname. It can include variables.

Specify the read / write path to the file, to avoid the failure that results from attempting to create a new file in a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). For further discussion of the concept of read / write and read-only paths through the filespace, see the reference page for the fs mkmount command.

<mode>

Sets the file's UNIX mode bits. Acceptable values are the standard three- or four-digit numbers corresponding to combinations of permissions. Examples: `755` corresponds to

`rwxr-xr-x`, and `644` to `rw-r--r--`.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the file's owner in the output from the UNIX `ls -l` command. If the file resides in AFS, place the $UID variable in this field. If the file resides on the local disk, specify the username or UID of the uss command's issuer; otherwise, the account creation operation halts immediately.

<contents>

Specifies the one-line character string to write into the new file. Surround it with double quotes if it contains one or more spaces. It cannot contain the newline character, but can contain any of the standard variables, which the command interpreter resolves as it creates the file.

The F Instruction for Creating a File from a Prototype

The `F` instruction in a uss template file creates a file by copying the contents of an existing file (the <prototype>) into it. Its intended use is to create files in the user home directory created by the `V` instruction in the template file, or in a subdirectory created by a `D` instruction.

Any number of `F` instructions can appear in the template file. If the file resides in a directory created by a `D` instruction, the `F` instruction must follow the `D` instruction in the file.

The `E` and `F` instructions have complementary advantages. A file created using the `F` instruction has the same content for all users, whereas a file created by an `E` instruction can be customized for each user if it includes variables. However, a file created by an `E` instruction can be a single line only, whereas the prototype file copied by an `F` instruction can be any length.

Although it is possible to use the `F` instruction to create a file on the local disk of the machine where the uss command is issued, it is not recommended. The preferred method for automated creation of files on a local disk is the package program. The main complication is that designating any user other than the issuer as the new file's owner requires logging onto the machine as the local superuser `root`. For local disk files, only the local superuser `root` is allowed to issue the UNIX chown command that the uss command interpreter invokes to change the owner from the default value (the file's creator, which in this case is the issuer of the uss command). The issuer must then also use the -admin argument to the uss add or uss bulk command to authenticate as a privileged AFS administrator, which is required for creating the Authentication Database and Protection Database entries that the uss command interpreter always creates for a new account.

The instruction has the following syntax:

```
F &lt;pathname&gt; &lt;mode&gt; &lt;owner&gt; &lt;prototype_file&gt;
```

where

F

Indicates a file creation instruction. It must be a capital letter.

<pathname>

Specifies the full pathname of the file to create, including the filename. It can include variables.

Specify the read/write path to the file, to avoid the failure that results from attempting to create a new file in a read-only volume. By convention, the read/write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). For further discussion of the concept of read/write and read-only paths through the filespace, see the reference page for the fs mkmount command.

<mode>

Sets the file's UNIX mode bits. Acceptable values are the standard three- or four-digit numbers corresponding to combinations of permissions. Examples: `755` corresponds to `rwxr-xr-x`, and `644` to `rw-r--r--`.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the file's owner in the output from the UNIX `ls -l` command. If the file resides in AFS, place the $UID variable in this field. If the file resides on the local disk, specify the username or UID of the uss command's issuer; otherwise, the account creation operation halts immediately.

<prototype_file>

Names the AFS or local disk directory that houses the prototype file to copy. The prototype file's name must match the final element in the <pathname> field.

The G Instruction for Even Distribution of Home Directories

The `G` instruction in a uss template file creates a directory as one of the set of directories from which the uss command interpreter selects when choosing a new user home directory's parent directory. More specifically, when the $AUTO variable appears in the <mount_point> field of a `V` instruction, the command interpreter substitutes for it the directory defined by a `G` instruction that currently has the fewest entries.

The instruction's intended use is to distribute user accounts evenly among several directories, rather than using directories that reflect divisions such as departmental affiliation. Distributing home directories in this fashion is useful mainly in very large cells where storing all user home directories under a single parent directory potentially slows directory lookup, or where a workplace-based division results in unevenly sized directories such that some users consistently experience slower directory lookup than others. See the chapter on uss in the *IBM AFS Administration Guide* for more information.

Any number of `G` instructions can appear in the template file. If the `V` instruction includes the $AUTO variable, it must appear after all of the `G` instructions in the file.

The instruction has the following syntax:

```
G &lt;directory&gt;
```

where

G

Indicates an instruction that creates a directory to be considered as a value for the $AUTO variable. It must be a capital letter.

<directory>

Specifies the directory's name as either a complete pathname or only the directory name. The choice determines the appropriate format for the <mount_point> field of a `V` instruction, as discussed in the following example.

Specify the read / write path to the directory, to avoid the failure that results from attempting to create a new mount point in a read-only volume when the $AUTO variable is used in a `V` instruction's <mount_point> field. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, *`/afs/.abc.com`*). For further discussion of the concept of read / write and read-only paths through the filespace, see the reference page for the fs mkmount command.

The L and S Instructions for Creating a Link

The `L` instruction in a uss template file creates a hard link between two files, as achieved by the standard UNIX ln command. The `S` instruction creates a symbolic link between two files, as achieved by the standard UNIX `ln -s` command. A full explanation of links is beyond the scope of this document, but the basic effect is to create a second name for an existing file, enabling access via either name. Creating a link does not create a second copy of the file.

AFS allows hard links only if the linked files reside in the same directory, because it becomes difficult to determine which access control list (ACL) applies to the file if the two copies reside in directories with different ACLs. AFS allows symbolic links between two files that reside in different directories, or even different volumes. The File Server uses the ACL associated with the actual file rather than the link.

Any number of `L` and `S` instructions can appear in the template file. If the existing file or link is to reside in a directory created by a `D` instruction, or if the existing file was created by an `E` or `F` instruction, the `L` or `S` instruction must follow the `D`, `E`, or `F` instruction.

The instructions share the following syntax:

```
L &lt;existing_file&gt; &lt;link&gt;
S &lt;existing_file&gt; &lt;link&gt;
```

where

L

Indicates a hard link creation instruction. It must be a capital letter.

S

Indicates a symbolic link creation instruction. It must be a capital letter.

<existing_file>

Specifies the complete pathname of the existing file.

<link>

Specifies the complete pathname of the second name for the file.

Specify the read/write path to the link, to avoid the failure that results from attempting to create a new link in a read-only volume. By convention, the read/write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). For further discussion of the concept of read/write and read-only paths through the filespace, see the reference page for the fs mkmount command.

The V Instruction for Creating and Mounting a Volume

The `V` instruction in a uss template file creates a volume on a specified file server machine and partition and creates an entry for it in the Volume Location Database (VLDB). It mounts the volume at a location in the AFS file space that becomes the user's home directory, then designates the directory's owner and sets its access control list (ACL).

Only one `V` instruction can appear in the template file, and one must appear if the template file contains any instructions at all (is not empty). All other instructions are optional, except that the template must include `G` instructions if the $AUTO variable appears in it. (The `V` instruction is not necessarily the first line in the template. If the template includes the $AUTO variable, then the `G` instructions which provide values for the variable must precede it in the file.)

The instruction has the following syntax:

```
V &lt;vname&gt; &lt;server&gt; &lt;partition&gt; &lt;quota&gt; &lt;mount_point&gt; &lt;o
```

where

V

Indicates a volume creation instruction. It must be a capital letter.

<name>

Specifies the volume's name. To follow the convention for AFS user volume names, specify the value `user.$USER`. Provide a value for the $USER variable via the uss add command's -user argument or the <username> field in the bulk input file add instruction.

<server>

Names the file server machine on which to create the new user's volume. It is best to provide the fully-qualified hostname (for example, `fs1.abc.com`), but an abbreviated form is acceptable provided that the cell's naming service is available to resolve it at the time the volume is created. To read in the value from the uss add command's -server argument, specify the value $SERVER.

<partition>

Specifies the partition on which to create the user's volume; it must be on the file server machine named in the <server> field. Identify the partition by its complete name (for example, */vicepa*) or use or use one of the following abbreviations.

```
/vicepa     =     vicepa      =      a      =      0
/vicepb     =     vicepb      =      b      =      1
```

After */vicepz* (for which the index is 25) comes

```
/vicepaa    =     vicepaa     =      aa     =      26
/vicepab    =     vicepab     =      ab     =      27
```

and so on through

```
/vicepiv    =     vicepiv     =      iv     =      255
```

To read in the value from the uss add command's -partition argument, specify the value $PART.

<quota>

Sets the maximum number of kilobyte blocks the volume can occupy on the file server machine's disk. Specify an integer constant if all volumes have the same quota (`1024` equals a megabyte), or use one of the number variables ($1 through $9) to assign different values to different volumes.

<mount_point>

Creates a mount point for the volume, which serves as the volume's root directory. Include the $USER variable as part of the pathname to follow the convention that user home directory names include the username.

Specify the read/write path to the mount point, to avoid the failure that results from attempting to create a new mount point in a read-only volume. By convention, the read/write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). If the $AUTO variable appears in this field, the directories named by each `G` instruction possibly already indicate the read/write path. For further discussion of the concept of read/write and read-only paths through the filespace, see the reference page for the fs mkmount command.

<owner>

Specifies the username or UNIX user ID (UID) of the user to be designated the mount point's owner in the output from the UNIX `ls -ld` command. To follow the convention for home directory ownership, place the value $UID in this field.

<ACL>

Sets the ACL on the new directory. Provide one or more paired values, each pair consisting of an AFS username or group name and the desired permissions, in that order. Separate the two parts of the pair, and each pair, with a space. The fs setacl reference page describes the available permissions.

Grant all permissions to the new user at least. The appropriate value is `$USER all`.

AFS automatically grants the system:administrators group all permissions as well. It is not possible to grant any permissions to the issuer of the uss command. As the last step in account creation, the uss command interpreter automatically deletes that user from any ACLs set during the creation process.

The X Instruction for Running a Command

The `X` instruction in a uss template file runs the indicated command, which can be a standard UNIX or AFS command. It can include any variables from the template file, which the uss command interpreter resolves before passing the command on to the appropriate other command interpreter. It must be a single line only, however (cannot contain carriage returns or newline characters).

Any number of `X` instructions can appear in the template file. If an instruction manipulates an element created by another instruction, it must follow that instruction in the file.

The instruction has the following syntax:

```
X "&lt;command&gt;"
```

where

X

Indicates a command execution instruction. It must be a capital letter.

<command>

Specifies the command to run. Surround it with double quotes as shown if it contains one or more spaces. It can contain any variables from the template file, but not newline characters.

Examples

The following example A instruction sets a password lifetime of 254 days, prohibits password reuse, limits the number of consecutive failed authentication attempts to nine and sets the corresponding locktime to 25:30 minutes (which is a multiple of 8.5 minutes). The username is read in from the -user argument to the uss add command or from the *username* field in each `add` instruction in a bulk input file.

```
A $USER 254 noreuse 9 25:30
```

The following example `D` instruction creates a directory called *`public`* in a new user's home directory, designates the user as the directory's owner, and grants him or her all ACL permissions.

```
D $MTPT/public 0755 $UID $USER all
```

The following example `E` instruction creates a file in the current working directory called *username*`.etcp`. The contents are an entry suitable for incorporating into the cell's global */etc/password* file.

```
E  $USER.etcp  0644 root "$USER:X:$UID:10:$NAME:$MTPT:/bin/csh"
```

The following example `F` instruction, appropriate for the ABC Corporation cell, copies a prototype *.login* file into the user's home directory.

```
F $MTPT/.login 0644 $UID /afs/abc.com/common/uss/skel/.login
```

In the following example, the State University cell's administrators have decided to distribute user home directories evenly into three directories. They define three `G` instructions:

```
G usr1
G usr2
G usr3
```

and then put the following value in the <mount_point> field of the `V` instruction:

```
/afs/stateu.edu/$AUTO/$USER
```

Alternatively, if they include the entire directory pathname in the `G` instruction:

```
G /afs/stateu.edu/usr1
G /afs/stateu.edu/usr2
G /afs/stateu.edu/usr3
```

then the <mount_point> field of the `V` instruction specifies only the following:

```
$AUTO/$USER
```

The following example `L` instruction creates a hard link between the files *mail* and *mbox* in the user's home directory.

```
L $MTPT/mbox $MTPT/mail
```

The following example `S` instruction, appropriate for the ABC Corporation cell, links the file *Mail/outgoing* in the user's home directory to the file */afs/abc.com/common/mail/outgoing*.

```
S /afs/abc.com/common/mail/outgoing $MTPT/Mail/outgoing
```

The following example `V` instruction creates a volume called `user.`*username* on the */vicepa* partition of the specified file server machine, assigning it a quota of 3000 kilobyte blocks. The mount point is under */afs/abc.com/usr* and matches the username (the value of the $USER variable). The user owns the home directory and has all access rights to it. The instruction appears on two lines only for legibility; it must appear on a single line in the template file.

```
V user.$USER $SERVER.abc.com /vicepa 3000 \
        /afs/abc.com/usr/$USER $UID $USER all
```

The following example `X` instruction mounts the backup version of the user's volume at the *OldFiles* subdirectory.

```
X "fs mkm /afs/abc.com/usr/$USER/OldFiles   user.$USER.backup"
```

See Also

"uss_bulk(5)" on page 511, "fs_mkmount(1)" on page 105, "uss_add(8)" on page 895

Copyright

uss_bulk

Name

uss_bulk — Provides instructions for the uss bulk command

Description

The uss bulk input file lists instructions for the uss command interpreter to execute when running the uss bulk command. If the file includes add instructions that reference a uss template file, then the template file must also exist.

Summary of Bulk Input File Instructions

The bulk input file can include the following instructions, each on its own line. A more detailed description of each instruction's syntax follows this list.

add

Creates a user account. Equivalent to the uss add command.

delete

Deletes a user account. Equivalent to the uss delete command.

delvolume

Removes the volume and VLDB entry for each account referenced by a delete instruction that follows this instruction in the bulk input file.

exec

Executes a command.

savevolume

Preserves the volume and VLDB entry for each account referenced by a delete instruction that follows this instruction in the bulk input file.

The add Instruction for Creating an Account

The add instruction creates a user account. Each instance in the bulk input file is equivalent in effect to a uss add command issued on the command line. The order of the instruction's fields matches the order of arguments to the uss add command, although some arguments do not have a corresponding field. Like the uss add command's arguments, many of the fields correspond to (provide a value for) a variable in the uss template file, as indicated in the following description of each field.

The instruction's syntax is as follows. It appears on multiple lines here only for the sake of legibility -- each add instruction must appear on a single line in the bulk input file.

```
add &lt;username&gt;[:&lt;full_name&gt;][:&lt;password&gt;][:&lt;expires&gt;]
```

```
[:&lt;file_server&gt;][:&lt;partition&gt;][:&lt;mount_point&gt;][:&lt;uid&gt;]
[:&lt;var1&gt;][:&lt;var2&gt;][:&lt;var3&gt;][:&lt;var4&gt;][:&lt;var5&gt;][:&lt;var
[:&lt;var8&gt;][:&lt;var9&gt;][:]
```

To omit a value for a field (presumably because it is optional or the template specifies a constant value for it), type nothing between the two colons that surround it. After the last argument provided, end the line with either a colon and carriage return, or a carriage return alone.

The meaning of, and acceptable values for, each field are as follows.

<username>

Names the user's Authentication Database and Protection Database entries. It can include up to eight alphanumeric characters, but not the `:` (colon), `.` (period), or `@` (at-sign) characters. Because it becomes the username (the name under which a user logs in), it is best not to include shell metacharacters and to obey the restrictions that many operating systems impose on usernames (usually, to contain no more than eight lowercase letters).

Corresponding argument to the uss add command: -user. Corresponding variable in the template file: $USER.

<full_name>

Specifies the user's full name. Do not surround it with double quotes (""), even if it contains spaces. If not provided, it defaults to the username in the <username> field.

Corresponding argument to the uss add command: -realname. Corresponding variable in the template file: $NAME. Many operating systems include a field for the full name in a user's entry in the local password file (*/etc/passwd* or equivalent), and this variable can be used to pass a value to be used in that field.

<password>

Specifies the user's initial password. Although the AFS commands that handle passwords accept strings of virtually unlimited length, it is best to use a password of eight characters or less, which is the maximum length that many applications and utilities accept. If not provided, this argument defaults to the string `changeme`.

Corresponding argument to the uss add command: -pass. Corresponding variable in the template file: none.

<expires>

Sets the number of days after a user's password is changed that it remains valid. Provide an integer from the range `1` through `254` to specify the number of days until expiration, or the value `0` to indicate that the password never expires (the default).

When the password becomes invalid (expires), the user is unable to authenticate, but has 30 more days in which to issue the kpasswd command to change the password (after that, only an administrator can change it).

Corresponding argument to the uss add command: -pwexpires. Corresponding variable in the template file: $PWEXPIRES.

<file_server>

Names the file server machine on which to create the new user's volume. It is best to provide a fully-qualified hostname (for example, `fs1.abc.com`), but an abbreviated form is acceptable provided that the cell's naming service is available to resolve it at the time the volume is created.

Corresponding argument to the uss add command: -server. Corresponding variable in the template file: $SERVER.

<partition>

Specifies the partition on which to create the user's volume; it must reside on the file server machine named in the <file_server> field. Identify the partition by its complete name (for example, */vicepa*, or use one of the following abbreviations:

```
/vicepa     =     vicepa      =      a      =      0
/vicepb     =     vicepb      =      b      =      1
```

After */vicepz* (for which the index is 25) comes

```
/vicepaa    =     vicepaa     =      aa     =      26
/vicepab    =     vicepab     =      ab     =      27
```

and so on through

```
/vicepiv    =     vicepiv     =      iv     =      255
```

Corresponding argument to the uss add command: -partition. Corresponding variable in template: $PART.

<mount_point>

Specifies the complete pathname for the user's home directory.

Corresponding argument to the uss add command: -mount.

Corresponding variable in template: $MTPT, but in the template file's `V` instruction only. Occurrences of the $MTPT variable in template instructions that follow the `V` instruction take their value from the `V` instruction's <mount_point> field. Thus the value of this command line argument becomes the value for the $MTPT variable in instructions that follow the `V` instruction only if the string $MTPT appears alone in the `V` instruction's <mount_point> field.

<uid>

Specifies a positive integer other than `0` (zero) to assign as the user's AFS UID. If this argument is omitted, the Protection Server assigns an AFS UID that is one greater than the current value of the `max user id` counter (use the pts listmax command to display the counter). If including this argument, first use the pts examine command to verify that no existing account already has the desired AFS UID; if one does, the account-creation process terminates with an error.

Corresponding argument to the uss add command: -uid. Corresponding variable in template: $UID.

<var1> through <var9>

Specifies values for each of the number variables $1 through $9 that can appear in the template file. The number variables allow the administrator to provide values for variables other than the set defined by the uss command suite.

Corresponding argument to the uss add command: -var. Corresponding variables in template: $1 through $9.

If providing a value in any of the fields, then in every field that precedes it either provide an actual value or indicate an empty field by putting nothing between two colons. It is acceptable, but not necessary, to indicate empty fields by putting colons after the last field that contains an actual value.

The delete Instruction for Deleting an Account

The `delete` instruction deletes a user account from the system. Each instance in the bulk input file is equivalent in effect to a uss delete command issued on the command line. The order of the instruction's fields matches the order of arguments to the uss delete command:

```
delete &lt;username&gt;:&lt;mount_point&gt;[:( savevolume | delvolume )][:]
```

where

<username>

Names the entry to delete from the Protection and Authentication Databases.

<mount_point>

Specifies the complete pathname to the user's home directory, which is deleted from the filespace. By default, the volume mounted there is also deleted from the file server machine where it resides, as is its record from the Volume Location Database (VLDB). To prevent deletion, include the `savevolume` string in the instruction's third field, or precede this `delete` instruction with a `savevolume` instruction. Partial pathnames are interpreted relative to the current working directory.

savevolume

Retains the volume on its file server machine, and the corresponding entry in the VLDB. Provide this value or `delvolume` in the third field, or omit both values to treat the volume according to the prevailing default, which is set by a preceding `savevolume` or `delvolume` instruction in the bulk input file.

delvolume

Removes the volume from its file server machine, and the corresponding entry from the VLDB. Provide this value or `savevolume` in the third field, or omit both values to treat the volume according to the prevailing default, which is set by a preceding `savevolume` or `delvolume` instruction in the bulk input file.

After the last argument provided, end the line with either a colon and carriage return or a carriage return alone.

The exec Instruction for Executing a Command

The exec instruction executes the specified command, which can be a UNIX shell script or command, a program, or an AFS command. The uss command interpreter must have the necessary privileges in AFS and the local file system; it assumes the AFS and local identities of the issuer of the uss bulk command.

The instruction's syntax is as follows:

```
exec &lt;command&gt;
```

The delvolume and savevolume Instructions

The `savevolume` and `delvolume` instructions determine the default treatment of volumes referenced by the `delete` instructions that follow them in the bulk input file. Their syntax is as follows:

```
savevolume
delvolume
```

The `savevolume` instruction prevents the removal of the volume and VLDB entry for all `delete` instruction that follow it in the bulk input file, and the `delvolume` instruction removes the volume and VLDB entry for all subsequent `delete` instructions. Either setting persists until its opposite appears in the file, or until the end of the bulk file.

If neither line appears in the bulk input file, the default is to remove the volume and the VLDB entry; `delete` instructions that appear before the first `savevolume` instruction are also subject to this default. If a `delete` instruction's third field specifies either `savevolume` or `delvolume`, that setting overrides the default.

Examples

The following example add instruction creates an authentication-only account. The user's initial password is `changeme` (the default).

```
add anderson
```

The following example add instructions refer to the indicated `V` instruction in a template file (which must appear on a single line in the template file).

```
add smith:John Smith:::fs1:a:::::marketing
add jones:Pat Jones:::fs3:c:::::finance
V user.$USER $SERVER.abc.com /vicep$PART 2000 \
```

```
/afs/abc.com/usr/$3/$USER $UID $USER all
```

The first add instruction creates 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. The volume `user.smith` resides on partition */vicepa* of file server machine `fs1.abc.com` and is mounted at */afs/abc.com/usr/marketing/smith*. He owns his home directory and has all access permissions on its root directory's access control list (ACL). The account for `jones` is similar, except that the volume resides on partition */vicepc* of file server machine `fs3.abc.com` and is mounted at */afs/abc.com/usr/finance/jones*.

Notice that the fields corresponding to the volume mount point, UID, $1 variable, and $2 variable are empty (between `a` and `marketing` on the first example line), because their corresponding variables do not appear in the template file. The initial password field is also empty.

The following add instructions are equivalent in effect to the preceding example, but explicitly indicate empty fields for all of the number variables that don't have a value:

```
add smith:John Smith:::fs1:a:::::marketing::::::
add jones:Pat Jones:::fs3:c:::::finance::::::
```

The following example shows a complete bulk file containing a set of `delete` instructions combined with a `savevolume` instruction. Because the `delete` instruction for users `smith`, `pat`, and `rogers` appear before the `savevolume` instruction and the third field is blank in each, the corresponding home volumes are removed. The volume for user `terry` is retained because the default established by the `savevolume` instruction applies to it, but user `johnson`'s volume is removed because the third field of her `delete` instruction overrides the current default.

```
delete smith:/afs/abc.com/usr/smith
delete pat:/afs/abc.com/usr/pat
delete rogers:/afs/abc.com/usr/rogers
savevolume
delete terry:/afs/abc.com/usr/terry
delete johnson:/afs/abc.com/usr/johnson:delvolume
```

The following example exec instruction appears between sets of `add` and `delete` instructions in a bulk input file. A message appears in the command shell where the uss bulk command is issued, to indicate when the additions are finished and the deletions beginning.

```
exec echo "Additions completed; beginning deletions..."
```

See Also

"uss(5)" on page 497, "uss_add(8)" on page 895, "uss_bulk(8)" on page 903, "uss_delete(8)" on page 907

Copyright

vldb.DB0, vldb.DBSYS1

Name

`vldb.DB0, vldb.DBSYS1` — Contain the Volume Location Database and associated log

Description

The file *`vldb.DB0`* contains the Volume Location Database (VLDB), which tracks the location of all AFS volumes stored on file server machines in the cell. The Volume Location (VL) Server (vlserver process) provides information from the database to Cache Managers when they need to access AFS data.

The file *`vldb.DBSYS1`* is a log file in which the VL Server logs each database operation before performing it. When an operation is interrupted, the VL Server replays the log to complete the operation.

Both files are in binary format and reside in the */usr/afs/db* directory on each of the cell's database server machines. When the VL Server starts or restarts on a given machine, it establishes a connection with its peers and verifies that its copy of the database matches the copy on the other database server machines. If not, the VL Servers call on AFS's distributed database technology, Ubik, to distribute to all of the machines the copy of the database with the highest version number.

Always use the commands in the vos suite to administer the VLDB. It is advisable to create an archive copy of the database on a regular basis, using a tool such as the UNIX tar command.

See Also

"vldb_check(8)" on page 913, "vlserver(8)" on page 915, "vos(1)" on page 289

Copyright

III. AFS System Commands

afsd

Name

`afsd` — Initializes the Cache Manager and starts related daemons

Synopsis

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] [-shutdown] [-enable_peer_stats] [-enable_process_stats] [-help]

Description

The afsd command initializes the Cache Manager on an AFS client machine by transferring AFS-related configuration information into kernel memory and starting several daemons. More specifically, the afsd command performs the following actions:

- Sets a field in kernel memory that defines the machine's cell membership. Some Cache Manager-internal operations and system calls consult this field to learn which cell to execute in. (The AFS command interpreters refer to the `/usr/vice/etc/ThisCell` file instead.) This information is transferred into the kernel from the `/usr/vice/etc/ThisCell` file and cannot be changed until the afsd program 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.

  The list of database server machines is transferred into the kernel from the `/usr/vice/etc/CellServDB` file. After initialization, use the fs newcell command to change the kernel-resident list without having to reboot.
- Mounts the root of the AFS filespace on a directory on the machine's local disk, according to either the first field in the `/usr/vice/etc/cacheinfo` file (the default) or the afsd command's -mountdir argument. The conventional value is `/afs`.
- Determines which volume to mount at the root of the AFS file tree. The default is the volume `root.afs`; use the -rootvol argument to override it. Although the base (read / write) form of the volume name is the appropriate value, the Cache Manager has a bias for accessing the read-only version of the volume (by convention, `root.afs.readonly`) if it is available.
- Configures the cache on disk (the default) or in machine memory if the -memcache argument is provided. In the latter case, the afsd program allocates space in machine

memory for caching, and the Cache Manager uses no disk space for caching even if the machine has a disk.

- Defines the name of the local disk directory devoted to caching, when the -memcache argument is not used. If necessary, the afsd program creates the directory (its parent directory must already exist). It does not remove the directory that formerly served this function, if one exists.

  The second field in the */usr/vice/etc/cacheinfo* file is the source for this name, and the standard value is the */usr/vice/cache* directory. Use the -cachedir argument to override the value in the cacheinfo file.

- Sets the size of the cache. The default source for the value is the third field in the */usr/vice/etc/cacheinfo* file, which specifies a number of kilobytes.

  For a memory cache, the following arguments to the afsd command override the value in the cacheinfo file:

  - The -blocks argument, to specify a different number of kilobyte blocks.
  - The -dcache and -chunksize arguments together, to set both the number of dcache entries and the chunk size (see below for definition of these parameters). In this case, the afsd program 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.
  - The -dcache argument by itself. In this case, the afsd program derives cache size by multiplying the value specified by the -dcache argument by the default memory cache chunk size of eight kilobytes. Using this argument is not recommended, as it requires the issuer to perform the calculation beforehand to determine the resulting cache size.

  For satisfactory memory cache performance, the specified value must leave enough memory free to accommodate all other processes and commands that can run on the machine. If the value exceeds the amount of memory available, the afsd program exits without initializing the Cache Manager and produces the following message on the standard output stream:

  ```
  afsd: memCache allocation failure at &lt;number&gt; KB
  ```

  where <number> is how many kilobytes were allocated just before the failure.

  For a disk cache, use the -blocks argument to the afsd command to override the value in the cacheinfo file. The value specified in either way sets an absolute upper limit on cache size; values provided for other arguments (such as -dcache and -chunksize) never result in a larger cache. The afsd program rejects any setting larger than 95% of the partition size, and exits after generating an error message on the standard output stream, because the cache implementation itself requires a small amount of disk space and overfilling the partition can cause the client machine to panic.

  To change the size of a disk cache after initialization without rebooting, use the fs setcachesize command; the setting persists until the afsd command runs again or the fs setcachesize command is reissued. The fs setcachesize command does not work for memory caches.

- Sets the size of each cache *chunk*, 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, a chunk is a `Vn` file and this parameter sets the maximum size to which each one can expand; the default is 64 KB. For a memory cache, each chunk is a collection of contiguous memory blocks; the default is size is 8 KB.

To override the default chunk size for either type of cache, use the -chunksize argument to provide an integer to be used as an exponent of two; see "OPTIONS" on page for details. For a memory cache, if total cache size divided by chunk size leaves a remainder, the afsd program rounds down the number of dcache entries, resulting in a slightly smaller cache.

- Sets the number of chunks in the cache. For a memory cache, the number of chunks is equal to the cache size divided by the chunk size. For a disk cache, the number of chunks (`Vn` files) is set to the largest of the following unless the -files argument is used to set the value explicitly:
  - 100
  - 1.5 times the result of dividing cache size by chunk size (*cachesize* / *chunksize* * 1.5)
  - The result of dividing cachesize by 10 KB (*cachesize* / 10240)

- Sets the number of *dcache entries* allocated in machine memory for storing information about the chunks in the cache.

  For a disk cache, the `/usr/vice/cache/CacheItems` file contains one entry for each `Vn` file. By default, one half the number of these entries (but not more that 2,000) 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, the afsd program derives it by dividing the cache size by the chunk size.

  To set the number of dcache entries, use the -dcache argument; the specified value can exceed the default limit of 2,000. Using this argument is not recommended for either type of cache. Increasing the number of dcache entries for a disk cache sometimes improves performance (because more entries are retrieved from memory rather than from disk), but only marginally. Using this argument for a memory cache requires the issuer to calculate the cache size by multiplying this value by the chunk size.
- 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 the afsd command 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, the afsd program starts the following daemons. (On most system types, these daemons appear as nameless entries in the output of the UNIX ps command.)

- One *callback* daemon, which handles callbacks. It also responds to the File Server's periodic probes, which check that the client machine is still alive.
- One *maintenance* daemon, which performs the following tasks:
  - Garbage collects obsolete data (for example, expired tokens) from kernel memory.
  - Synchronizes files.
  - Refreshes information from read-only volumes once per hour.
  - Does delayed writes for NFS clients if the machine is running the NFS/AFS Translator.
- One *cache-truncation* daemon, which flushes the cache when free space is required, by writing cached data and status information to the File Server.
- One *server connection* daemon, which sends a probe to the File Server every few minutes to check that it is still accessible. It also synchronizes the machine's clock with the clock on a randomly-chosen file server machine, unless the -nosettime flag is used. There is always one server connection daemon.
- One or more *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 two, enough to service at least five simultaneous users of the machine. To increase the number, use the -daemons argument. A value greater than six is not generally necessary.
- On some system types, one *Rx listener* daemon, which listens for incoming RPCs.
- On some system types, one *Rx event* daemon, which reviews the Rx system's queue of tasks and performs them as appropriate. Most items in the queue are retransmissions of failed packets.
- On machines that run AIX with virtual memory (VM) integration, one or more *VM* daemons (sometimes called *I/O* daemons, which transfer data between disk and machine memory. The number of them depends on the setting of the -biods and -daemons arguments:
  - If the -biods argument is used, it sets the number of VM daemons.
  - If only the -daemons argument is used, the number of VM daemons is twice the number of background daemons.
  - If neither argument is used, there are five VM daemons.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Cautions

Do not use the -shutdown parameter. It does not shutdown the Cache Manager effectively. Instead, halt Cache Manager activity by using the standard UNIX umount command to unmount the AFS root directory (by convention, `/afs`). The machine must then be rebooted to reinitialize the Cache Manager.

Options

-blocks <*blocks in cache*>

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 the `/usr/vice/etc/cacheinfo` file. For a disk cache, the value cannot exceed 95% of the space available in the cache partition. If using a memory cache, do not combine this argument with the -dcache argument, since doing so can possibly result in a chunk size that is not an exponent of 2.

-files <*files in cache*>

Specifies the number of `Vn` files to create in the cache directory for a disk cache, overriding the default that is calculated as described in "DESCRIPTION" on page . Each `Vn` file accommodates a chunk of data, and can grow to a maximum size of 64 KB by default. Do not combine this argument with the -memcache argument.

-rootvol <*name of AFS root volume*>

Names the read / write volume corresponding to the root directory for the AFS file tree (which is usually the `/afs` directory). This value overrides the default of the `root.afs` volume.

-stat <*number of stat entries*>

Specifies the number of entries to allocate in the machine's memory for recording status information about the AFS files in the cache. This value overrides the default of `300`.

-memcache

Initializes a memory cache rather than a disk cache. Do not combine this flag with the -files argument.

-cachedir <*cache directory*>

Names the local disk directory to be used as the cache. This value overrides the default defined in the second field of the `/usr/vice/etc/cacheinfo` file.

-mountdir <*mount location*>

Names the local disk directory on which to mount the root of the AFS filespace. This value overrides the default defined in the first field of the `/usr/vice/etc/cacheinfo` file. If a value other than the `/afs` directory is used, the machine cannot access the filespace of cells that do use that value.

-daemons <*number of daemons to use*>

Specifies the number of background daemons to run on the machine. These daemons improve efficiency by doing prefetching and background writing of saved data. This value overrides the default of `2`, which is adequate for a machine serving up to five users. Values greater than `6` are not generally more effective than `6`.

Note: On AIX machines with integrated virtual memory (VM), the number of VM daemons is set to twice the value of this argument, if it is provided and the -biods argument is not. If both arguments are omitted, there are five VM daemons.

-nosettime

Prevents the Cache Manager from synchronizing its clock with the clock on a server machine selected at random, by checking the time on the server machine every five minutes. Use this flag only on a machine that is already using another time synchronization protocol (for example, a server machine that is running the runntp process).

-verbose

Generates a detailed trace of the afsd program's actions on the standard output stream.

-rmtsys

Initializes an additional daemon to execute AFS-specific system calls on behalf of NFS client machines. Use this flag only if the machine is an NFS/AFS translator machine serving users of NFS client machines who execute AFS commands.

-debug

Generates a highly detailed trace of the afsd program's actions on the standard output stream. The information is useful mostly for debugging purposes.

-chunksize <*chunk size*>

Sets the size of each cache chunk. The integer provided, which must be from the range `0` to `30`, is used as an exponent on the number 2. It overrides the default of 16 for a disk cache (2^16 is 64 KB) and 13 for a memory cache (2^13 is 8 KB). 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 1 KB) are not recommended. Combining this argument with the -dcache argument is not recommended because it requires that the issuer calculate the cache size that results.

-dcache <*number of dcache entries*>

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, which is 50% of the number of *Vn* files (cache chunks). For a memory cache, this argument effectively sets the number of cache chunks, but its use is not recommended, because it requires the issuer to calculate the resulting total cache size (derived by multiplying this value by the chunk size). Do not combine this argument with the -blocks argument, since doing so can possibly result in a chunk size that is not an exponent of 2.

-volumes <*number of volume entries*>

Specifies the number of memory structures to allocate for storing volume location information. The default value is `50`.

-biods <*number of I/O daemons*>

Sets the number of VM daemons dedicated to performing I/O operations on a machine running a version of AIX with virtual memory (VM) integration. If both this argument and

the -daemons argument are omitted, the default is five. If this argument is omitted but the -daemons argument is provided, the number of VM daemons is set to twice the value of the -daemons argument.

-prealloc <*number of preallocated blocks*>

Specifies the number of pieces of memory to preallocate for the Cache Manager's internal use. The default initial value is `400`, but the Cache Manager dynamically allocates more memory as it needs it.

-confdir <*configuration directory*>

Names a directory other than the `/usr/vice/etc` directory from which to fetch the `cacheinfo`, `ThisCell`, and `CellServDB` configuration files.

-logfile <*log file location*>

Is obsolete and has no real effect. It specifies an alternate file in which to record a type of trace that the Cache Manager no longer generates; the default value is `/usr/vice/etc/AFSLog`.

-waitclose

Has no effect on the operation of the Cache Manager. The behavior it affected in previous versions of the Cache Manager, to perform synchronous writes to the File Server, is now the default behavior. To perform asynchronous writes in certain cases, use the fs storebehind command.

-shutdown

Shuts down the Cache Manager, but not in the most effective possible way. Do not use this flag.

-enable_peer_stats

Activates the collection of Rx statistics and allocates memory for their storage. For each connection with a specific UDP port on another machine, a separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received. To display or otherwise access the records, use the Rx Monitoring API.

-enable_process_stats

Activates the collection of Rx statistics and allocates memory for their storage. A separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received, aggregated over all connections to other machines. To display or otherwise access the records, use the Rx Monitoring API.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The afsd command is normally included in the machine's AFS initialization file, rather than typed at the command shell prompt. For most disk caches, the appropriate form is

```
/usr/vice/etc/afsd
```

The following command 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 command initializes a memory cache and sets chunk size to 16 KB (2^14).

```
/usr/vice/etc/afsd -memcache -chunksize 14
```

Privilege Required

The issuer must be logged in as the local superuser root.

See Also

"afs_cache(5)" on page 443, "CellServDB(5)" on page 415, "cacheinfo(5)" on page 467

Copyright

backup

Name

backup — Introduction to the backup command suite

Description

The commands in the backup command suite are the administrative interface to the AFS Backup System. There are several categories of commands in the suite:

- Commands to copy data from AFS volumes to tape or a backup data file, and to restore it to the file system: backup diskrestore, backup dump, backup volrestore, and backup volsetrestore.
- Commands to administer the records in the Backup Database: backup adddump, backup addhost, backup addvolentry, backup addvolset, backup deldump, backup deletedump, backup delhost, backup delvolentry, backup delvolset, backup dumpinfo, backup listdumps, backup listhosts, backup listvolsets, backup scantape, backup setexp, and backup volinfo.
- Commands to write and read tape labels: backup labeltape and backup readlabel.
- Commands to list and change the status of backup operations and the machines performing them: backup jobs, backup kill, and backup status.
- Commands to enter and leave interactive mode: backup interactive and backup quit.
- Commands to check for and repair corruption in the Backup Database: backup dbverify, backup restoredb, and backup savedb.
- Commands to obtain help: backup apropos and backup help.

The backup command interpreter interacts with two other processes:

- The Backup Server (buserver) process. It maintains the Backup Database, which stores most of the administrative information used by the Backup System. In the standard configuration, the Backup Server runs on each database server machine in the cell, and uses AFS's distributed database technology, Ubik, to synchronize its copy of the database with the copies on the other database server machines.
- The Backup Tape Coordinator (butc) process. A separate instance of the process controls each tape device or backup data file used to dump or restore data. The Tape Coordinator runs on a Tape Coordinator machine, which is an AFS server or client machine that has one or more tape devices attached, or has sufficient disk space to accommodate one or more backup data files on its local disk.

  Each Tape Coordinator must be registered in the Backup Database and in the `/usr/afs/backup/tapeconfig` configuration file on the Tape Coordinator machine's local disk, and information in the two places must be consistent for proper Backup System performance. The optional `/usr/afs/backup/CFG_`*device_name* for each Tape Coordinator records information used to automate its operation.

In addition to the standard command line interface, the backup command suite provides an *interactive* interface, which has several useful features described in "backup_interactive(8)" on page 591. Three of the commands in the suite are available only in interactive mode: backup jobs, backup kill, and backup quit.

Options

The following options are available on many commands in the backup suite. The reference page for each command also lists them, but they are described here in greater detail.

-cell <*cell name*>

Names the cell in which to run the command. It is acceptable to abbreviate the cell name to the shortest form that distinguishes it from the other entries in the */usr/vice/etc/CellServDB* file on the local machine. If the -cell argument is omitted, the command interpreter determines the name of the local cell by reading the following in order:

- The value of the AFSCELL environment variable.
- The local */usr/vice/etc/ThisCell* file.

Do not combine the -cell and -localauth options. A command on which the -localauth flag is included always runs in the local cell (as defined in the server machine's local */usr/afs/etc/ThisCell* file), whereas a command on which the -cell argument is included runs in the specified foreign cell.

The -cell argument is not available on commands issued in interactive mode. The cell defined when the backup command interpreter enters interactive mode applies to all commands issued during the interactive session.

-help

Prints a command's online help message on the standard output stream. Do not combine this flag with any of the command's other options; when it is provided, the command interpreter ignores all other options, and only prints the help message.

-localauth

Constructs a server ticket using the server encryption key with the highest key version number in the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents the ticket, which never expires, to the Backup Server, Volume Server and Volume Location (VL) Server during mutual authentication.

Use this flag only when issuing a command on a server machine; client machines do not usually have a */usr/afs/etc/KeyFile* file. The issuer of a command that includes this flag must be logged on to the server machine as the local superuser `root`. The flag is useful for commands invoked by an unattended application program, such as a process controlled by the UNIX cron utility or by a cron entry in the machine's */usr/afs/local/BosConfig* file. It is also useful if an administrator is unable to authenticate to AFS but is logged in as the local superuser `root`.

Do not combine the -cell and -localauth options. A command on which the -localauth flag is included always runs in the local cell (as defined in the server machine's local */usr/afs/etc/ThisCell* file), whereas a command on which the -cell argument is included runs in the specified foreign cell.

The -localauth argument is not available on commands issued in interactive mode. The local identity and AFS tokens with which the backup command interpreter enters interactive mode apply to all commands issued during the interactive session.

-portoffset <*TC port offset*>

Specifies the port offset number of the Tape Coordinator that is to execute the backup command. The port offset number uniquely identifies a pairing of a Tape Coordinator (butc) process and tape device or backup data file.

The backup command interpreter and Tape Coordinator process communicate via a UDP socket, or port. Before issuing a backup command that involves reading or writing a tape, the backup operator must start a butc process that controls the appropriate tape device and listens for requests sent to its port number. If a Backup System machine has multiple tape devices attached, they can perform backup operations simultaneously because each device has its own associated butc process and port offset number.

The Backup System associates a tape capacity and file mark size with each port offset (as defined in the *tapeconfig* file). For a compressing tape device, the capacity and file mark values differ for compression and non-compression modes, so the two modes have distinct port offset numbers.

The Backup Database can store up to 58,511 port offsets, so the legal values for this argument are the integers `0` through `58510`. If the issuer omits the argument, it defaults to `0`. (The limit of 58,511 port offsets results from the fact that UDP socket numbers are identified by a 16-bit integer, and the lowest socket number used by the Backup System is 7025. The largest number that a 16-bit integer can represent is 65,535. Subtracting 7,025 yields 58,510. The addition of port offset 0 (zero) increases the maximum to 58,511.)

Although it is possible to define up to 58,511 port offset numbers for a cell, it is not possible to run 58,511 tape devices simultaneously, due to the following limits:

- The maximum number of dump or restore operations that can run simultaneously is 64.
- The maximum number of tape devices that can work together on a restore operation is 128 (that is the maximum number of values that can be provided for the -portoffset argument to the backup diskrestore, backup volrestore, or backup volsetrestore command).

The Backup System does not reserve UDP sockets. If another application is already using the Tape Coordinator's socket when it tries to start, the butc process fails and the following error message appears at the shell prompt:

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

Privilege Required

To issue any backup command that accesses the Backup Database only, the issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running. To issue any backup command that accesses volume data, the issuer must appear in the *UserList* file on every Backup Server machine, every Volume Location (VL) Server machine, and every file server machine that houses affected volumes. By convention, a common *UserList* file is distributed to all database server and file server machines in the cell. See the chapter on privileged users in the *IBM AFS Administration Guide* for more information on this type of privilege.

If the -localauth flag is included, the user must instead be logged on as the local superuser `root` on the server machine where the backup command is issued.

See Also

"BosConfig(5)" on page 409, "CellServDB(5)" on page 415, "KeyFile(5)" on page 423, "ThisCell(5)" on page 435, "UserList(5)" on page 437, "butc(5)" on page 453, "tapeconfig(5)" on page 493, "backup_adddump(8)" on page 535, "backup_addhost(8)" on page 539, "backup_addvolentry(8)" on page 541, "backup_addvolset(8)" on page 545, "backup_dbverify(8)" on page 549, "backup_deldump(8)" on page 553, "backup_deletedump(8)" on page 555, "backup_delhost(8)" on page 559, "backup_delvolentry(8)" on page 561, "backup_delvolset(8)" on page 563, "backup_diskrestore(8)" on page 565, "backup_dump(8)" on page 571, "backup_dumpinfo(8)" on page 581, "backup_help(8)" on page 589, "backup_interactive(8)" on page 591, "backup_jobs(8)" on page 593, "backup_kill(8)" on page 597, "backup_labeltape(8)" on page 601, "backup_listdumps(8)" on page 605, "backup_listhosts(8)" on page 609, "backup_listvolsets(8)" on page 611, "backup_quit(8)" on page 613, "backup_readlabel(8)" on page 615, "backup_restoredb(8)" on page 621, "backup_savedb(8)" on page 625, "backup_scantape(8)" on page 629, "backup_setexp(8)" on page 635, "backup_status(8)" on page 639, "backup_volinfo(8)" on page 643, "backup_volrestore(8)" on page 647, "backup_volsetrestore(8)" on page 653, "buserver(8)" on page 763, "butc(8)" on page 767

Copyright

backup adddump

Name

`backup adddump` — Defines a dump level in the dump hierarchy

Synopsis

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

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

Description

The backup adddump command creates one or more dump levels in the dump hierarchy stored in the Backup Database, and optionally assigns an expiration date to each one. All of the dump levels in the Backup Database collectively constitute the dump hierarchy.

Use the -expires argument to associate an expiration date with each dump level. When the Backup System subsequently creates a dump at the dump level, it uses the specified value to derive the dump's expiration date, which it records on the label of the tape (or backup data file). The Backup System refuses to overwrite a tape until after the latest expiration date of any dump that the tape contains, unless the backup labeltape command is used to relabel the tape. If a dump level does not have an expiration date, the Backup System treats dumps created at the level as expired as soon as it creates them.

(Note that the Backup System does not automatically remove a dump's record from the Backup Database when the dump reaches its expiration date, but only if the tape that contains the dump is recycled or relabeled. To remove expired and other obsolete dump records, use the backup deletedump command.)

Define either an absolute or relative expiration date:

- An absolute expiration date defines the month / day / year (and, optionally, hour and minutes) at which a dump expires. If the expiration date predates the dump creation time, the Backup System immediately treats the dump as expired.
- A relative date defines the number of years, months, or days (or a combination of the three) after the dump's creation that it expires. When the Backup System creates a dump at the dump level, it calculates an actual expiration date by adding the relative date to the start time of the dump operation.

Options

-dump <*dump level name*>+

Names each dump level to add to the dump hierarchy. Precede full dump level names with a slash (for example, `/full`). Indicate an incremental dump level by preceding it with

an ordered list of the dump levels directly above it in the hierarchy (its parent dump levels); use the slash as a separator. The parent dump levels must already exist. For example, the dump levels `/full` and `/full/incremental1` must exist when the incremental dump level `/full/incremental1/incremental2` is created.

Dump level names can have any number of levels, but cannot exceed 256 characters in length, including the slashes. The maximum length for any single level (the text between slashes) is 28 characters, not including the preceding slash.

All alphanumeric characters are allowed in dump level names. Do not use the period (`.`), however, because it is the separator between the volume set name and dump level name in the dump name assigned automatically by the backup dump command. It is best not to include other metacharacters either; if using them, enclose them in double quotes (`" "`) when issuing the backup adddump command outside interactive mode.

-expires <*expiration date*>+

Defines the absolute or relative expiration date to associate with each dump level named by the -dump argument. Absolute expiration dates have the following format:

```
[at] {NEVER | &lt;mm&gt;/&lt;dd&gt;/&lt;yyyy&gt; [&lt;hh&gt;:&lt;MM&gt;] }
```

where the optional word at is followed either by the string `NEVER`, which indicates that dumps created at the dump level never expire, or by a date value with a required portion (<mm> for month, <dd> for day, and <yyyy> for year) and an optional portion (<hh> for hours and <MM> for minutes).

Omit the *hh:MM* portion to use the default of midnight (00:00 hours), or provide a value in 24-hour format (for example, `20:30` is 8:30 p.m.). Valid values for the year range from `1970` to `2037`; higher values are not valid because the latest possible date in the standard UNIX representation is in February 2038. The command interpreter automatically reduces later dates to the maximum value.

Relative expiration dates have the following format:

```
[in] [&lt;years&gt;y] [&lt;months&gt;m] [&lt;days&gt;d]
```

where the optional word in is followed by at least one of a number of years (maximum `9999`) followed by the letter `y`, a number of months (maximum `12`) followed by the letter `m`, or a number of days (maximum `31`) followed by the letter `d`. If providing more than one of the three, list them in the indicated order. If the date that results from adding the relative expiration value to a dump's creation time is later than the latest possible date in the UNIX time representation, the Backup System automatically reduces it to that date.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command defines a full dump called `/1999` with a relative expiration date of one year:

```
% backup adddump -dump /1999 -expires in 1y
```

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

```
% backup adddump -dump /sunday1/monday1 -expires in 13d
```

The following command defines two dump incremental dump levels, `/Monthly/Week1` and `/Monthly/Week2`. Their parent, the full dump level `/Monthly`, must already exist. The expiration date for both levels is 12:00 a.m. on 1 January 2000.

```
% backup adddump -dump /Monthly/Week1 /Monthly/Week2 -expires at 01/01/2000
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_deldump(8)" on page 553, "backup_deletedump(8)" on page 555, "backup_listdumps(8)" on page 605, "backup_setexp(8)" on page 635

Copyright

backup addhost

Name

`backup addhost` — Adds a Tape Coordinator entry to the Backup Database

Synopsis

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

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

Description

The backup addhost command creates a Tape Coordinator entry in the Backup Database. The entry records

- The host name of the Tape Coordinator machine where the Tape Coordinator (butc) process runs, as specified with the -tapehost argument.
- The Tape Coordinator's port offset number, as specified with the -portoffset argument. An entry for the port offset must also appear in the `/usr/afs/backup/tapeconfig` file on the Tape Coordinator machine, where it is mapped to a UNIX device name (for a tape device) or pathname (for a backup data file).

Each Tape Coordinator must have its own port offset number, and the command fails if a Backup Database entry already exists for the requested port offset number. To display existing Tape Coordinator entries, use the backup listhosts command.

Options

-tapehost <*tape machine name*>

Specifies the fully-qualified hostname of the machine for which to create a Tape Coordinator entry in the Backup Database. The machine must have an entry in either the cell's naming service (such as the Domain Name Service) or the host file (`/etc/hosts` or equivalent) on the machine where the command is issued.

-portoffset <*TC port offset*>

Specifies the Tape Coordinator's port offset number. Provide an integer from the range `0` through `58510`, or omit this argument to use the default value of `0` (zero). The value must match the port offset number recorded for the same combination of Tape Coordinator and tape device or file in the `/usr/afs/backup/tapeconfig` file on the Tape Coordinator machine named by the -tapehost argument.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command creates an entry in the Backup Database that assigns port offset number 4 to a Tape Coordinator running on the machine `backup1.abc.com`:

```
% backup addhost -tapehost backup1.abc.com -portoffset 4
```

The following command creates a Backup Database entry that assigns port offset number 0 to a Tape Coordinator on the machine `backup3.abc.com`:

```
% backup addhost backup3.abc.com
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_delhost(8)" on page 559, "backup_listhosts(8)" on page 609

Copyright

backup addvolentry

Name

`backup addvolentry` — Defines a volume entry in a volume set

Synopsis

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

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

Description

The backup addvolentry command adds a volume entry definition to the existing volume set named by the -name argument. A volume entry definition can match one or more volumes, depending on the combination of the -server, -partition, and -volumes arguments.

For the -server and -partition arguments, provide either

- The name of one machine or partition.
- The metacharacter expression .* (period and asterisk), which matches every machine name or partition name in the Volume Location Database (VLDB).

For the -volumes argument, specify a combination of alphanumeric characters and one or more metacharacters to wildcard part or all of the volume name. "OPTIONS" on page lists the acceptable metacharacters.

Cautions

It is best to issue this command in interactive mode. If issuing it at the shell prompt, enclose any strings containing metacharacters in double quotes, or escape the metacharacters with other delimiters, to prevent the shell from interpreting them. Adding volume entries to a temporary volume set is possible only within the interactive session in which the volume set was created.

Options

-name <*volume set name*>

Names the volume set to which to add this volume entry definition. The volume set must already exist (use the backup addvolset command to create it).

-server <*machine name*>

Defines the set of one or more file server machines that house the volumes in the volume entry. Provide either one fully-qualified hostname (such as `fs1.abc.com`) or the

metacharacter expression `.*` (period and asterisk), which matches all machine names in the VLDB.

-partition <*partition name*>

Defines the set of one or more partitions that house the volumes in the volume entry. Provide either one complete partition name (such as `/vicepa`) or the metacharacter expression `.*` (period and asterisk), which matches all partition names.

-volumes <*volume name*>

Defines the set of one or more volumes included in the volume entry. Specify the volumes by name, by using any combination of regular alphanumeric characters and one or more of the following metacharacter expressions:

.

The period matches any single character.

*

The asterisk matches zero or more instances of the preceding character. Combine it with any other alphanumeric character or metacharacter.

[ ]

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 single `a` or `b` or `c`, but not `d` or `A`. This expression can be combined with the asterisk.

^

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

\

A backslash preceding any of the metacharacters in this list makes it match its literal value only. For example, the expression `\.` (backslash and period) matches a single period, `\*` a single asterisk, and `\\` a single backslash. Such expressions can be combined with the asterisk (for example, `\.*` 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, because the period matches any character and the asterisk means any number of that character. As mentioned, it is the only acceptable metacharacter expression for the -server and -partition arguments. In a volume definition it can stand alone (in which case it matches every volume listed in the VLDB), or can combine with regular characters. The following example matches any volume name that begins with the string `user` and ends with `backup`:

```
user.*backup
```

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command adds a volume entry to the volume set called `sys`. The entry matches all volumes on any machine or partition whose names begin with the string `sun4x_56` followed by a period:

```
backup&gt; addvolentry sys .* .* sun4x_56\..*
```

The following command adds a volume entry to the volume set called `fs2`, to match all volumes on the */vicepb* partition of file server machine `fs2.abc.com`. Because it is issued at the shell prompt, double quotes surround the metacharacters in the -volumes argument. (The command is shown here on two lines only for legibility reasons.)

```
% backup addvolentry -name fs2 -server fs2.abc.com \
                     -partition /vicepb -volumes ".*"
```

The chapter in the *IBM AFS Administration Guide* about configuring the AFS Backup System presents additional examples as well as advice on grouping volumes.

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_addvolset(8)" on page 545, "backup_delvolentry(8)" on page 561, "backup_delvolset(8)" on page 563, "backup_listvolsets(8)" on page 611

Copyright

backup addvolset

Name

`backup addvolset` — Creates a new (empty) volume set

Synopsis

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

backup addvols -n <*volume set name*> [-t] [-l] [-c <*cell name*>] [-h]

Description

The backup addvolset command creates a new volume set, by default adding it to the Backup Database. It is best that the volume set's name indicate the volume set's contents; for example, define the volume entries in the `user` volume set to match all user volumes. The volume set name must be unique within the Backup Database of the local cell.

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

Sometimes it is convenient to create volume sets without recording them permanently in the Backup Database, for example when using the backup volsetrestore command to restore a group of volumes that were not necessarily backed up together. To create a *temporary* volume set, include the -temporary flag. A temporary volume set exists only during the lifetime of the current interactive session, so the flag is effective only when used during an interactive session (opened by issuing the backup interactive command). If it is included when the command is issued at the regular command shell prompt, the command appears to succeed, but the volume set is not created. As noted, a temporary volume set ceases to exist when the current interactive session ends, or use the backup delvolset command to delete it before that.

One advantage of temporary volume sets is that the backup addvolset command, and any backup addvolentry commands subsequently used to add volume entries to it, complete more quickly than for regular volume sets, because no records are created in the Backup Database.

Options

-name <*volume set name*>

Names the new volume set. The name can include up to 31 of any character other than the period. Avoid other metacharacters as well.

-temporary

Creates a volume set that exists only within the context of the current interactive session. It is not added to the Backup Database.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command creates a volume set called sys:

```
% backup addvolset sys
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser root if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_addvolentry(8)" on page 541, "backup_delvolentry(8)" on page 561, "backup_delvolset(8)" on page 563, "backup_listvolsets(8)" on page 611, "backup_volsetrestore(8)" on page 653

Copyright

backup apropos

Name

`backup apropos` — Displays each help entry containing a keyword string

Synopsis

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

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

Description

The backup apropos command displays the first line of the online help entry for any backup command that has in its name or short description the string specified by the -topic argument.

To display the syntax for a command, use the backup help command.

Options

-topic <*help string*>

Specifies the keyword string to match, in lowercase letters only. If the string is more than a single word, surround it with double quotes (" ") or other delimiters.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of a command's online help entry names it and briefly describes its function. This command displays the first line for any backup command where the string specified with the -topic argument is part of the command name or first line.

Examples

The following example lists all backup commands that include the word `tape` in their names or short descriptions:

```
% backup apropos 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

See Also

"backup(8)" on page 531, "backup_help(8)" on page 589

Copyright

backup dbverify

Name

`backup dbverify` — Checks the integrity of the Backup Database

Synopsis

backup dbverify [-detail] [-localauth] [-cell <*cell name*>] [-help]

backup db [-d] [-l] [-c <*cell name*>] [-h]

Description

The backup dbverify command checks the integrity of the Backup Database. The command's output indicates whether the Backup Database is damaged (data is corrupted) or not. If the Backup Database is undamaged, it is safe to continue using it. If it is corrupted, discontinue any backup operations until it is repaired.

Cautions

While this command runs, no other backup operation can access the Backup Database; the other commands do not run until this command completes. Avoid issuing this command when other backup operations are likely to run. The backup savedb command repairs some types of corruption.

Options

-detail

Reports the number of orphaned blocks found, any inconsistencies, and the name of the server machine running the Backup Server that is checking its copy of the database.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The command displays one of the following two messages:

Database OK

The database is undamaged and can be used.

Database not OK

The database is damaged. You can use the backup savedb command to repair many kinds of corruption as it creates a backup copy. For more detailed instructions, see the *IBM AFS Administration Guide* chapter about performing backup operations.

The -detail flag provides additional information:

- The number of *orphan blocks* found. These are ranges of memory that the Backup Server preallocated in the database but cannot use. Orphan blocks do not interfere with database access, but do waste disk space. To free the unusable space, dump the database to tape by using the backup savedb command, and then restore it by using the backup restoredb command.
- Any inconsistencies in the database, such as invalid hostnames for Tape Coordinator machines.
- The name of the database server machine on which the Backup Database was checked, designated as the `Database checker`. For a detailed trace of the verification operation, see the `/usr/afs/logs/BackupLog` file on the indicated machine. You can use the bos getlog command to display it.

Examples

The following command confirms that the Backup Database is undamaged:

```
% backup dbverify
Database OK
```

The following command confirms that the Backup Database is undamaged and that it has no orphan blocks or invalid Tape Coordinator entries. The Backup Server running on the machine `db1.abc.com` checked its copy of the Database.

```
% backup dbverify -detail
Database OK
Orphan blocks 0
Database checker was db1.abc.com
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"BackupLog(5)" on page 407, "backup(8)" on page 531, "backup_restoredb(8)" on page 621, "backup_savedb(8)" on page 625, "bos_getlog(8)" on page 691

Copyright

backup deldump

Name

`backup deldump` — Deletes a dump level from the Backup Database

Synopsis

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

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

Description

The backup deldump command 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.

Options

-dump <*dump level name*>

Specifies the complete pathname of the dump level to delete.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command deletes the dump level `/sunday1/monday1` from the dump hierarchy, along with any of its child dump levels.

```
% backup deldump /sunday1/monday1
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_adddump(8)" on page 535, "backup_listdumps(8)" on page 605

Copyright

backup deletedump

Name

`backup deletedump` — Deletes one or more dump records from the Backup Database

Synopsis

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

backup dele [-d <*dump id*>+] [-f <*date time*>+] [-t <*date time*>+] [-l] [-c <*cell name*>] [-h]

Description

The backup deletedump command deletes one or more dump records from the Backup Database. Either use the -dumpid argument to specify the dump ID number of one or more dumps, or use the -from and -to arguments to delete the records for all regular dumps created during the time period bracketed by the specified values.

Use this command to remove dump records that are incorrect (possibly because a dump operation was interrupted or failed), or that correspond to dumps that are expired or otherwise no longer needed.

Cautions

The only way to remove the dump record for an appended dump is to remove the record for its initial dump, and doing so removes the records for all of the initial dump's associated appended dumps.

The only way to remove the record for a Backup Database dump (created with the backup savedb command) is to specify its dump ID number with the -dumpid argument. Using the -from and -to arguments never removes database dump records.

Removing records of a dump makes it impossible to restore data from the corresponding tapes or from any dump that refers to the deleted dump as its parent, directly or indirectly. That is, restore operations must begin with the full dump and continue with each incremental dump in order. If the records for a specific dump are removed, it is not possible to restore data from later incremental dumps unless the deleted records are restored by running the backup scantape command with the -dbadd flag.

If a dump set contains any dumps that were created outside the time range specified by the -from and -to arguments, the command does not delete any of the records associated with the dump set, even if some of them represent dumps created during the time range.

Options

-dumpid <*dump id*>+

Specifies the dump ID of each dump record to delete. The corresponding dumps must be initial dumps; it is not possible to delete appended dump records directly, but only by deleting the record of their associated initial dump. Using this argument is the only way to delete records of Backup Database dumps (created with the backup savedb command).

Provide either this argument or the -to (and optionally -from) argument.

-from <*date time*>+

Specifies the beginning of a range of dates; the record for any dump created during the indicated period of time is deleted.

Omit this argument to indicate the default of midnight (00:00 hours) on 1 January 1970 (UNIX time zero), or provide a date value in the format *mm/dd/yyyy* [*hh:MM*]. The month (*mm*), day (*dd*), and year (*yyyy*) are required. The hour and minutes (*hh:MM*) are optional, but if provided must be in 24-hour format (for example, the value `14:36` represents 2:36 p.m.). If omitted, the time defaults to midnight (00:00 hours).

The -to argument must be provided along with this one.

-to <*date time*>+

Specifies the end of a range of dates; the record of any dump created during the range is deleted from the Backup Database.

Provide either the value `NOW` to indicate the current date and time, or a date value in the same format as for the -from argument. Valid values for the year (*yyyy*) range from `1970` to `2037`; higher values are not valid because the latest possible date in the standard UNIX representation is in February 2038. The command interpreter automatically reduces any later date to the maximum value.

If the time portion (*hh:MM*) is omitted, it defaults to 59 seconds after midnight (00:00:59 hours). Similarly, the backup command interpreter automatically adds 59 seconds to any time value provided. In both cases, adding 59 seconds compensates for how the Backup Database and backup dumpinfo command represent dump creation times in hours and minutes only. For example, the Database records a creation timestamp of `20:55` for any dump operation that begins between 20:55:00 and 20:55:59. Automatically adding 59 seconds to a time thus includes the records for all dumps created during that minute.

Provide either this argument, or the -dumpid argument. This argument is required if the -from argument is provided.

Caution: Specifying the value `NOW` for this argument when the -from argument is omitted deletes all dump records from the Backup Database (except for Backup Database dump records created with the backup savedb command).

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL

Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell *<cell name>*

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

At the conclusion of processing, the output lists the dump IDs of all dump records deleted in the following format:

```
The following dumps were deleted:
     dump ID 1
     dump ID 2
     etc.
```

Examples

The following command deletes the dump record with dump ID 653777462, and for any appended dumps associated with it:

```
% backup deletedump -dumpid 653777462
The following dumps were deleted:
     653777462
```

The following command deletes the Backup Database record of all dumps created between midnight on 1 January 1997 and 23:59:59 hours on 31 December 1997:

```
% backup deletedump -from 01/01/1997 -to 12/31/1997
The following dumps were deleted:
     598324045
     598346873
        ...
        ...
     653777523
     653779648
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_dumpinfo(8)" on page 581, "backup_scantape(8)" on page 629

Copyright

backup delhost

Name

`backup delhost` — Deletes a Tape Coordinator entry from the Backup Database

Synopsis

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

backup delh -t <*tape machine name*> [-p <*TC port offset*>] [-l] [-c <*cell name*>] [-h]

Description

The backup delhost command deletes the indicated Tape Coordinator entry from the Backup Database. It is then impossible to submit backup operations to that Tape Coordinator, even if it is still running. To keep configuration information consistent, also remove the corresponding entry from the */usr/afs/backup/tapeconfig* file on the Tape Coordinator machine.

To list the Tape Coordinator machines and port offsets defined in the Backup Database, issue the backup listhosts command.

Options

-tapehost <*tape machine name*>

Specifies the hostname of the machine housing the Tape Coordinator to delete.

-portoffset <*TC port offset*>

Specifies the port offset number of the Tape Coordinator to delete. If omitted, it defaults to `0`. If provided, it is an integer between `0` (zero) and `58510`, and must match the port offset number assigned to the same combination of Tape Coordinator and tape device or file in the */usr/afs/backup/tapeconfig* file on the Tape Coordinator machine indicated by the -tapehost argument.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command deletes the Backup Database entry for the Tape Coordinator with port offset 2 on the Tape Coordinator machine `backup3.abc.com`:

```
% backup delhost -tapehost backup3.abc.com -portoffset 2
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_addhost(8)" on page 539, "backup_listhosts(8)" on page 609

Copyright

backup delvolentry

Name

`backup delvolentry` — Deletes a volume entry from a volume set

Synopsis

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

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

Description

The backup delvolentry command deletes the indicated volume entry from the volume set specified with the -name argument. Use the -entry argument to identify the volume entry by its index number. To display the index numbers, use the backup listvolsets command.

If there are any remaining volume entries with index numbers higher than the deleted entry, their indexes are automatically decremented to eliminate any gaps in the indexing sequence.

Cautions

Deleting volume entries from a temporary volume set is possible only within the interactive session in which the volume set was created.

Options

-name <*volume set name*>

Names the volume set from which to delete a volume entry.

-entry <*volume set index*>

Specifies the index number of the volume entry to delete. Use the backup listvolsets command to display the index numbers for a volume set's volume entries.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

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

```
% backup delvolentry -name sys -entry 4
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_addvolentry(8)" on page 541, "backup_addvolset(8)" on page 545, "backup_delvolset(8)" on page 563, "backup_listvolsets(8)" on page 611

Copyright

backup delvolset

Name

`backup delvolset` — Deletes one or more volume sets from the Backup Database

Synopsis

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

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

Description

The backup delvolset command deletes each volume set named by the -name argument, and the volume entries each contains, from the Backup Database. The backup listvolsets command lists the volume sets (and their volume entries) currently defined in the Backup Database.

Cautions

Deleting a temporary volume set is possible only within the interactive session in which it was created. Exiting the interactive session also destroys the temporary volume set automatically.

Options

-name <*volume set name*>+

Names each volume set to delete.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command deletes the volume set called user and all volume entries in it:

```
% backup delvolset user
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_addvolentry(8)" on page 541, "backup_addvolset(8)" on page 545, "backup_delvolentry(8)" on page 561, "backup_listvolsets(8)" on page 611

Copyright

backup diskrestore

Name

`backup diskrestore` — Restores the entire contents of a partition

Synopsis

backup diskrestore -server <*machine to restore*> -partition <*partition to restore*> [-portoffset <*TC port offset*>+] [-newserver <*destination machine*>] [-newpartition <*destination partition*>] [-extension <*new volume name extension*>] [-n] [-localauth] [-cell <*cell name*>] [-help]

backup di -s <*machine to restore*> -pa <*partition to restore*> [-po <*TC port offset*>+] [-news <*destination machine*>] [-newp <*destination partition*>] [-e <*new volume name extension*>] [-n] [-l] [-c <*cell name*>] [-h]

Description

The backup diskrestore command restores all of the volumes for which the Volume Location Database (VLDB) lists a read / write site on the partition specified with the -server and -partition arguments. It is useful if a disk or machine failure corrupts or destroys the data on an entire partition. (To restore any read-only or backup volumes that resided on the partition, use the vos release and vos backup commands, respectively, after restoring the read / write version.)

If restoring only selected volumes to a single site, it is usually more efficient to use the backup volrestore command. To restore multiple volumes to many different sites, use the backup volsetrestore command.

(If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file on the Tape Coordinator machine associated with the specified port offset, then the Backup System restores data from the backup data file listed for that port offset in the Tape Coordinator's */usr/afs/backup/tapeconfig* file, instead of from tape. For the sake of clarity, the following text refers to tapes only, but the Backup System handles backup data files in much the same way.)

The Backup System determines whether the read / write or backup version of each volume was dumped more recently, and restores the dumps of that version, starting with the most recent full dump. It resets the creation timestamp of each restored volume to the date and time at which it begins restoring the volume (the creation timestamp appears in the `Creation` field of the output from the vos examine and vos listvol commands).

If all of the full and incremental dumps of all relevant volumes were not written on compatible tape devices, use the -portoffset argument to list multiple port offset numbers in the order in which the tapes are needed (first list the port offset for the full dump, second the port offset for the level 1 incremental dump, and so on). This implies that the full dumps of all relevant volumes must have been written to a type of tape that the first Tape Coordinator can read, the level 1 incremental dumps to a type of tape the second Tape Coordinator can read, and so on. If dumps are on multiple incompatible tape types, use the backup volrestore command to restore individual volumes, or the backup volsetrestore command after defining groups of volumes

that were dumped to compatible tape types. For further discussion, see the *IBM AFS Administration Guide*.

By default, the Backup System restores the contents of the specified partition to that same partition. To restore the contents to an alternate site, combine the following options as indicated. The Backup System removes each volume from the original site, if it still exists, and records the change of site in the VLDB.

- To restore to a different partition on the same file server machine, provide the -newpartition argument.
- To restore to the partition with the same name on a different file server machine, provide the -newserver argument.
- To restore to a completely different site, combine the -newserver and -newpartition arguments.

By default, the Backup System overwrites the contents of existing volumes with the restored data. To create a new volume to house the restored data instead, use the -extension argument. The Backup System creates the new volume at the site designated by the -newserver and -newpartition arguments if they are used or the -server and -partition arguments otherwise. It derives the volume name by adding the extension to the read/write base name listed in the VLDB, and creates a new VLDB entry. The command does not affect the existing volume in any way. However, if a volume with the specified extension also already exists, the command overwrites it.

To print out a list of the tapes containing the needed dumps, without actually performing the restore operation, include the -n flag along with the other options to be used on the actual command.

The Tape Coordinator's default response to this command is to access the first tape it needs by invoking the `MOUNT` instruction in the local *`CFG_device_name`* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *`CFG_device_name`* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, or is the wrong tape, the Tape Coordinator invokes the `MOUNT` instruction or prompts the operator. It also invokes the `MOUNT` instruction or prompts for any additional tapes needed to complete the restore operation; the backup operator must arrange to provide them.

Cautions

If issuing this command to recover data after a disk crash or other damage, be sure not to issue the vos syncserv command first. Doing so destroys the VLDB record of the volumes that resided on the partition.

Options

-server <*machine to restore*>

Names the file server machine that the VLDB lists as the site of the volumes that need to be

restored.

-partition <*partition to restore*>

Names the partition that the VLDB lists as the site of the volumes that need to be restored.

-portoffset <*TC port offset*>+

Specifies one or more port offset numbers (up to a maximum of 128), each corresponding to a Tape Coordinator to use in the operation. If there is more than one value, the Backup System uses the first one when restoring the full dump of each volume, the second one when restoring the level 1 incremental dump of each volume, and so on. It uses the final value in the list when restoring dumps at the corresponding depth in the dump hierarchy and at all lower levels.

Provide this argument unless the default value of 0 (zero) is appropriate for all dumps. If `0` is just one of the values in the list, provide it explicitly in the appropriate order.

-newserver <*destination machine*>

Names an alternate file server machine to which to restore the volumes. If this argument is omitted, the volumes are restored to the file server machine named by the -server argument.

-newpartition <*destination partition*>

Names an alternate partition to which to restore the data. If this argument is omitted, the volumes are restored to the partition named by the -partition argument.

-extension <*new volume name extension*>

Creates a new volume for each volume being restored, to house the restored data. The Backup System derives the new volume's name by appending the specified string to the read/write base name listed in the VLDB, and creates a new VLDB volume entry. The Backup System preserves the contents of the volumes on the partition, if any still exist. Any string other than `.readonly` or `.backup` is acceptable, but the combination of the base name and extension cannot exceed 22 characters in length. To use a period to separate the extension from the name, specify it as the first character of the string (as in `.rst`, for example).

-n

Displays a list of the tapes necessary to perform the requested restore, without actually performing the operation.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

```
Restore operation on volume I&lt;name&gt; 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. Enter the value y to continue the operation without restoring the indicated volume or the value n to terminate the operation. In the latter case, the operator can then attempt to determine the cause of the tape error.

If the issuer includes the -n flag with the command, the following string appears at the head of the list of the tapes necessary to perform the restore operation:

```
Tapes needed:
```

Examples

The following command restores the volumes for which the VLDB lists a read / write site on the */vicepd* partition of the machine `fs5.abc.com`. The Tape Coordinator associated with port offset 3 performs the operation.

```
% backup diskrestore -server fs5.abc.com -partition /vicepd -portoffset 3
```

The following command restores the volumes for which the VLDB lists a read / write site on the */vicepb* partition of the machine `fs1.abc.com` to a new site: the */vicepa* partition on the machine `fs3.abc.com`. The Tape Coordinator associated with port offset 0 performs the operation. (The command appears here on two lines only for legibility.)

```
% backup diskrestore  -server fs1.abc.com -partition /vicepb   \
                      -newserver fs3.abc.com -newpartition /vicepa
```

The following command lists the tapes required to restore the volumes for which the VLDB lists a read / write site on the */vicepm* partition of the machine `fs4.abc.com`:

```
% backup diskrestore -server fs4.abc.com -partition /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 the */usr/afs/etc/UserList* file on every machine where the Backup Server or Volume Location (VL) Server is running, and on every file server machine that houses an affected volume. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_dump(8)" on page 571, "backup_volrestore(8)" on page 647, "backup_volsetrestore(8)" on page 653, "butc(8)" on page 767, "vos_backup(1)" on page 299, "vos_examine(1)" on page 323, "vos_listvol(1)" on page 339, "vos_release(1)" on page 355

Copyright

backup dump

Name

`backup dump` — Creates a dump (dumps a volume set at a particular dump level)

Synopsis

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*>] [-localauth] [-cell <*cell name*>] [-help]

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*>] [-l] [-c <*cell name*>] [-h]

Description

The backup dump command either dumps the volume set specified by the -volumeset argument at the dump level specified by the -dump argument and creates a Backup Database dump record about it, or executes the dump instructions listed in the file named by the -file argument. The Tape Coordinator indicated by the -portoffset argument (or on each command in the file) executes the operation.

(If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file on the Tape Coordinator machine associated with the specified port offset, then the Backup System dumps data to the backup data file listed for that port offset in the Tape Coordinator's */usr/afs/backup/tapeconfig* file, rather than to tape. For the sake of clarity, the following text refers to tapes only, but the Backup System handles backup data files in much the same way.)

The term *dumping* refers to copying a collection of data to tape or a backup data file, and the resulting collection is termed a *dump*. The set of tapes that contain one or more dumps is called a *dump set*. The first dump in a dump set is its *initial dump*, and any dumps subsequently added to the dump set (by use of the -append argument) are *appended dumps*. Creating appended dumps is optional, and appended dumps can be of different volume sets, and at different dump levels, than the initial dump.

A *full dump*, created at a full dump level in the dump hierarchy, contains all of the data that existed at the time of the dump in the volumes belonging to the volume set. An *incremental dump*, created at an incremental dump level, contains only data that has changed since the volume set was dumped at the incremental level's *parent dump level* (the dump level immediately above the incremental level in the hierarchy), which can be a full or incremental level. More specifically, an incremental dump includes only the files and directories that have modification timestamps later than the *clone date* of the volume included at the parent dump level. For backup and read-only volumes, the clone date is the time at which the volume was cloned from its read / write source before being included in the parent dump; for read / write volumes, it represents the time at which the volume was locked for inclusion in the parent dump. The clone date appears in the *clone date* field of the output from the backup volinfo command. As an example, an incremental dump at the `/full/week1/thursday` level includes

only files and directories that have changed since the volume set was dumped at the `/full/week1` level.

Initiating different types of dump operations

To initiate a dump operation that is to start as soon as the relevant Tape Coordinator is available, provide only the -volumeset, -dump, -portoffset, and optionally -append options. To schedule a single backup dump command to execute in the future, also include the -at argument to specify the start time.

To append a dump to an existing dump set, include the -append flag. The Backup System imposes the following conditions on appended dumps:

- If writing to tape, the Tape Coordinator checks that it is the final one in a dump set for which there are complete and valid tape and dump records in the Backup Database. If not, it rejects the tape and requests an acceptable one. The operator can use the -dbadd argument to the backup scantape command to insert the necessary records into the database.
- The most recent dump on the tape or in the backup data file must have completed successfully.
- The dump set must begin with an initial dump that is recorded in the Backup Database. If there are no dumps on the tape, then the Backup System treats the dump operation as an initial dump and imposes the relevant requirements (for example, checks the AFS tape name if appropriate).

To schedule multiple dump operations, list the operations in the file named by the -file argument. Optionally include the -at argument to specify when the backup command interpreter reads the file; otherwise it reads it immediately. Do not combine the -file argument with the command's first three arguments or the -append or -n flags. The commands in the file can include any of the backup dump command's arguments, including the -at argument to schedule them to run even later in the future.

To generate a list of the volumes included in a dump, without actually dumping them, combine the -n flag with the options to be used on the actual command.

How the Backup System executes a dump operation

Before beginning a dump operation, the Backup System verifies that there is a Backup Database entry for the volume set, dump level, and port offset. If the command is correctly formed and issued in interactive mode, it is assigned a job number and added to the jobs list. List jobs in interactive mode by using the backup jobs command; terminate them with the backup kill command.

After obtaining the list of volumes to dump from the Volume Location (VL) Server, the Backup System sorts the list by site (server and partition). It groups volumes from the same site together in the dump to minimize the number of times the operator must change tapes during restore operations.

The dependence of an incremental dump on its parent means that a valid parent dump must already exist for the Backup System to create its child incremental dump. If the Backup System does not find a record of a dump created at the immediate 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 necessary. It creates an incremental dump at the level one below the lowest valid parent dump set that it finds. If it fails to find even a full dump, it dumps the volume set at the full dump level.

If the Backup System is unable to access a volume during a dump operation, it skips the volume and dumps the remaining volumes from the volume set. Possible reasons a volume is inaccessible include server machine or process outages, or that the volume was moved between the time the Volume Location (VL) Server generated the list of sites for the volume in the volume set and the time the Backup System actually attempts to dump the data in it. After the first dumping pass, the Backup System attempts to dump each volume it skipped. If it still cannot dump a volume and the `ASK NO` instruction does not appear in the *`CFG_device_name`* file, it queries the operator as to whether it needs to attempt to dump the volume again, omit the volume from the dump, or halt the dump operation altogether. When prompted, the operator can attempt to solve whatever problem prevented the Backup System from accessing the volumes. If the `ASK NO` instruction appears in the *`CFG_device_name`* file, the Backup System omits the volume from the dump.

Before scheduling a dump operation, the Backup System verifies that the date specified by the -at argument is in the future, and checks the validity of the volume set, dump level and port offset as for a regular dump operation. It checks the validity of the parameters again just before actually running the scheduled operation.

Before writing an initial dump to a tape that does not have a permanent name on the label, the Backup System checks that the AFS tape name on the label is acceptable. If desired, disable name checking by including the `NAME_CHECK NO` instruction in the *`CFG_device_name`* file.

If AFS tape name checking is enabled, the Backup System accepts the following three types of values for the AFS tape name. If the name on the label does not conform, the Backup System obtains a tape with an acceptable label by invoking the `MOUNT` instruction in the *`CFG_device_name`* file or prompting the operator.

- A name of the form *volume_set_name.dump_level_name.tape_index*, where *volume_set_name* matches the value of the -volumeset argument, *dump_level_name* matches the last element in the pathname value of the -dump argument, and *tape_index* reflects the tape's place in a multitape dump set. As an example, the first tape in a dump set for which the initial dump is of volume set `user` at the dump level `/sunday2/monday` has AFS tape name `user.monday.1`. If the label records this type of AFS tape name, the Backup System retains the AFS tape name and writes the dump to the tape.
- The string `<NULL>`, which usually indicates that a backup operator has used the backup labeltape command to write a label on the tape, but did not include the -name argument to assign an AFS tape name. Presumably, the operator did include the -pname argument to assign a permanent name. If the label records a `<NULL>` value, the Backup System constructs and records on the label the appropriate AFS tape name, and writes the dump on the tape.
- No value at all, because the tape has never been labeled or used in the Backup System. As when the AFS tape name is `<NULL>`, the Backup System constructs and records on the label the appropriate AFS tape name, and writes the dump on the tape.

To determine how much data it can write to a tape, the Tape Coordinator reads the capacity recorded on the tape's label (placed there by including the -size argument to the backup

labeltape command). If the label's capacity field is empty, the Tape Coordinator uses the capacity recorded for the specified port offset in the local *tapeconfig* file. If the capacity field in the *tapeconfig* file is also empty, the Tape Coordinator uses the maximum capacity of 2 TB.

During a dump operation, the Tape Coordinator tracks how much data it has written and stops shortly before it reaches what it believes is the tape's capacity. If it is in the middle of writing the data for a volume when it reaches that point, it writes a special marker that indicates an interrupted volume and continues writing the volume on the next tape. It can split a volume this way during both an initial and an appended dump, and the fact that the volume resides on multiple tapes is automatically recorded in the Backup Database.

If the tape is actually larger than the expected capacity, then the Tape Coordinator simply does not use the excess tape. If the tape is smaller than the expected capacity, the Tape Coordinator can reach the end-of-tape (EOT) unexpectedly while it is writing data. If the Tape Coordinator is in the middle of the writing data from a volume, it obtains a new tape and rewrites the entire contents of the interrupted volume to it. The data from the volume that was written to the previous tape remains there, but is never used.

The Backup System allows recycling of tapes (writing a new dump set over an old dump set that is no longer needed), but imposes the following conditions:

- All dumps in the old dump set must be expired. The Backup System always checks expiration dates, even when name checking is disabled.
- If the tape to be recycled does not have a permanent name and name checking is enabled, then the AFS tape name derived from the new initial dump's volume set name and dump level name must match the AFS tape name already recorded on the label.
- The tape cannot already have data on it that belongs to the dump currently being performed, because that implies that the operator or automated tape device has not removed the previous tape from the drive, or has mistakenly reinserted it. The Tape Coordinator generates the following message and attempts to obtain another tape:

  ```
  Can't overwrite tape containing the dump in progress
  ```

- The tape cannot contain data from a parent dump of the current (incremental) dump, because overwriting a parent dump makes it impossible to restore data from the current dump. The Tape Coordinator generates the following message and attempts to obtain another tape:

  ```
  Can't overwrite the parent dump I&lt;parent_name&gt; (I&lt;parent_dump_ID&gt;)
  ```

To recycle a tape before all dumps on it have expired or if the AFS tape name is wrong, use the backup labeltape command to overwrite the tape's label and remove all associated tape and dump records from the Backup Database.

The Tape Coordinator's default response to this command is to access the first tape by invoking the `MOUNT` instruction in the *CFG_device_name* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *CFG_device_name* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, the Tape Coordinator invokes the `MOUNT` instruction or prompts the operator. It also

invokes the `MOUNT` instruction or prompts for any additional tapes needed to complete the dump operation; the issuer must arrange to provide them.

Cautions

If a dump operation is interrupted or fails for any reason, data from all volumes written to tape before the interrupt are valid can be used in a restore operation. The Backup Database includes an entry for the failed dump and for each volume that was successfully dumped. See the *IBM AFS Administration Guide* for information on dealing with interrupted dumps.

If dumping to tape rather than a backup data file, it is best to use only compatible tape devices (ones that can read the same type of tape). Using compatible devices greatly simplifies restore operations. The -portoffset argument to the backup diskrestore and backup volsetrestore commands accepts multiple port offset numbers, but the Backup System uses the first listed port offset when restoring all full dumps, the second port offset when restoring all level 1 dumps, and so on. At the very least, use compatible tape devices to perform dumps at each level. If compatible tape devices are not used, the backup volrestore command must be used to restore one volume at a time.

Valid (unexpired) administrative tokens must be available to the backup command interpreter both when it reads the file named by the -file argument and when it runs each operation listed in the file. Presumably, the issuer is scheduling dumps for times when no human operator is present, and so must arrange for valid tokens to be available on the local machine. One option is to issue all commands (or run all scripts) on file server machines and use the -localauth flag on the backup and vos commands. To protect against improper access to the machine or the tokens, the machine must be physically secure (perhaps even more protected than a Tape Coordinator machine monitored by a human operator during operation). Also, if an unattended dump requires multiple tapes, the operator must properly configure a tape stacker or jukebox and the device configuration file.

When the command is issued in regular (non-interactive) mode, the command shell prompt does not return until the dump operation completes. To avoid having to open additional connections, issue the command in interactive mode, especially when including the -at argument to schedule dump operations.

Options

-volumeset <*volume set name*>

Names the volume set to dump. The -dump argument must be provided along with this one; do not combine them with the -file argument. If using a temporary volume set, the vos dump command must be issued within the interactive session in which the backup addvolset command was issued with the -temporary flag.

-dump <*dump level name*>

Specifies the complete pathname of the dump level at which to dump the volume set. The -volumeset argument must be provided along with this one; do not combine them with

the -file argument.

-portoffset <*TC port offset*>

Specifies the port offset number of the Tape Coordinator handling the tapes for this operation. It must be provided unless the default value of 0 (zero) is appropriate; do not combine it with the -file argument.

-at <*date/time to start dump*>

Specifies the date and time in the future at which to run the command, or to read the file named by the -file argument. Provide a value in the format *mm/dd/yyyy* [*hh:MM*], where the month (*mm*), day (*dd*), and year (*yyyy*) are required. Valid values for the year range from `1970` to `2037`; higher values are not valid because the latest possible date in the standard UNIX representation is in February 2038. The Backup System automatically reduces any later date to the maximum value.

The hour and minutes (*hh:MM*) are optional, but if provided must be in 24-hour format (for example, the value `14:36` represents 2:36 p.m.). If omitted, the time defaults to midnight (00:00 hours).

As an example, the value 04/23/1999 20:20 schedules the command for 8:20 p.m. on 23 April 1999.

-append

Appends the dump onto the end of a tape that already contains data from another dump. However, if the tape is not in fact part of an existing dump set, the Backup System creates a new dump set using the parameters of this dump. If the tape is not the last tape in the dump set, the Tape Coordinator prompts for insertion of the appropriate tape. Do not combine this argument with the -file argument.

-n

Displays the names of volumes to be included in the indicated dump, without actually performing the dump operation. Do not combine this argument with the -file argument.

-file <*load file*>

Specifies the local disk or AFS pathname of a file containing backup commands. The Backup System reads the file immediately, or at the time specified by the -at argument if it is provided. A partial pathname is interpreted relative to the current working directory.

Place each backup dump command on its own line in the indicated file, using the same syntax as for the command line, but without the word backup at the start of the line. Each command must include a value for the -volumeset and -dump arguments, and for the -portoffset argument unless the default value of 0 is appropriate. Commands in the file can also include any of the backup dump command's optional options. In the following example file, the first command runs as soon as the Backup System reads the file, whereas the other commands are themselves scheduled; the specified date and time must be later than the date and time at which the Backup System reads the file.

```
dump user /sunday1/wednesday -port 1
dump sun4x_56 /sunday1/friday -port 2 -at 04/08/1999
dump sun4x_55 /sunday1/friday -port 2 -at 04/08/1999 02:00 -append
```

Do not combine this argument with the -volumeset, -dump, -portoffset, -append, or -n options.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The command interpreter first generates a list of the volumes to be included in the dump by matching the entries in the volume set against the volumes listed in the Volume Location Database (VLDB). It prints the list following the header:

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

The following message then indicates that the command interpreter has passed the dump request to the appropriate Tape Coordinator for processing:

```
Starting dump.
```

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

```
Starting dump of volume set '&lt;volume set&gt;' (dump set '&lt;dump level&gt;')
Total number of volumes : &lt;number dumped&gt;
Would have dumped the following volumes:
&lt;list_of_volumes&gt;
```

where *list_of_volumes* identifies each volume by name and volume ID number.

If the Tape Coordinator is unable to access a volume, it prints an error message in its window and records the error in its log and error files.

Examples

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

```
% backup dump -volumeset user -dump /full/sunday2/monday -portoffset 5
Preparing to dump the following volumes:
user.jones.backup   387623900
user.pat.backup     486219245
user.smith.backup   597315841
       .               .
       .               .
Starting dump.
```

The following command displays the list of volumes to be dumped when the user dumps the sys_sun volume set at the /full dump level.

```
% backup dump -volumeset sys_sun -dump /full -n
Starting dump of volume set 'sys_sun' (dump set '/full')
Total number of volumes: 24
Would have dumped the following volumes:
sun4x_56      124857238
sun4x_56.bin  124857241
    .            .
    .            .
sun4x_55      124857997
    .            .
    .            .
```

The following command schedules a dump of the volumes in the volume set user at the dump level /sunday2/monday1 for 11:00 p.m. on 14 June 1999. The appropriate Tape Coordinator has port offset 0 (zero), so that argument is omitted.

```
% backup dump -volumeset user -dump /sunday2/monday1 -at 06/14/1999 23:00
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server or Volume Location (VL) Server is running, and on every file server machine that houses an affected volume. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser root.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_adddump(8)" on page 535, "backup_addvolentry(8)" on page 541, "backup_addvolset(8)" on page 545, "backup_diskrestore(8)" on page 565, "backup_labeltape(8)" on page 601, "backup_volrestore(8)" on page 647, "butc(8)" on page 767

Copyright

backup dumpinfo

Name

`backup dumpinfo` — Displays a dump record from the Backup Database

Synopsis

backup dumpinfo [-ndumps <*number of dumps*>] [-id <*dump id*>] [-verbose] [-localauth] [-cell <*cell name*>] [-help]

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

Description

The backup dumpinfo command formats and displays the Backup Database record for the specified dumps. To specify how many of the most recent dumps to display, starting with the newest one and going back in time, use the -ndumps argument. To display more detailed information about a single dump, use the -id argument. To display the records for the 10 most recent dumps, omit both the -ndumps and -id arguments.

The -verbose flag produces very detailed information that is useful mostly for debugging purposes. It can be combined only with the -id argument.

Options

-ndumps <*number of dumps*>

Displays the Backup Database record for each of the specified number of dumps that were most recently performed. If the database contains fewer dumps than are requested, the output includes the records for all existing dumps. Do not combine this argument with the -id or -verbose options; omit all options to display the records for the last 10 dumps.

-id <*dump id*>

Specifies the dump ID number of a single dump for which to display the Backup Database record. Precede the *dump id* value with the -id switch; otherwise, the command interpreter interprets it as the value of the -ndumps argument. Combine this argument with the -verbose flag, but not with the -ndumps argument; omit all options to display the records for the last 10 dumps.

-verbose

Provides more detailed information about the dump specified with the -id argument, which must be provided along with it. Do not combine this flag with the -ndumps argument.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If the -ndumps argument is provided, the output presents the following information in table form, with a separate line for each dump:

dumpid

The dump ID number.

parentid

The dump ID number of the dump's parent dump. A value of `0` (zero) identifies a full dump.

lv

The depth in the dump hierarchy of the dump level used to create the dump. A value of `0` (zero) identifies a full dump, in which case the value in the `parentid` field is also `0`. A value of `1` or greater indicates an incremental dump made at the corresponding 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 that contain the data in the dump. A value of `0` (zero) indicates that the dump operation was terminated or failed. Use the backup deletedump command to remove such entries.

nvols

The number of volumes from which the dump includes data. If a volume spans tapes, it is counted twice. A value of `0` (zero) indicates that the dump operation was terminated or failed; the value in the `nt` field is also `0` in this case.

dump name

The dump name in the form

```
&lt;volume_set_name&gt;.&lt;dump_level_name&gt; (&lt;initial_dump_ID&gt;)
```

where <volume_set_name> is the name of the volume set, and <dump_level_name> is the last element in the dump level pathname at which the volume set was dumped.

The <initial_dump_ID>, if displayed, is the dump ID of the initial dump in the dump set to which this dump belongs. If there is no value in parentheses, the dump is the initial dump in a dump set that has no appended dumps.

If the -id argument is provided alone, the first line of output begins with the string `Dump` and reports information for the entire dump in the following fields:

id

The dump ID number.

level

The depth in the dump hierarchy of the dump level used to create the dump. A value of `0` (zero) identifies a full dump. A value of `1` (one) or greater indicates an incremental dump made at the specified level in the dump hierarchy.

volumes

The number of volumes for which the dump includes data.

created

The date and time at which the dump operation began.

If an XBSA server was the backup medium for the dump (rather than a tape device or backup data file), the following line appears next:

```
Backup Service: &lt;XBSA_program&gt;: Server: &lt;hostname&gt;
```

where <XBSA_program> is the name of the XBSA-compliant program and <hostname> is the name of the machine on which the program runs.

Next the output includes an entry for each tape that houses volume data from the dump. Following the string `Tape`, the first two lines of each entry report information about that tape in the following fields:

name

The tape's permanent name if it has one, or its AFS tape name otherwise, and its tape ID number in parentheses.

nVolumes

The number of volumes for which this tape includes dump data.

created

The date and time at which the Tape Coordinator began writing data to this tape.

Following another blank line, the tape-specific information concludes with a table that includes a line for each volume dump on the tape. The information appears in columns with the following headings:

Pos

The relative position of each volume in this tape or file. On a tape, the counter begins at position 2 (the tape label occupies position 1), and increments by one for each volume. For volumes in a backup data file, the position numbers start with 1 and do not usually increment only by one, because each is the ordinal of the 16 KB offset in the file at which the volume's data begins. The difference between the position numbers therefore indicates how many 16 KB blocks each volume's data occupies. For example, if the second volume is at position 5 and the third volume in the list is at position 9, that means that the dump of the second volume occupies 64 KB (four 16-KB blocks) of space in the file.

Clone time

For a backup or read-only volume, the time at which it was cloned from its read / write source. For a Read / Write volume, it is the same as the dump creation date reported on the first line of the output.

Nbytes

The number of bytes of data in the dump of the volume.

Volume

The volume name, complete with `.backup` or `.readonly` extension if appropriate.

If both the -id and -verbose options are provided, the output is divided into several sections:

- The first section, headed by the underlined string `Dump`, includes information about the entire dump. The fields labeled `id`, `level`, `created`, and `nVolumes` report the same values (though in a different order) as appear on the first line of output when the -id argument is provided by itself. Other fields of potential interest to the backup operator are:

  Group id

  The dump's *group ID number*, which is recorded in the dump's Backup Database record if the `GROUPID` instruction appears in the Tape Coordinator's */usr/afs/backup/CFG_tcid* file when the dump is created.

  maxTapes

  The number of tapes that contain the dump set to which this dump belongs.

  Start Tape Seq

  The ordinal of the tape on which this dump begins in the set of tapes that contain the dump set.

- For each tape that contains data from this dump, there follows a section headed by the underlined string `Tape`. The fields labeled `name`, `written`, and `nVolumes` report the same values (though in a different order) as appear on the second and third lines of output when the -id argument is provided by itself. Other fields of potential interest to the backup operator are:

  expires

  The date and time when this tape can be recycled, because all dumps it contains have expired.

  nMBytes Data and nBytes Data

  Summed together, these fields represent the total amount of dumped data actually from volumes (as opposed to labels, filemarks, and other markers).

  KBytes Tape Used

  The number of kilobytes of tape (or disk space, for a backup data file) used to store the dump data. It is generally larger than the sum of the values in the `nMBytes Data` and `nBytes Data` fields, because it includes the space required for the label, file marks and other markers, and because the Backup System writes data at 16 KB offsets, even if the data in a given block doesn't fill the entire 16 KB.

- For each volume on a given tape, there follows a section headed by the underlined string `Volume`. The fields labeled `name`, `position`, `clone`, and `nBytes` report the same values (though in a different order) as appear in the table that lists the volumes in each tape when the -id argument is provided by itself. Other fields of potential interest to the backup operator are:

  id

  The volume ID.

  tape

  The name of the tape containing this volume data.

Examples

The following example displays information about the last five dumps:

```
% backup dumpinfo -ndumps 5
   dumpid   parentid lv created          nt nvols dump name
924424000          0 0  04/18/1999 04:26  1    22 usr.sun (924424000)
924685000  924424000 1  04/21/1999 04:56  1    62 usr.wed (924424000)
924773000  924424000 1  04/22/1999 05:23  1    46 usr.thu (924424000)
924860000  924424000 1  04/23/1999 05:33  1    58 usr.fri (924424000)
```

```
925033000          0 0  04/25/1999 05:36  2    73 sys.week
```

The following example displays a more detailed record for a single dump.

```
% backup dumpinfo -id 922097346
Dump: id 922097346, level 0, volumes 1, created Mon Mar 22 05:09:06 1999
Tape: name monday.user.backup (922097346)
nVolumes 1, created 03/22/1999 05:09
 Pos       Clone time   Nbytes Volume
   1 03/22/1999 04:43 27787914 user.pat.backup
```

The following example displays even more detailed information about the dump displayed in the previous example (dump ID 922097346). This example includes only one exemplar of each type of section (`Dump`, `Tape`, and `Volume`):

```
% backup dumpinfo -id 922097346 -verbose
Dump
----
id = 922097346
Initial id = 0
Appended id = 922099568
parent = 0
level = 0
flags = 0x0
volumeSet = user
dump path = /monday1
name = user.monday1
created = Mon Mar 22 05:09:06 1999
nVolumes = 1
id  = 0
tapeServer =
format= user.monday1.%d
maxTapes = 1
Start Tape Seq = 1
name = pat
instance =
cell =
Tape
----
tape name = monday.user.backup
AFS tape name = user.monday1.1
flags = 0x20
written = Mon Mar 22 05:09:06 1999
expires = NEVER
kBytes Tape Used = 121
nMBytes Data = 0
nBytes  Data = 19092
nFiles = 0
nVolumes = 1
seq = 1
tapeid = 0
```

```
useCount = 1
dump = 922097346
Volume
------
name = user.pat.backup
flags = 0x18
id = 536871640
server =
partition = 0
nFrags = 1
position = 2
clone = Mon Mar 22 04:43:06 1999
startByte = 0
nBytes = 19092
seq = 0
dump = 922097346
tape = user.monday1.1
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_deletedump(8)" on page 555

Copyright

backup help

Name

`backup help` — Displays help for backup commands

Synopsis

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

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

Description

The backup help command displays the complete online help entry (short description and syntax statement) for each operation code specified by the -topic argument. If the -topic argument is omitted, the output includes the first line (name and short description) of the online help entry for every backup command.

To list every backup command whose name or short description includes a specified keyword, use the backup apropos command.

Options

-topic <*help string*>+

Indicates each command for which to display the complete online help entry. Omit the backup part of the command name, providing only the operation code (for example, specify dump, not backup dump). If this argument is omitted, the output briefly describes every backup command.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The first line names the command and briefly describes its function.
- The second line lists aliases for the command, if any.
- The final line, which begins with the string `Usage`, lists the command's options in the prescribed order. Online help entries use the same symbols (for example, brackets) as the reference pages in this document.

Examples

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

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

Privilege Required

None

See Also

"backup(8)" on page 531, "backup_apropos(8)" on page 547

Copyright

backup interactive

Name

`backup interactive` — Enters interactive mode

Synopsis

backup [interactive] [-localauth] [-cell <*cell name*>] [-help]

backup [i] [-l] [-c <*cell name*>] [-h]

Description

The backup interactive initiates an interactive session for issuing backup commands. As indicated in the syntax statement, the operation code (interactive) is optional.

Several features of interactive mode distinguish it from regular mode:

- In interactive mode, the backup> prompt replaces the system (shell) prompt. The operator enters only a command's operation code (omitting the command suite name, backup).
- If the -localauth flag or the -cell argument is included on the backup interactive command, the settings apply to all commands issued during that interactive session. The issuer does not need to type them on every command. Another consequence is that the flag and argument do not appear in the syntax statement generated by the help subcommand or -help flag on an individual command issued at the backup> prompt.
- The backup jobs and backup kill commands are available only in interactive mode. It is not possible to track and terminate backup operations as cleanly in non-interactive mode.
- It is not necessary to enclose strings that include metacharacters in double quotes or other delimiters.
- The backup command interpreter establishes a connection to the Backup Server, Volume Server and Volume Location (VL) Server processes as it enters interactive mode, and uses the same connection for all commands during the session. Execution time can therefore be faster than in non-interactive mode, in which the command interpreter must establish a new connection for each command.

To exit an interactive session, issue the backup quit command.

Options

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example shows how the -localauth flag and -cell argument do not appear when the help dump subcommand is issued in interactive mode.

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

Privilege Required

None. However, backup commands that require privilege in regular mode still require it in interactive mode.

See Also

"backup(8)" on page 531, "backup_jobs(8)" on page 593, "backup_kill(8)" on page 597, "backup_quit(8)" on page 613, "butc(8)" on page 767

Copyright

backup jobs

Name

`backup jobs` — Lists pending and running operations in interactive mode

Synopsis

jobs [-help]

j [-h]

Description

The backup jobs command lists the job ID number and status of each backup operation running or pending in the current interactive session.

This command can be issued in interactive mode only. If the issuer of the backup interactive command included the -localauth flag, the -cell argument, or both, those settings apply to this command also.

To terminate operations that appear in the output, issue the backup kill command and identify the operation to cancel with the job ID number from this command's output.

To check the status of a Tape Coordinator, rather than of a certain operation, use the backup status command.

Options

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output always includes the expiration date and time of the tokens that the backup command interpreter is using during the current interactive session, in the following format:

```
&lt;date&gt;   &lt;time&gt;: TOKEN EXPIRATION
```

If the execution date and time specified for a scheduled dump operation is later than <date time>, then its individual line (as described in the following paragraphs) appears below this line to indicate that the current tokens will not be available to it.

If the issuer of the backup command included the -localauth flag when entering interactive mode, the line instead reads as follows:

```
:  TOKEN NEVER EXPIRES
```

The entry for a scheduled dump operation has the following format:

```
Job &lt;job_ID&gt;:  &lt;timestamp&gt;:  dump  &lt;volume_set&gt;  &lt;dump_level&gt;
```

where

<job_ID>

Is a job identification number assigned by the Backup System.

<timestamp>

Indicates the date and time the dump operation is to begin, in the format *month*/*date*/*year hours*:*minutes* (in 24-hour format)

<volume_set>

Indicates the volume set to dump.

<dump_level>

Indicates the dump level at which to perform the dump operation.

The line for a pending or running operation of any other type has the following format:

```
Job &lt;job_ID&gt;:  &lt;operation&gt;  &lt;status&gt;
```

where

<job_ID>

Is a job identification number assigned by the Backup System.

<operation>

Identifies the operation the Tape Coordinator is performing, which is initiated by the indicated command:

Dump (*dump name*)

Initiated by the backup dump command. The *dump name* has the following format:

```
&lt;volume_set_name&gt;.&lt;dump_level_name&gt;
```

Restore

Initiated by the backup diskrestore, backup volrestore, or backup volsetrestore command.

Labeltape (*tape_label*)

Initiated by the backup labeltapen command. The *tape_label* is the name specified by the backup labeltape command's -name or -pname argument.

Scantape

Initiated by the backup scantape command.

SaveDb

Initiated by the backup savedb command.

RestoreDb

Initiated by the backup restoredb command.

<status>

Indicates the job's current status in one of the following messages. If no message appears, the job is either still pending or has finished.

number Kbytes, volume *volume_name*

For a running dump operation, indicates the number of kilobytes copied to tape or a backup data file so far, and the volume currently being dumped.

number Kbytes, restore.volume

For a running restore operation, indicates the number of kilobytes copied into AFS from a tape or a backup data file so far.

[abort requested]

The backup kill command was issued, but the termination signal has yet to reach the Tape Coordinator.

[abort sent]

The operation is canceled by the backup kill command. Once the Backup System removes an operation from the queue or stops it from running, it no longer appears at all in the output from the command.

[butc contact lost]

The backup command interpreter cannot reach the Tape Coordinator. The message can mean either that the Tape Coordinator handling the operation was terminated or failed while the operation was running, or that the connection to the Tape Coordinator timed out.

[done]

The Tape Coordinator has finished the operation.

[drive wait]

The operation is waiting for the specified tape drive to become free.

[operator wait]

The Tape Coordinator is waiting for the backup operator to insert a tape in the drive.

Examples

The following example shows that two restore operations and one dump operation are running (presumably on different Tape Coordinators) and that the backup command interpreter's tokens expire on 22 April 1999 at 10:45 am:

```
backup&gt; jobs
Job 1: Restore, 1306 Kbytes, restore.volume
Job 2: Dump (user.sunday1), 34 Kbytes, volume user.pat.backup
Job 3: Restore, 2498 Kbytes, restore.volume
       04/22/1999 10:45: TOKEN EXPIRATION
```

Privilege Required

None. However, queuing any operation requires privilege, and it is possible to issue this command only within the interactive session in which the jobs are queued.

See Also

"backup(8)" on page 531, "backup_interactive(8)" on page 591, "backup_kill(8)" on page 597, "backup_quit(8)" on page 613

Copyright

backup kill

Name

`backup kill` — Terminates a pending or running operation

Synopsis

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

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

Description

The backup kill command dequeues a Backup System operation that is pending, or terminates an operation that is running, in the current interactive session. It is available only in interactive mode. If the issuer of the backup interactive command included the -localauth flag, the -cell argument, or both, then those settings apply to this command also.

To terminate a dump operation, specify either the dump name (*volume_set_name.dump_level_name*) or its job ID number, which appears in the output from the backup jobs command. To terminate any other type of operation, provide the job ID number.

The effect of terminating an operation depends on the type and current state of the operation:

- If an operation is still pending, the Tape Coordinator removes it from the queue with no other lasting effects.
- If the Tape Coordinator is unable to process the termination signal before an operation completes, it simply confirms the operation's completion. The operator must take the action necessary to undo the effects of the incorrect operation.
- If a tape labeling operation is running, the effect depends on when the Tape Coordinator receives the termination signal. The labeling operation is atomic, so it either completes or does not begin at all. Use the backup readlabel command to determine if the labeling operation completed, and reissue the backup labeltape command to overwrite the incorrect label if necessary.
- If a tape scanning operation is running, it terminates with no other effects unless the -dbadd flag was included on the backup command. In that case, the Backup System possibly has already written new Backup Database records to represent dumps on the scanned tape. If planning to restart the scanning operation, first locate and remove the records created during the terminated operation: a repeated backup scantape operation exits automatically when it finds that a record that it needs to create already exists.
- If a dump operation is running, all of the volumes written to the tape or backup data file before the termination signal is received are complete and usable. If the operation is restarted, the Backup System performs all the dumps again from scratch, and assigns a new dump ID number. If writing the new dumps to the same tape or file, the operator must relabel it first if the interrupted dump is not expired. If writing the new dump to a different

tape or file, the operator can remove the dump record associated with the interrupted dump to free up space in the database.

- If a restore operation is running, 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 termination signal comes before the Backup System has accessed all of the necessary tapes, each volume is only partially written and is never brought online. It is best to restart the restore operation from scratch to avoid possible inconsistencies. See also "CAUTIONS" on page .

Cautions

It is best not to issue the backup kill command against restore operations. If the termination signal interrupts a restore operation as the Backup System is overwriting an existing volume, it is possible to lose the volume entirely (that is, to lose both the contents of the volume as it was before the restore and any data that was restored before the termination signal arrived). The data being restored still exists on the tape, but some data can be lost permanently.

Options

-id <*job ID or dump set name*>

Identifies the backup operation to terminate. Provide one of two types of values:

- The operation's job ID number, as displayed in the output of the backup jobs command.
- For a dump operation, either the job ID number or a dump name of 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 pathname at which the volume set is being dumped. The dump name appears in the output of the backup jobs command along with the job ID number.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command terminates the operation with job ID 5:

```
backup&gt; kill 5
```

The following command terminates the dump operation called `user.sunday1`:

```
backup&gt; kill user.sunday1
```

Privilege Required

The issuer must have the privilege required to initiate the operation being cancelled. Because this command can be issued only within the interactive session during which the operation was initiated, the required privilege is essentially guaranteed.

See Also

"backup(8)" on page 531, "backup_interactive(8)" on page 591, "backup_jobs(8)" on page 593

Copyright

backup labeltape

Name

`backup labeltape` — Creates the magnetic label on a tape

Synopsis

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

backup la [-n <*AFS tape name, defaults to NULL*>] [-s <*tape size in Kbytes, defaults to size in tapeconfig*>] [-po <*TC port offset*>] [-pn <*permanent tape name*>] [-l] [-c <*cell name*>] [-h]

Description

The backup labeltape command creates a magnetic label, readable by the Backup System, at the beginning of a tape. The label records the tape's name (either a *permanent name*, or an *AFS tape name* that reflects the tape's contents in a prescribed format) and its capacity.

(If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file on the Tape Coordinator machine associated with the specified port offset, then the backup command writes label information to the first 16 KB block in the backup data file listed for that port offset in the Tape Coordinator's */usr/afs/backup/tapeconfig* file, rather than at the beginning of a tape. For the sake of clarity, the following text refers to tapes only, but the Backup System handles backup data files in much the same way.)

Relabeling a tape that already contains AFS backup data effectively makes the data unusable, because the command removes the Backup Database record of the complete dump set of which the tape is a part. Use this command to enable recycling of a tape that contains unexpired dumps that are not actually still needed.

To write a permanent name on the label, include the -pname argument to specify a string of up to 32 characters. The permanent name persists until the -pname argument is again included on the backup labeltape command, regardless of the tape's contents and of how often the tape is otherwise relabeled or recycled. Include this argument or the -name argument, but not both. If this argument is included, the AFS tape name is set to `<NULL>`. The permanent name is set to `<NULL>` if this argument is omitted and no permanent name already exists.

The issuer must ensure that a permanent name is unique among the tapes used for AFS backup in the cell, because the backup command interpreter does not verify that another tape does not already have the same permanent name. When a tape has a permanent name, the Backup System uses it instead of the AFS tape name in most prompts and when referring to the tape in output from backup commands. The permanent name appears in the `tape name` field of the output from the backup readlabel command.

To write an AFS tape name on the label, provide a value for the -name argument in the required format described in "OPTIONS" on page . Include the -name argument or the -pname argument, but not both. If this argument is omitted, the AFS tape name is set to `<NULL>`, but the Backup System automatically assigns the appropriate name when the tape is used in a future

backup dump or backup savedb operation. The AFS tape name appears in the `AFS tape name` field of the output from the backup readlabel and backup scantape commands.

The backup command interpreter does not accept the -name argument if the tape already has a permanent name. To erase a tape's permanent name, provide a null value to the -pname argument by issuing the following command:

```
% backup labeltape -pname ""
```

To record the tape's capacity on the label, specify a number of kilobytes as the -size argument. If the argument is omitted the first time a tape is labeled, the Backup System records the default tape capacity recorded for the specified port offset in the */usr/afs/backup/tapeconfig* file on the Tape Coordinator machine. Subsequently, the value in the size field persists until the -size argument is again included on the backup labeltape command.

To determine how much data can be written to a tape during a backup dump or backup savedb operation, the Tape Coordinator reads the capacity recorded on the tape's label (or uses the value associated with its port offset in the */usr/afs/backup/tapeconfig* file, if the tape was never labeled). For further description, see the backup dump reference page.

The Tape Coordinator's default response to this command is to access the tape by invoking the `MOUNT` instruction in the local */usr/afs/backup/CFG_device_name* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *CFG_device_name* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, the Tape Coordinator invokes the `MOUNT` instruction or prompts the operator.

Options

-name <*AFS tape name*>

Specifies the AFS tape name to record on the label. Include this argument or the -pname argument, but not both. If this argument is omitted, the AFS tape name is set to `<NULL>`. If this argument is provided, it must have the following format:

```
&lt;volume_set_name&gt;.&lt;dump_level_name&gt;.&lt;tape_index&gt;
```

for the tape to be acceptable for use in a future backup dump operation. The <volume_set_name> must match the volume set name of the initial dump to be written to the tape, <dump_level_name> must match the last element of the dump level pathname at which the volume set will be dumped, and <tape_index> indicates the order of the tape in the dump set (indexing begins with `1`). To disable this type of name checking, include the `NAME_CHECK NO` instruction in the *CFG_device_name* file.

For the tape to be acceptable for use in a future backup savedb operation, the value specified for the -name argument must have the following format:

```
Ubik_db_dump.&lt;tape_index&gt;
```

where <tape_index> indicates the order of the tape in the set of tapes that house the Backup Database dump; indexing begins with `1` (one).

-size <*tape size*>

Specifies the tape capacity to record on the label. Provide an integer value followed by a letter that indicates units, with no intervening space. A unit value of `k` or `K` indicates kilobytes, `m` or `M` indicates megabytes, and `g` or `G` indicates gigabytes. If the units letter is omitted, the default is kilobytes.

If this argument is omitted the first time a tape is labeled, the Backup System records the capacity that is associated with the specified port offset in the */usr/afs/backup/tapeconfig* file on the Tape Coordinator machine. The value recorded the first time then persists until the -size argument is provided on a future issuance of the command.

-portoffset <*TC port offset*>

Specifies the port offset number of the Tape Coordinator handling the tape for this operation.

-pname <*permanent tape name*>

Specifies the permanent name to record on the label. It can be up to 32 characters in length, and include any alphanumeric characters. Avoid metacharacters that have a special meaning to the shell, to avoid having to mark them as literal in commands issued at the shell prompt.

Include this argument or the -name argument, but not both. If this argument is provided, the AFS tape name is set to `<NULL>`. If this argument is omitted, any existing permanent name is retained.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command records the AFS tape name `user.monthly.1` on the label of the tape in the device with port offset 3:

```
% backup labeltape -name user.monthly.1 -portoffset 3
```

The following three commands are equivalent in effect: they all record a capacity of 2 GB on the label of the tape in the device with port offset 4. They set the AFS tape name to `<NULL>` and leave the permanent name unchanged.

```
% backup labeltape -size 2g -portoffset 4
% backup labeltape -size 2048M -portoffset 4
% backup labeltape -size 2097152 -portoffset 4
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_readlabel(8)" on page 615, "butc(8)" on page 767

Copyright

backup listdumps

Name

`backup listdumps` — Displays the dump hierarchy from the Backup Database

Synopsis

backup listdumps [-localauth] [-cell <*cell name*>] [-help]

backup listd [-l] [-c <*cell name*>] [-h]

Description

The backup listdumps command displays the dump hierarchy from the Backup Database.

Options

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output displays the complete dump hierarchy and indicates the relationship between full and incremental dump levels. Full dump levels appear at the left margin. The hierarchy can include more than one full dump level; each one 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 can be either full or incremental. Since multiple incremental dump levels can share the same parent, an incremental dump level is not always directly below its parent; the amount of indentation indicates the parent/child relationship.

If a dump level has an associated expiration date, it appears along with the level name. Absolute expiration dates appear in the format

```
&lt;dump_level&gt; expires at &lt;day month date time year&gt;
```

and relative expiration dates in the format

```
&lt;dump_level&gt; expires in {&lt;years&gt;y | &lt;months&gt;m | &lt;days&gt;d}
```

to indicate the number of years, months, days, or combination of the three after creation a dump expires when created at this level.

Examples

The following example depicts six dump hierarchies. The expiration date 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.

```
% 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
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_adddump(8)" on page 535, "backup_deldump(8)" on page 553

Copyright

backup listhosts

Name

`backup listhosts` — Lists Tape Coordinators registered in the Backup Database

Synopsis

backup listhosts [-localauth] [-cell <*cell name*>] [-help]

backup listh [-l] [-c <*cell name*>] [-h]

Description

The backup listhosts command displays the Backup Database record of the port offset numbers defined for Tape Coordinator machines. A Tape Coordinator must have an entry in the list to be available for backup operations.

The existence of an entry does not necessarily indicate that the Tape Coordinator process (butc) is currently running at that port offset. To check, issue the backup status command.

Options

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The hostname of the machine housing the Tape Coordinator. The format of this name depends on the hostname format used when the backup addhost command was issued.
- The Tape Coordinator's port offset number.

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

Examples

The following example shows the result of the command in the ABC Corporation cell:

```
% backup listhosts
Tape hosts:
    Host backup1.abc.com, port offset 0
    Host backup1.abc.com, port offset 1
    Host backup3.abc.com, port offset 4
    Host backup2.abc.com, port offset 3
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_addhost(8)" on page 539, "backup_delhost(8)" on page 559, "backup_status(8)" on page 639

Copyright

backup listvolsets

Name

`backup listvolsets` — Lists volume set entries from the Backup Database

Synopsis

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

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

Description

The backup listvolsets command displays the Backup Database records for either

- All volume sets and their volume entries, if the -name argument is omitted.
- The volume set specified by the -name argument, along with its volume entries.

Options

-name <*volume set name*>

Names the volume set to display. If this argument is omitted, the output lists all volume sets defined in the Backup Database.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The entry for each volume set begins with the `Volume set` header and the volume set's name. A temporary volume set's name is followed by the string `(temporary)`. Each volume entry follows on a separate line, indicating the entry's index number and the server, partition, and volume names it matches. The output uses the metacharacter notation described on the backup

addvolentry reference page. Use the index number to identify volume entries when deleting them with the backup delvolentry command.

Examples

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

```
% backup listvolsets
Volume set user:
    Entry   1: server .*, partition .*, volumes: user.*\.backup
Volume set sun
    Entry   1: server .*, partition .*, volumes: sun4x_55\..*
    Entry   2: server .*, partition .*, volumes: sun4x_56\..*
Volume set rs
    Entry   1: server .*, partition .*, volumes: rs_aix42\..*
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_addvolentry(8)" on page 541, "backup_addvolset(8)" on page 545, "backup_delvolentry(8)" on page 561, "backup_delvolset(8)" on page 563

Copyright

backup quit

Name

`backup quit` — Leaves interactive mode

Synopsis

quit [-help]

q [-h]

Description

The backup quit command exits interactive mode, returning the issuer to the regular shell prompt at which the backup or backup interactive command was issued to enter interactive mode. The command has no effect when issued outside interactive mode. Issuing the Ctrl-D command also exits interactive mode.

Cautions

To exit interactive mode, all jobs must be completed. Use the backup jobs command to list any jobs currently pending or executing, and the backup kill command to terminate them as necessary.

Options

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command exits interactive mode:

```
backup&gt; quit
%
```

Privilege Required

None

See Also

"backup(8)" on page 531, "backup_interactive(8)" on page 591, "backup_jobs(8)" on page 593, "backup_kill(8)" on page 597

Copyright

backup readlabel

Name

`backup readlabel` — Reads and displays a tape's label

Synopsis

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

backup rea [-p <*TC port offset*>] [-l] [-c <*cell name*>] [-h]

Description

The backup readlabel command displays information from the magnetic tape label of a tape. The information includes the tape's name (either a *permanent name*, or an *AFS tape name* that reflects the tape's contents in a prescribed format) and its capacity.

If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file associated with the specified port offset, then the backup readlabel command reads the label information from the first 16 KB block in the backup data file listed for that port offset in the Tape Coordinator's */usr/afs/backup/tapeconfig* file, rather than from the beginning of a tape.

The Tape Coordinator's default response to this command is to access the tape by invoking the `MOUNT` instruction in the local */usr/afs/backup/CFG_device_name* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *CFG_device_name* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, the Tape Coordinator invokes the `MOUNT` instruction or prompts the operator.

Options

-portoffset <*TC port offset*>

Specifies the port offset number of the Tape Coordinator handling the tapes for this operation.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

Output from this command appears in both the shell window where the command is issued, and in the Tape Coordinator window.

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

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

Otherwise, the output in the shell window has the following format:

```
Tape read was labelled: &lt;tape name&gt; (&lt;dump id&gt;)
     size: &lt;size&gt; Kbytes
```

where <tape name> is the permanent name if the tape has one, or the AFS tape name otherwise. The <dump ID> is dump ID of the initial dump on the tape, and <size> is the recorded capacity of the tape in kilobytes.

The output in the Tape Coordinator windows is bounded by an underlined `Tape label` header at the top, and the following string at the bottom:

```
-- End of tape label --
```

In between are lines reporting the following information:

tape name

The permanent name assigned by using the -pname argument of the backup labeltape command. This name remains on the tape until that argument is used again, no matter how many times the tape is recycled or otherwise relabeled. If the tape does not have a permanent name, the value `<NULL>` appears in this field.

AFS tape name

A tape name in one of the following prescribed formats. The Backup System automatically writes the appropriate AFS tape name to the label as part of a backup dump or backup savedb operation, or the operator can assign it with the -name argument to the backup labeltape command.

- *volume_set_name.dump_level_name.tape_index*, if the tape contains volume data. The *volume_set_name* is the name of the volume set that was dumped to create the initial dump in the dump set of to which this tape belongs; *dump_level_name* is the last pathname element 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.

- `Ubik.db.dump.`*tape_index* if the tape contains a dump of the Backup Database, created with the backup savedb command. The *tape_index* is the ordinal of the tape in the dump set.
- `<NULL>` if the tape has no AFS tape name. This is normally the case if the -name argument was not included the last time the backup labeltape command was used on this tape, and no data has been written to it since.

creationTime

The date and time at which the Backup System started performing the dump operation that created the initial dump.

cell

The cell in which the dump set was created. This is the cell whose Backup Database contains a record of the dump set.

size

The tape's capacity (in kilobytes) as recorded on the label, rather than the amount of data on the tape. The value is assigned by the -size argument to the backup labeltape command or derived from the */usr/afs/backup/tapeconfig* file on the Tape Coordinator machine, not from a measurement of the tape.

dump path

The dump level of the initial dump in the dump set.

dump id

The dump ID number of the initial dump in the dump set, as recorded in the Backup Database.

useCount

The number of times a dump has been written to the tape, or it has been relabeled.

The message `ReadLabel: Finished` indicates the completion of the output.

Examples

The following example shows the output for the tape with permanent name `oct.guest.dump` and capacity 2 MB, expressed in kilobyte units (2097152 equals 2 times 10242).

```
% backup readlabel -portoffset 6
Tape read was labelled: oct.guest.dump (907215000)
     size: 2097152 Kbytes
```

The output in the Tape Coordinator window reads:

```
Tape label
```

```
----------
tape name = oct.guest.dump
AFS tape name = guests.monthly.3
creationTime = Thu Oct 1 00:10:00 1998
cell = abc.com
size = 2097152 Kbytes
dump path = /monthly
dump id = 907215000
useCount = 5
---- End of tape label ----
```

The following example is for a tape that does not have a permanent tape.

```
% backup readlabel -portoffset 6
Tape read was labelled: guests.monthly.2 (909899900)
     size: 2097152 Kbytes
```

The output in the Tape Coordinator window reads:

```
Tape label
----------
tape name = &lt;NULL&gt;
AFS tape name = guests.monthly.2
creationTime = Sun Nov 1 00:58:20 1998
cell = abc.com
size = 2097152 Kbytes
dump path = /monthly
dump id = 909899900
useCount = 1
---- End of tape label ----
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_labeltape(8)" on page 601, "butc(8)" on page 767

Copyright

backup restoredb

Name

`backup restoredb` — Restores a saved copy of the Backup Database

Synopsis

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

backup res [-p <*TC port offset*>] [-l] [-c <*cell name*>] [-h]

Description

The backup restoredb command restores to the Backup Server machine's local disk a version of the Backup Database previously written to tape by using the backup savedb command.

(If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file associated with the specified port offset, then the backup restoredb command restores data from the backup data file listed for that port offset in the Tape Coordinator's */usr/afs/backup/tapeconfig* file, instead of from tape. For the sake of clarity, the following text refers to tapes only, but the Backup System handles backup data files in much the same way.)

The most common reason to run this command is to replace a corrupted or otherwise damaged Backup Database; use the backup dbverify command to determine the database's status. The command can also be used to restore records that were removed from the database when the -archive argument was included on a previous backup savedb command.

The command completely overwrites the existing Backup Database records for volume sets, Tape Coordinators, and the dump hierarchy with the corresponding information from the saved version. It does not overwrite existing dump records, but instead interleaves the records from the copy being restored. If both the existing database (on the Backup Server machine's disk) and the copy being restored include a record about the same dump, the Backup System retains the one in the existing database.

The Tape Coordinator's default response to this command is to access the first tape it needs by invoking the `MOUNT` instruction in the local */usr/afs/backup/CFG_device_name* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *CFG_device_name* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, or is the wrong tape, the Tape Coordinator invokes the `MOUNT` instruction or prompts the operator. It also invokes the `MOUNT` instruction or prompts for any additional tapes needed to complete the restore operation; the backup operator must arrange to provide them.

Cautions

If the database is corrupted, do not attempt to restore a saved database on top of it. Instead, use the instructions for repairing a corrupted database in the *IBM AFS Administration Guide* chapter

about performing backup operations.

Options

-portoffset <*TC port offset*>

Specifies the port offset number of the Tape Coordinator handling the tapes for this operation.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

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

```
% backup restoredb
```

Privilege Required

The issuer must be listed in the `/usr/afs/etc/UserList` file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_dbverify(8)" on page 549, "backup_savedb(8)" on page 625, "butc(8)" on page 767

IBM AFS Administration Guide

Copyright

backup savedb

Name

`backup savedb` — Creates a saved copy of the Backup Database

Synopsis

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

backup sa [-p <*TC port offset*>] [-a <*date time*>+] [-l] [-c <*cell name*>] [-h]

Description

The backup savedb command creates a backup copy of the entire Backup Database and writes it to the tape in the device controlled by the Tape Coordinator indicated with the -portoffset argument. If the database is damaged (as reported by the backup dbverify command), this command repairs as much of the corruption as possible as it creates the saved copy. The Backup Server creates a dump record for the saved database in the Backup Database (but in the disk version of the database only, not in the version written to tape).

If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file associated with the specified port offset, then the backup savedb command dumps the database copy to the backup data file listed for that port offset in the Tape Coordinator's */usr/afs/backup/tapeconfig* file, instead of to tape. For the sake of clarity, the following text refers to tapes only, but the Backup System handles backup data files in much the same way.

If the -archive flag is provided, after writing the saved copy of the database the Backup System truncates the copy of the database on disk by deleting volume dump records with timestamps prior to the specified date and time (it does not delete the dump records created by previous backup savedb commands, however).

If the tape to which the database copy is written has an AFS tape name, it must be `Ubik_db_dump.1` or `<NULL>`. Any permanent name is acceptable.

The Tape Coordinator's default response to this command is to access the first tape by invoking the `MOUNT` instruction in the local */usr/afs/backup/CFG_device_name* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *CFG_device_name* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, the Tape Coordinator invokes the `MOUNT` instruction or prompts the operator. It also invokes the `MOUNT` instruction or prompts for any additional tapes needed to complete the operation; the backup operator must arrange to provide them.

Options

-portoffset <*TC port offset*>

Specifies the port offset number of the Tape Coordinator handling the tapes for this operation.

-archive <*date time*>+

Specifies a date and time; volume dump records with earlier timestamps are deleted from the disk copy of the Backup Database after the Backup System dumps the database (a dump's timestamp appears in the `created` field of the output from the backup dumpinfo command). However, if a dump set contains any dump created after the specified date, none of the dump records associated with the dump set are deleted. Dump records for previous dumps of the database (created with the backup savedb command) are never deleted; use the backup deletedump command to remove them.

Provide one of two values:

- The string `NOW` to indicate the current date and time, in which case the Backup System deletes all dump records except those for dumps of the Backup Database itself.
- A date value in the format *mm/dd/yyyy* [*hh:MM*]. The month (*mm*), day (*dd*), and year (*yyyy*) are required, and valid values for the year range from `1970` to `2037`; higher values are not valid because the latest possible date in the standard UNIX representation is in February 2038. The Backup System automatically reduces any later date to the maximum value.

  The hour and minutes (*hh:MM*) are optional, but if provided must be in 24-hour format (for example, the value `14:36` represents 2:36 p.m.). If omitted, the time defaults to 59 seconds after midnight (00:00:59 hours). Similarly, the backup command interpreter automatically adds 59 seconds to any time value provided. In both cases, adding 59 seconds compensates for how the Backup Database and backup dumpinfo command represent dump creation times in hours and minutes only. That is, the Database records a creation timestamp of `20:55` for any dump created between 20:55:00 and 20:55:59. Automatically adding 59 seconds to a time thus includes the records for all dumps created during that minute.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example writes a copy of the Backup Database to the tape device controlled by the Tape Coordinator with port offset 1:

```
% backup savedb -portoffset 1
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_dbverify(8)" on page 549, "backup_restoredb(8)" on page 621, "butc(8)" on page 767

Copyright

backup scantape

Name

`backup scantape` — Extracts dump information from a tape

Synopsis

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

backup sc [-d] [-p <*TC port offset*>] [-l] [-c <*cell name*>] [-h]

Description

The backup scantape command extracts information from the dump labels and volume headers on the tape in the device controlled by the Tape Coordinator indicated by the -portoffset argument. The Tape Coordinator displays the information for each volume in its window as soon as it extracts it (rather than waiting until it has scanned the entire tape).

(If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file associated with the specified port offset, then the backup scantape command extracts dump information from the backup data file named in that port offset's entry in the */usr/afs/backup/tapeconfig* file on the Tape Coordinator machine, rather than from a tape. For the sake of clarity, the following text refers to tapes only, but the Backup System handles backup data files in much the same way.)

If the -dbadd flag is provided, the backup scantape command creates new dump and volume records in the Backup Database for the scanned information. However, if it finds that a record already exists in the database for the same dump, it terminates the scanning operation.

The scanning operation works only on tapes containing volume data. The command fails with an error message if the tape contains a copy of the Backup Database (was created with the backup savedb command, or has the AFS tape name `Ubik_db_dump.1`).

The Tape Coordinator's default response to this command is to access the tape by invoking the `MOUNT` instruction in the *CFG_device_name* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *CFG_device_name* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, the Tape Coordinator invokes the `MOUNT` instruction or prompts the operator.

To terminate a tape scanning operation in interactive mode, issue the backup kill command. In noninteractive mode, the only choice is to use a termination signal such as Ctrl-C to halt the Tape Coordinator completely.

Cautions

A scanning operation does not have to begin with the first tape in a dump set, but the Backup System can process tapes only in sequential order after the initial tape provided. The Tape Coordinator automatically requests any subsequent tapes by invoking the `MOUNT` instruction in

the local */usr/afs/backup/CFG_device_name* file, or by prompting the operator if there is no `MOUNT` instruction.

The Tape Coordinator's success in scanning a tape that is corrupted or damaged depends on the extent of the damage and what type of data is corrupted. It can almost always scan the tape successfully up to the point of damage. If the damage is minor, the Tape Coordinator can usually skip over it and scan the rest of the tape, but more major damage can prevent further scanning. Because a scanning operation can start on any tape in a dump set, damage on one tape does not prevent scanning of the others in the dump set. However, it is possible to scan either the tapes that precede the damaged one or the ones that follow it, but not both.

If a tape is relabeled with the backup labeltape command, it is not possible to recover data from it for the purposes of rebuilding the Backup Database.

If the -dbadd flag is included on the command, it is best not to terminate the tape scanning operation before it completes (for example, by issuing the backup kill command in interactive mode). The Backup System writes a new record in the Backup Database for each dump as soon as it scans the relevant information on the tape, and so it possibly has already written new records. If the operator wants to rerun the scanning operation, he or she must locate and remove the records created during the terminated operation: the second operation exits automatically if it finds that a record that it needs to create already exists.

If the -dbadd flag is included and the first tape provided is not the first tape in the dump set, the following restrictions apply:

- If the first data on the tape is a continuation of a volume that begins on the previous (unscanned) tape in the dump set, the Backup System does not add a record for that volume to the Backup Database.
- The Backup System must read the marker that indicates the start of an appended dump to add database records for the volumes in it. If the first volume on the tape belongs to an appended dump, but is not immediately preceded by the appended-dump marker, the Backup System does not create a Backup Database record for it or any subsequent volumes that belong to that appended dump.

Options

-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 <*TC port offset*>

Specifies the port offset number of the Tape Coordinator handling the tapes for this operation.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL

Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

For every dump on a tape, the backup scantape command displays in the Tape Coordinator window the dump label and the volume header of each volume in the dump. If a dump spans more than one tape, the dump label does not repeat at the beginning of subsequent tapes.

A dump label contains the following fields, which are the same as in the output from the backup readlabel command:

tape name

The permanent name assigned by using the -pname argument of the backup labeltape command. This name remains on the tape until that argument is used again, no matter how many times the tape is recycled or otherwise relabeled. If the tape does not have a permanent name, the value <NULL> appears in this field.

AFS tape name

A tape name in one of the following prescribed formats. The Backup System automatically writes the appropriate AFS tape name to the label as part of a backup dump operation, or the operator can assign it with the -name argument to the backup labeltape command.

- *volume_set_name.dump_level_name.tape_index*, if the tape contains volume data. The *volume_set_name* is the name of the volume set that was dumped to create the initial dump in the dump set of which this tape is a part; *dump_level_name* is the last pathname element 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.
- <NULL> if the tape has no AFS tape name. This is normally the case if the -name argument was not included the last time the backup labeltape command was used on this tape, and no data has been written to it since.

creationTime

The date and time at which the Backup System started performing the dump operation that created the initial dump.

cell

The cell in which the dump set was created. This is the cell whose Backup Database contains a record of the dump set.

size

The tape's capacity (in kilobytes) as recorded on the label, rather than the amount of data on the tape. The value is assigned by the -size argument to the backup labeltape command or derived from the */usr/afs/backup/tapeconfig* file on the Tape Coordinator machine, not from a measurement of the tape.

dump path

The dump level of the initial dump in the dump set.

dump id

The dump ID number of the initial dump in the dump set, as recorded in the Backup Database.

useCount

The number of times a dump has been written to the tape, or it has been relabeled.

The volume header contains the following fields:

volume name

The volume name, complete with a `.backup` or `.readonly` extension, if appropriate.

volume ID

The volume's volume ID.

dumpSetName

The dump to which the volume belongs. The dump name is of the form *volume_set_name.dump_level_name* and matches the name displayed in the dump label.

dumpID

The dump ID of the dump named in the `dumpSetName` field.

level

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 a dump performed at an incremental level to include a full dump of an individual volume if the volume was omitted from previous dumps.

parentID

The dump ID number of `dumpSetName`'s parent dump. It is `0` if the value in the `level` field is `0`.

endTime

Is always `0`; it is reserved for internal use.

cloneDate

The date and time at which the volume was created. For a backup or read-only volume, this represents the time at which it was cloned from its read / write source. For a read / write volume, it indicates the time at which the Backup System locked the volume for purposes of including it in the dump named in the `dumpSetName` field.

The message `Scantape: Finished` indicates the completion of the output.

In normal circumstances, the Backup System writes a marker to indicate that a volume is the last one on a tape, or that the volume continues on the next tape. However, if a backup operation terminated abnormally (for example, because the operator terminated the Tape Coordinator by issuing the Ctrl-C command during the operation), then there is no such marker. Some very early versions of the Backup System also did not write these markers. If a tape does not conclude with one of the expected markers, the Tape Coordinator cannot determine if there is a subsequent tape in the dump set and so generates the following message in its window:

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

Examples

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

```
% backup scantape
Dump label
----------
tape name = monthly_guest
AFS tape name = guests.monthly.3
creationTime =  Mon Feb  1 04:06:40 1999
cell = abc.com
size = 2150000 Kbytes
dump path = /monthly
dump id = 917860000
useCount = 44
-- End of dump label --
-- volume --
volume name: user.guest10.backup
volume ID 1937573829
dumpSetName: guests.monthly
dumpID 917860000
level 0
parentID 0
endTime 0
clonedate Mon Feb  1 03:03:23 1999
-- volume --
```

```
volume name: user.guest11.backup
volume ID 1938519386
dumpSetName: guests.monthly
dumpID 917860000
level 0
parentID 0
endTime 0
clonedate Mon Feb  1 03:05:15 1999
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_dump(8)" on page 571, "backup_dumpinfo(8)" on page 581, "butc(8)" on page 767

Copyright

backup setexp

Name

`backup setexp` — Sets the expiration date for existing dump levels.

Synopsis

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

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

Description

The backup setexp command sets or changes the expiration date associated with each specified dump level, which must already exist in the dump hierarchy.

Use the -expires argument to associate an expiration date with each dump level. When the Backup System subsequently creates a dump at the dump level, it uses the specified value to derive the dump's expiration date, which it records on the label of the tape (or backup data file). The Backup System refuses to overwrite a tape until after the latest expiration date of any dump that the tape contains, unless the backup labeltape command is used to relabel the tape. If a dump level does not have an expiration date, the Backup System treats dumps created at the level as expired as soon as it creates them.

(Note that the Backup System does not automatically remove a dump's record from the Backup Database when the dump reaches its expiration date, but only if the tape that contains the dump is recycled or relabeled. To remove expired and other obsolete dump records, use the backup deletedump command.)

Define either an absolute or relative expiration date:

- An absolute expiration date defines the month / day / year (and, optionally, hour and minutes) at which a dump expires. If the expiration date predates the dump creation time, the Backup System immediately treats the dump as expired.
- A relative date defines the number of years, months, or days (or a combination of the three) after the dump's creation that it expires. When the Backup System creates a dump at the dump level, it calculates an actual expiration date by adding the relative date to the start time of the dump operation.

If the command is used to change an existing expiration date associated with a dump level, the new date applies only to dumps created after the change. Existing dumps retain the expiration date assigned at the time they were created.

Options

-dump <*dump level name*>+

Specifies the full pathname of each dump level to assign the expiration date specified by the -expires argument.

-expires <*expiration date*>+

Defines the absolute or relative expiration date to associate with each dump level named by the -dump argument. Absolute expiration dates have the following format:

```
[at] {NEVER | &lt;mm&gt;/&lt;dd&gt;/&lt;yyyy&gt; [&lt;hh&gt;:&lt;MM&gt;] }
```

where the optional word at is followed either by the string `NEVER`, which indicates that dumps created at the dump level never expire, or by a date value with a required portion (<mm> for month, <dd> for day, and <yyyy> for year) and an optional portion (<hh> for hours and <MM> for minutes).

Omit the <hh>:<MM> portion to use the default of midnight (00:00 hours), or provide a value in 24-hour format (for example, `20:30` is 8:30 p.m.). Valid values for the year range from `1970` to `2037`; higher values are not valid because the latest possible date in the standard UNIX representation is in February 2038. The command interpreter automatically reduces later dates to the maximum value.

Relative expiration dates have the following format:

```
[in] [&lt;years&gt;y] [&lt;months&gt;m] [&lt;days&gt;d]
```

where the optional word in is followed by at least one of a number of years (maximum `9999`) followed by the letter `y`, a number of months (maximum `12`) followed by the letter `m`, or a number of days (maximum `31`) followed by the letter `d`. If providing more than one of the three, list them in the indicated order. If the date that results from adding the relative expiration value to a dump's creation time is later than the latest possible date in the UNIX time representation, the Backup System automatically reduces it to that date.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example associates an absolute expiration date of 10:00 p.m. on 31 December 1999 with the dump level `/1998/december`:

```
% backup setexp -dump /1998/december -expires at 12/31/1999 22:00
```

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

```
% backup setexp -dump /monthly/week1 /monthly/week -expires 7d
```

Privilege Required

The issuer must be listed in the *`/usr/afs/etc/UserList`* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "backup_adddump(8)" on page 535, "backup_deldump(8)" on page 553, "backup_listdumps(8)" on page 605

Copyright

backup status

Name

`backup status` — Reports a Tape Coordinator's status

Synopsis

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

backup st [-p <*TC port offset*>] [-l] [-c <*cell name*>] [-h]

Description

The backup status command displays which operation, if any, the indicated Tape Coordinator is currently executing.

Options

-portoffset <*TC port offset*>

Specifies the port offset number of the Tape Coordinator for which to report the status.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The following message indicates that the Tape Coordinator is not currently performing an operation:

```
Tape coordinator is idle
```

Otherwise, the output includes a message of the following format for each running or pending operation:

```
   Task &lt;task_ID&gt;:  &lt;operation&gt;:   &lt;status&gt;
```

where

<task_ID>

Is a task identification number assigned by the Tape Coordinator. It begins with the Tape Coordinator's port offset number.

<operation>

Identifies the operation the Tape Coordinator is performing, which is initiated by the indicated command:

Dump

The backup dump command.

Restore

The backup diskrestore, backup volrestore, or backup volsetrestore commands.

Labeltape

The backup labeltape command.

Scantape

The backup scantape command.

SaveDb

The backup savedb command.

RestoreDb

The backup restoredb command.

<status>

Indicates the job's current status in one of the following messages.

number Kbytes transferred, volume *volume_name*

For a running dump operation, indicates the number of kilobytes copied to tape or a backup data file so far, and the volume currently being dumped.

number Kbytes, restore.volume

For a running restore operation, indicates the number of kilobytes copied into AFS from a tape or a backup data file so far.

[abort requested]

The backup kill command was issued, but the termination signal has yet to reach the Tape Coordinator.

[abort sent]

The operation is canceled by the backup kill command. Once the Backup System removes an operation from the queue or stops it from running, it no longer appears at all in the output from the command.

[butc contact lost]

The backup command interpreter cannot reach the Tape Coordinator. The message can mean either that the Tape Coordinator handling the operation was terminated or failed while the operation was running, or that the connection to the Tape Coordinator timed out.

[done]

The Tape Coordinator has finished the operation.

[drive wait]

The operation is waiting for the specified tape drive to become free.

[operator wait]

The Tape Coordinator is waiting for the backup operator to insert a tape in the drive.

If the Tape Coordinator is communicating with an XBSA server (a third-party backup utility that implements the Open Group's Backup Service API [XBSA]), the following message appears last in the output:

```
&lt;XBSA_program&gt; Tape coordinator
```

where <XBSA_program> is the name of the XBSA-compliant program.

Examples

The following example shows that the Tape Coordinator with port offset 4 has so far dumped about 1.5 MB of data for the current dump operation, and is currently dumping the volume named `user.pat.backup`:

```
% backup status -portoffset 4
Task 4001:  Dump:   1520 Kbytes transferred,  volume user.pat.backup
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server is running, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"backup(8)" on page 531, "butc(8)" on page 767

Copyright

backup volinfo

Name

`backup volinfo` — Displays a volume's dump history from the Backup Database

Synopsis

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

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

Description

The backup volinfo command displays a dump history of the specified volume, reporting information such as the date on which the volume was dumped and the tapes that contain it. Include the `.backup` extension on the volume name if the backup version of the volume was dumped.

Options

-volume <*volume name*>

Names the volume for which to display the dump history. Include the `.backup` or `.readonly` extension if the backup or read-only version of the volume was dumped.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output includes a line for each Backup Database dump record that mentions the specified volume, order from most to least recent. The output for each record appears in a table with six columns:

dumpID

The dump ID of the dump that includes the volume.

lvl

The depth in the dump hierarchy of the dump level at which the volume was dumped. A value of `0` indicates a full dump. A value of `1` or greater indicates an incremental dump made at the specified depth 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, the value in the `lvl` column is also `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 read-only volume, the time at which it was cloned from its read / write source. For a read / write volume, the same as the value in the `creation date` field.

tape name

The name of the tape containing the dump: either the permanent tape name, or an AFS 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; *tape_index* is the ordinal of the tape in the dump set. Either type of name can be followed by a dump ID in parentheses; if it appears, it is the dump ID of the initial dump in the dump set to which this appended dump belongs.

Examples

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

```
% backup volinfo -volume user.smith.backup
DumpID    lvl parentID  creation date    clone date       tape name
924600000 1   924427600 04/20/1999 05:20 04/20/1999 05:01 user_incr_2 (924514392)
924514392 1   924427600 04/19/1999 05:33 04/19/1999 05:08 user_incr_2
924427600 0           0 04/18/1999 05:26 04/18/1999 04:58 user_full_6
    .     .      .          .       .       .      .          .
    .     .      .          .       .       .      .          .
```

Privilege Required

None

See Also

"backup(8)" on page 531, "backup_dumpinfo(8)" on page 581, "backup_volrestore(8)" on page 647

Copyright

backup volrestore

Name

`backup volrestore` — Restores one or more volumes

Synopsis

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] [-localauth] [-cell <*cell name*>] [-help]

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] [-l] [-c <*cell name*>] [-h]

Description

The backup volrestore command restores the contents of one or more volumes to the site indicated by the -server and -partition arguments. Use the command either to overwrite the contents of existing volumes with the restored data or to create new volumes while retaining the existing ones. The specified site does not have to be the current site for the volumes.

(If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file associated with the specified port offset, then the backup volrestore command restores data from the backup data file listed for that port offset in the Tape Coordinator's */usr/afs/backup/tapeconfig* file, rather than from tape. For the sake of clarity, the following text refers to tapes only, but the Backup System handles backup data files in much the same way.)

The command's arguments can be combined as indicated:

- To preserve a volume's current contents and also create a new volume to house the restored version, use the -extension argument. The Backup System creates the new volume on the server and partition named by the -server and -partition arguments, assigns it the same name as the current volume with the addition of the specified extension, and creates a new Volume Location Database (VLDB) entry for it. Creating a new volume enables the administrator to compare the two versions.
- To overwrite a volume's existing contents with the restored version, omit the -extension argument, and specify the site as indicated:
  - To retain the current site, specify it with the -server and -partition arguments.
  - To move the volume to a different site while overwriting it, specify the new site with the -server argument, -partition argument, or both. The Backup System creates a new volume at that site, removes the existing volume, and updates the site information in the volume's VLDB entry. The backup version of the volume is not removed automatically from the original site, if it exists. Use the vos remove command to remove it and the vos backup command to create a backup version at the new site.

- To restore a volume that no longer exists in the file system, specify its name with the -volume argument and use the -server and -partition arguments to place it at the desired site. The Backup System creates a new volume and new VLDB entry.

In each case, the command sets each volume's creation date to the date and time at which it restores it. The creation date appears in the `Creation` field in the output from the vos examine and vos listvol commands.

If restoring all of the volumes that resided on a single partition, it is usually more efficient to use the backup diskrestore command. If restoring multiple volumes to many different sites, it can be more efficient to use the backup volsetrestore command.

By default, the backup volrestore command restores the most recent full dump and all subsequent incremental dumps for each volume, bringing the restored volumes to the most current possible state. To restore the volumes to their state at some time in the past, use the -date argument. The Backup System restores the most recent full dump and each subsequent incremental dump for which the *clone date* of the volume included in the dump is before the indicated date and time (the clone date timestamp appears in the `clone date` field of the output from the backup volinfo command). For backup and read-only volumes, the clone date represents the time at which the volume was copied from its read / write source; for read / write volumes, it represents the time at which the volume was locked for inclusion in the dump. The resemblance of a restored volume to its actual state at the indicated time depends on the amount of time that elapsed between the volume's clone date in the last eligible dump and the specified time.

If the -volume argument specifies the base (read / write) form of the volume name, the Backup System searches the Backup Database for the newest dump set that includes a dump of either the read / write or the backup version of the volume. It restores the dumps of that version of the volume, starting with the most recent full dump. If, in contrast, the volume name explicitly includes the `.backup` or `.readonly` extension, the Backup System restores dumps of the corresponding volume version only.

To generate a list of the tapes the Backup System needs to perform the restore operation, without actually performing it, combine the -n flag with the options to be used on the actual command.

If all of the full and incremental dumps of all relevant volumes were not written to a type of tape that a single Tape Coordinator can read, use the -portoffset argument to list multiple port offset numbers in the order in which the tapes are needed (first list the port offset for the full dump, second the port offset for the level 1 incremental dump, and so on). If restoring multiple volumes, the same ordered list of port offsets must apply to all of them. If not, either issue this command separately for each volume, or use the vos volsetrestore command after defining groups of volumes that were dumped to compatible tape types. For further discussion, see the *IBM AFS Administration Guide*.

The Tape Coordinator's default response to this command is to access the first tape it needs by invoking the MOUNT instruction in the local */usr/afs/backup/CFG_device_name* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *CFG_device_name* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, or is the wrong tape, the Tape Coordinator invokes the `MOUNT`

instruction or prompts the operator. It also invokes the `MOUNT` instruction or prompts for any additional tapes needed to complete the restore operation; the backup operator must arrange to provide them.

Options

-server <*destination machine*>

Names the file server machine on which to restore each volume. If this argument and the -partition argument indicate a site other than the current site for each volume, and the -extension argument is not also provided, the Backup System removes the existing volumes from their current sites, places the restored contents at the specified site, and changes the site information in the volume's VLDB entry.

-partition <*destination partition*>

Names the partition to which to restore each volume. If this argument and the -server argument indicate a site other than the current site for each volume, and the -extension argument is not also provided, the Backup System removes the existing volumes from their current sites, places the restored contents at the specified site, and changes the site information in the volume's VLDB entry.

-volume <*volume to restore*>+

Names one or more volumes to restore, using the volume name as listed in the Backup Database. Provide the base (read / write) name of each volume to have the Backup System search the Backup Database for the newest dump set that includes a dump of either the read / write or the backup version of the volume; it restores the dumps of that version of the volume, starting with the most recent full dump. If, in contrast, a volume name explicitly includes the `.backup` or `.readonly` extension, the Backup System restores dumps of the corresponding volume version only.

-extension <*new volume name extension*>

Creates a new volume to house the restored data, with a name derived by appending the specified string to each volume named by the -volume argument. The Backup System creates a new VLDB entry for the volume. Any string other than `.readonly` or `.backup` is acceptable, but the combination of the existing volume name and extension cannot exceed 22 characters in length. To use a period to separate the extension from the name, specify it as the first character of the string (as in `.rst`, for example).

-date <*date from which to restore*>+

Specifies a date and optionally time; the restored volume includes data from dumps performed before the date only. Provide a value in the format *mm/dd/yyyy* [*hh*:*MM*], where the required *mm/dd/yyyy* portion indicates the month (*mm*), day (*dd*), and year (*yyyy*), and the optional *hh:MM* portion indicates the hour and minutes in 24-hour format (for example, the value `14:36` represents 2:36 p.m.). If omitted, the time defaults to 59 seconds after midnight (00:00:59 hours).

Valid values for the year range from `1970` to `2037`; higher values are not valid because the latest possible date in the standard UNIX representation is in February 2038. The command interpreter automatically reduces any later date to the maximum value.

If this argument is omitted, the Backup System restores all possible dumps including the most recently created.

-portoffset <*TC port offest*>+

Specifies one or more port offset numbers (up to a maximum of 128), each corresponding to a Tape Coordinator to use in the operation. If there is more than one value, the Backup System uses the first one when restoring the full dump of each volume, the second one when restoring the level 1 incremental dump of each volume, and so on. It uses the final value in the list when restoring dumps at the corresponding depth in the dump hierarchy and all dumps at lower levels.

Provide this argument unless the default value of 0 (zero) is appropriate for all dumps. If `0` is just one of the values in the list, provide it explicitly in the appropriate order.

-n

Displays the list of tapes that contain the dumps required by the restore operation, without actually performing the operation.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If the issuer includes the -n flag with the command, the following string appears at the head of the list of the tapes necessary to complete the restore operation.

```
Tapes needed:
```

Examples

The following command restores the volume user.pat to partition */vicepa* on machine `fs5.abc.com`:

```
% backup volrestore -server fs5.abc.com -partition a -volume user.pat
```

The following command restores the volumes `user.smith` and `user.terry` to partition */vicepb* on machine `fs4.abc.com`, adding a `.rst` extension to each volume name and preserving the existing `user.smith` and `user.terry` volumes. Only dumps created before 5:00 p.m. on 31 January 1998 are restored. (The command is shown here on multiple lines only for legibility reasons.)

```
% backup volrestore -server fs4.abc.com -partition b  \
                    -volume user.smith user.terry  \
                    -extension .rst -date 1/31/1998 17:00
```

The following command restores the volume user.pat to partition */vicepb* on machine `fs4.abc.com`. The Tape Coordinator with port offset 1 handles the tape containing the full dump; the Tape Coordinator with port offset 0 handles all tapes containing incremental dumps. (The command is shown here on two lines only for legibility reasons.)

```
% backup volrestore -server fs5.abc.com -partition a  \
                    -volume user.pat -portoffset 1 0
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server or Volume Location (VL) Server is running, and on every file server machine that houses an affected volume. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_dump(8)" on page 571, "backup_diskrestore(8)" on page 565, "backup_volsetrestore(8)" on page 653, "butc(8)" on page 767, "vos_backup(1)" on page 299, "vos_remove(1)" on page 359

Copyright

backup volsetrestore

Name

`backup volsetrestore` — Restores all volumes in a volume set

Synopsis

backup volsetrestore [-name <*volume set name*>] [-file <*file name*>] [-portoffset <*TC port offset*>+] [-extension <*new volume name extension*>] [-n] [-localauth] [-cell <*cell name*>] [-help]

backup vols [-na <*volume set name*>] [-f <*file name*>] [-p <*TC port offset*>+] [-e <*new volume name extension*>] [-n] [-l] [-c <*cell name*>] [-h]

Description

The backup volsetrestore command restores the complete contents of a group of read / write volumes to the file system, by restoring data from the last full dump and all subsequent incremental dumps of each volume. It is most useful for recovering from loss of data on multiple partitions, since it can restore each of a defined set of volumes to a different site.

(If the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file associated with the specified port offset, then the backup volsetrestore command restores data from the backup data file listed for that port offset in the Tape Coordinator's */usr/afs/backup/tapeconfig* file, instead of from tape. For the sake of clarity, the following text refers to tapes only, but the Backup System handles backup data files in much the same way.)

If restoring one or more volumes to a single site only, it is usually more efficient to use the backup volrestore command. If restoring all volumes that resided on a single partition, it is usually more efficient to use the backup diskrestore command.

Indicate the volumes to restore by providing either the -name argument or the -file argument:

- The -name argument names a volume set. The Backup System restores all volumes listed in the Volume Location Database (VLDB) that match the server, partition, and volume name criteria defined in the volume set's volume entries, and for which dumps are available. It restores the volumes to their current site (machine and partition), and by default overwrites the existing volume contents.

  It is not required that the volume set was previously used to back up volumes (was used as the -volumeset option to the backup dump command). It can be defined especially to match the volumes that need to be restored with this command, and that is usually the better choice. Indeed, a *temporary* volume set, created by including the -temporary flag to the backup addvolset command, can be especially useful in this context. A temporary volume set is not added to the Backup Database and exists only during the current interactive backup session, which is suitable if the volume set is needed only to complete the single restore operation initialized by this command.

  The reason that a specially defined volume set is probably better is that volume sets previously defined for use in dump operations usually match the backup version of

volumes, whereas for a restore operation it is best to define volume entries that match the base (read / write) name. In that case, the Backup System searches the Backup Database for the newest dump set that includes either the read / write or the backup version of the volume. If, in contrast, a volume entry explicitly matches the volume's backup or read-only version, the Backup System restores dumps of that volume version only.

- The -file argument names a file that lists specific volumes and the site to which to restore each. The volume name must match the name used in Backup Database dump records rather than in the VLDB, if they differ, because the Backup System does not look up volumes in the VLDB. The specified site can be different than the volume's current one; in that case, the Backup System removes the current version of the volume and updates the volume's location information in the VLDB.

If all of the full and incremental dumps of all relevant volumes were not written to a type of tape that a single Tape Coordinator can read, use the -portoffset argument to list multiple port offset numbers in the order in which the tapes are needed (first list the port offset for the full dump, second the port offset for the level 1 incremental dump, and so on). This implies that the full dumps of all relevant volumes must have been written to a type of tape that the first Tape Coordinator can read, the level 1 incremental dumps to a type of tape the second Tape Coordinator can read, and so on. If dumps are on multiple incompatible tape types, use the backup volrestore command to restore individual volumes, or use this command after defining new volume sets that group together volumes that were dumped to compatible tape types. For further discussion, see the *IBM AFS Administration Guide*.

By default, the Backup System overwrites the contents of an existing volume with the restored data. To create a new volume to house the restored version instead, use the -extension argument. The Backup System derives the new volume's name by adding the specified extension to the read / write base name, and creates a new VLDB entry. The command does not affect the existing volume in any way. However, if a volume with the specified extension also already exists, the command overwrites it.

The -n flag produces a list of the volumes to be restored if the -n flag were not included, without actually restoring any volumes. See "OUTPUT" on page for a detailed description of the output, and suggestions on how to combine it most effectively with the -file and -name arguments.

The execution time for a backup volsetrestore command depends on the number of volumes to be restored and the amount of data in them, but it can take hours to restore a large number of volumes. One way to reduce the time is to run multiple instances of the command simultaneously, either using the -name argument to specify disjoint volume sets for each command, or the -file argument to name files that list different volumes. This is possible if there are multiple available Tape Coordinators that can read the required tapes. Depending on how the volumes to be restored were dumped to tape, specifying disjoint volume sets can also reduce the number of tape changes required.

The Tape Coordinator's default response to this command is to access the first tape it needs by invoking the `MOUNT` instruction in the local */usr/afs/backup/CFG_device_name* file, or by prompting the backup operator to insert the tape if there is no `MOUNT` instruction. However, if the `AUTOQUERY NO` instruction appears in the *CFG_device_name* file, or if the issuer of the butc command included the -noautoquery flag, the Tape Coordinator instead expects the tape to be in the device already. If it is not, or is the wrong tape, the Tape Coordinator invokes the `MOUNT`

instruction or prompts the operator. It also invokes the `MOUNT` instruction or prompts for any additional tapes needed to complete the restore operation; the backup operator must arrange to provide them.

Options

-name <*volume set name*>

Names a volume set to restore. The Backup System restores all of the volumes listed in the VLDB that match the volume set's volume entries. Provide this argument or the -file argument, but not both.

-file <*file name*>

Specifies the full pathname of a file that lists one or more volumes and the site (file server machine and partition) to which to restore each. Use either this argument or the -name argument, but not both.

Each volume's entry must appear on its own (unbroken) line in the file, and have the following format:

```
&lt;machine&gt; &lt;partition&gt; &lt;volume&gt; [&lt;comments&gt; ...]
```

where

<machine>

Names the file server machine to which to restore the volume.

<partition>

Names the partition to which to restore the volume.

<volume>

Names the volume to restore. It is generally best to specify the base (read / write) name of each volume. In this case, the Backup System searches the Backup Database for the newest dump set that includes a dump of either the read / write or the backup version of the volume. It restores the dumps of that version of the volume, starting with the most recent full dump. If, in contrast, the name explicitly includes the `.backup` or `.readonly` extension, the Backup System restores dumps of that volume version only.

<comments> ...

Is any other text. The Backup System ignores any text on each line that appears after the volume name, so this field can be used for notes helpful to the backup operator or other administrator.

Do not use wildcards (for example, `.*`) in the <machine>, <partition>, or <volume> fields. It is acceptable for multiple lines in the file to name the same volume, but the Backup System processes only the first of them.

-extension <*new volume name extension*>

Creates a new volume for each volume specified by the -name or -file argument, to house the restored data from that volume. The Backup System derives the new volume's name by appending the specified string to the read/write base name, and creates a new VLDB volume entry. It preserves the contents of each existing volume. Any string other than `.readonly` or `.backup` is acceptable, but the combination of the base name and extension cannot exceed 22 characters in length. To use a period to separate the extension from the name, specify it as the first character of the string (as in `.rst`, for example).

-portoffset <*TC port offset*>+

Specifies one or more port offset numbers (up to a maximum of 128), each corresponding to a Tape Coordinator to use in the operation. If there is more than one value, the Backup System uses the first one when restoring the full dump of each volume, the second one when restoring the level 1 incremental dump of each volume, and so on. It uses the final value in the list when restoring dumps at the corresponding depth in the dump hierarchy and all dumps at lower levels.

Provide this argument unless the default value of 0 (zero) is appropriate for all dumps. If `0` is just one of the values in the list, provide it explicitly in the appropriate order.

-n

Displays a list of the volumes to be restored if the flag were not included, without actually restoring them. "OUTPUT" on page details the format of the output. When combined with the -name argument, its output is easily edited for use as input to the -file argument on a subsequent backup volsetrestore command.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The backup command interpreter presents it to the Backup Server, Volume Server and VL Server during mutual authentication. Do not combine this flag with the -cell argument. For more details, see "backup(8)" on page 531.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "backup(8)" on page 531.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If the -n flag is not provided, the command displays a unique task ID number for the operation, in two places:

- In the shell window, directly following the command line.
- In the Tape Coordinator window, if the butc process was started at debug level 1.

The task ID number is not the same as the job ID number displayed by the backup jobs command when the backup volsetrestore command is issued in interactive mode. The Backup System does not assign either type of ID number until the restoration process actually begins.

When the -n flag is included, no task ID or job ID numbers are reported because none are assigned. Instead, the output begins with a count of the number of volumes to be restored, followed by a line for each dump of a volume. For each volume, the line representing the most recent full dump appears first, and lines for any subsequent incremental dumps follow, ordered by dump level. The lines for a given volume do not necessarily appear all together, however.

The format of each line is as follows (the output is shown here on two lines only for legibility reasons):

```
&lt;machine&gt; &lt;partition&gt; &lt;volume_dumped&gt; # as &lt;volume_restored&gt;; \
    &lt;tape_name&gt; (&lt;tape_ID&gt;); pos &lt;position_number&gt;; &lt;date&gt;
```

where

<machine>

Names the file server machine that currently houses the volume, as listed in the VLDB.

<partition>

Names the partition that currently houses the volume, as listed in the VLDB.

<volume_dumped>

Specifies the version (read / write or backup) of the volume that was dumped, as listed in the Backup Database.

<volume_restored>

Specifies the name under which to restore the volume. The Backup System only restores data to read / write volumes. If the -extension argument is included, then the specified extension appears on the name in this field (for example, `user.pat.rst`).

<tape_name>

Names the tape containing the dump of the volume, from the Backup Database. If the tape has a permanent name, it appears here; otherwise, it is the AFS tape name.

<tape_ID>

The tape ID of the tape containing the dump of the volume, from the Backup Database.

<position_number>

Specifies the dump's position on the tape (for example, `31` indicates that 30 volume dumps precede the current one on the tape). If the dump was written to a backup data file, this number is the ordinal of the 16 KB-offset at which the volume's data begins.

<date>

The date and time when the volume was dumped.

One way to generate a file for use as input to the -file argument is to combine the -name and -n options, directing the output to a file. The *IBM AFS Administration Guide* section on using the Backup System to restore data explains how to edit the file as necessary before using it as input to the -file argument.

The output of this command includes only volumes for which the Backup Database includes at least one dump record. The command interpreter generates a message on the standard error stream about volumes that do not have dump records but either are listed in the file named by the -file argument, or appear in the VLDB as a match to a volume entry in the volume set named by the -name argument.

Examples

The following command restores all volumes included in entries in the volume set named `data.restore`, which was created expressly to restore data to a pair of file server machines on which all data was corrupted due to a software error. All volumes are restored to the sites recorded in their entries in the VLDB.

```
% backup volsetrestore -name data.restore
Starting restore
backup: task ID of restore operation: 112
backup: Finished doing restore
```

The following command restores all volumes that have entries in the file named */tmp/restore*:

```
% backup volsetrestore -file /tmp/restore
Starting restore
backup: task ID of restore operation: 113
backup: Finished doing restore
```

The */tmp/restore* file has the following contents:

```
fs1.abc.com b user.pat
fs1.abc.com b user.terry
fs1.abc.com b user.smith
fs2.abc.com c user.jones
       .         .     .
       .         .     .
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server or Volume Location (VL) Server is running, and on every file server machine that houses an affected volume. If the -localauth flag is included, the issuer must instead be logged on to a server machine as the local superuser `root`.

See Also

"butc(5)" on page 453, "backup(8)" on page 531, "backup_addvolentry(8)" on page 541, "backup_addvolset(8)" on page 545, "backup_diskrestore(8)" on page 565, "backup_dump(8)" on page 571, "backup_volrestore(8)" on page 647, "butc(8)" on page 767

Copyright

bos

Name

bos — Introduction to the bos command suite

Description

The commands in the bos command suite are the administrative interface to the Basic OverSeer (BOS) Server, which runs on every file server machine to monitor the other server processes on it. If a process fails, the BOS Server can restart it automatically, taking into account interdependencies between it and other processes. The BOS Server frees system administrators from constantly monitoring the status of server machines and processes.

There are several categories of commands in the bos command suite:

- Commands to administer server process binary files: bos getdate, bos install, bos prune, and bos uninstall.
- Commands to maintain system configuration files: bos addhost, bos addkey, bos adduser, bos listhosts, bos listkeys, bos listusers, bos removehost, bos removekey, bos removeuser, and bos setcellname.
- Commands to start and stop processes: bos create, bos delete, bos restart, bos shutdown, bos start, bos startup, and bos stop.
- Commands to set and verify server process and server machine status: bos getlog, bos getrestart, bos setauth, bos setrestart, and bos status.
- A command to restore file system consistency: bos salvage.
- Commands to obtain help: bos apropos and bos help.

The BOS Server and the bos commands use and maintain the following configuration and log files:

- The */usr/afs/etc/CellServDB* file lists the local cell's database server machines. These machines run the Authentication, Backup, Protection and Volume Location (VL) Server processes, which maintain databases of administrative information. The database server processes consult the file to learn about their peers, whereas the other server processes consult it to learn where to access database information as needed. To administer the *CellServDB* file, use the following commands: bos addhost, bos listhosts, bos removehost, and bos setcellname.
- The */usr/afs/etc/KeyFile* file lists the server encryption keys that the server processes use to decrypt tickets presented by client processes and one another. To administer the *KeyFile* file, use the following commands: bos addkey, bos listkeys, and bos removekey.
- The */usr/afs/etc/ThisCell* file defines the cell to which the server machine belongs for the purposes of server-to-server communication. Administer it with the bos setcellname command. There is also a */usr/vice/etc/ThisCell* file that defines the machine's cell membership with respect to the AFS command suites and Cache Manager access to AFS data.

- The */usr/afs/etc/UserList* file lists the user name of each administrator authorized to issue privileged bos and vos commands. To administer the *UserList* file, use the following commands: bos adduser, bos listusers, and bos removeuser.
- The */usr/afs/local/BosConfig* file defines which AFS server processes run on the server machine, and whether the BOS Server restarts them automatically if they fail. It also defines when all processes restart automatically (by default once per week), and when the BOS Server restarts processes that have new binary files (by default once per day). To administer the *BosConfig* file, use the following commands: bos create, bos delete, bos getrestart, bos setrestart, bos start, and bos stop.
- The */usr/afs/log/BosLog* file records important operations the BOS Server performs and error conditions it encounters.

For more details, see the reference page for each file.

Options

The following arguments and flags are available on many commands in the bos suite. The reference page for each command also lists them, but they are described here in greater detail.

-cell <*cell name*>

Names the cell in which to run the command. It is acceptable to abbreviate the cell name to the shortest form that distinguishes it from the other entries in the */usr/vice/etc/CellServDB* file on the local machine. If the -cell argument is omitted, the command interpreter determines the name of the local cell by reading the following in order:

- The value of the AFSCELL environment variable.
- The local */usr/vice/etc/ThisCell* file.

Do not combine the -cell and -localauth options. A command on which the -localauth flag is included always runs in the local cell (as defined in the server machine's local */usr/afs/etc/ThisCell* file), whereas a command on which the -cell argument is included runs in the specified foreign cell.

-help

Prints a command's online help message on the standard output stream. Do not combine this flag with any of the command's other options; when it is provided, the command interpreter ignores all other options, and only prints the help message.

-localauth

Constructs a server ticket using the server encryption key with the highest key version number in the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket, which never expires, to the BOS Server during mutual authentication.

Use this flag only when issuing a command on a server machine; client machines do not usually have a */usr/afs/etc/KeyFile* file. The issuer of a command that includes this flag must be logged on to the server machine as the local superuser `root`. The flag is useful for commands invoked by an unattended application program, such as a process

controlled by the UNIX cron utility or by a cron entry in the machine's */usr/afs/local/BosConfig* file. It is also useful if an administrator is unable to authenticate to AFS but is logged in as the local superuser `root`.

Do not combine the -cell and -localauth options. A command on which the -localauth flag is included always runs in the local cell (as defined in the server machine's local */usr/afs/etc/ThisCell* file), whereas a command on which the -cell argument is included runs in the specified foreign cell. Also, do not combine the -localauth and -noauth flags.

-noauth

Establishes an unauthenticated connection to the BOS Server, in which the BOS Server treats the issuer as the unprivileged user `anonymous`. It is useful only when authorization checking is disabled on the server machine (during the installation of a file server machine or when the bos setauth command has been used during other unusual circumstances). In normal circumstances, the BOS Server allows only privileged users to issue commands that change the status of a server or configuration file, and refuses to perform such an action even if the -noauth flag is provided. Do not combine the -noauth and -localauth flags.

-server <*machine name*>

Indicates the AFS server machine on which to run the command. Identify the machine by its IP address in dotted decimal format, its fully-qualified host name (for example, `fs1.abc.com`), or by an abbreviated form of its host name that distinguishes it from other machines. Successful use of an abbreviated form depends on the availability of a name service (such as the Domain Name Service or a local host table) at the time the command is issued.

For the commands that alter the administrative files shared by all server machines in the cell (the bos addhost, bos addkey, bos adduser, bos removehost, bos removekey, and bos removeuser commands), the appropriate machine depends on whether the cell uses the United States or international version of AFS:

- If the cell (as recommended) uses the Update Server to distribute the contents of the */usr/afs/etc* directory, provide the name of the system control machine. After issuing the command, allow up to five minutes for the Update Server to distribute the changed file to the other AFS server machines in the cell. If the specified machine is not the system control machine but is running an upclient process that refers to the system control machine, then the change will be overwritten when the process next brings over the relevant file from the system control machine.
- Otherwise, repeatedly issue the command, naming each of the cell's server machines in turn. To avoid possible inconsistency problems, finish issuing the commands within a fairly short time.

Privilege Required

To issue any bos command that changes a configuration file or alters process status, the issuer

must be listed in the */usr/afs/etc/UserList* file on the server machine named by the -server argument. Alternatively, if the -localauth flag is included the issuer must be logged on as the local superuser `root`.

To issue a bos command that only displays information (other than the bos listkeys command), no privilege is required.

See Also

"BosConfig(5)" on page 409, "CellServDB(5)" on page 415, "KeyFile(5)" on page 423, "ThisCell(5)" on page 435, "UserList(5)" on page 437, "bos_addhost(8)" on page 665, "bos_addkey(8)" on page 669, "bos_adduser(8)" on page 673, "bos_apropos(8)" on page 675, "bos_create(8)" on page 677, "bos_delete(8)" on page 685, "bos_exec(8)" on page 687, "bos_getdate(8)" on page 689, "bos_getlog(8)" on page 691, "bos_getrestart(8)" on page 695, "bos_help(8)" on page 699, "bos_install(8)" on page 701, "bos_listhosts(8)" on page 705, "bos_listkeys(8)" on page 707, "bos_listusers(8)" on page 711, "bos_prune(8)" on page 713, "bos_removehost(8)" on page 717, "bos_removekey(8)" on page 719, "bos_removeuser(8)" on page 721, "bos_restart(8)" on page 723, "bos_salvage(8)" on page 727, "bos_setauth(8)" on page 733, "bos_setcellname(8)" on page 735, "bos_setrestart(8)" on page 739, "bos_shutdown(8)" on page 743, "bos_start(8)" on page 747, "bos_startup(8)" on page 749, "bos_status(8)" on page 751, "bos_stop(8)" on page 755, "bos_uninstall(8)" on page 757

Copyright

bos addhost

Name

`bos addhost` — Adds a database server machine to the CellServDB file

Synopsis

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

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

Description

The bos addhost command adds an entry for each database server machine specified with the -host argument to the */usr/afs/etc/CellServDB* file on the machine named by the -server argument.

Cautions

After executing this command (and waiting for the Update Server to propagate the changes, if it is used), restart the database server processes on all database server machines to force election of a quorum that includes the new set of machines listed in the */usr/afs/etc/CellServDB* file. The *IBM AFS Quick Beginnings* explains in more detail how to add and remove database server machines.

It is best to maintain a one-to-one mapping between hostnames and IP addresses on a multihomed database server machine (this is actually the conventional configuration for any AFS machine). The BOS Server uses the gethostbyname() routine to obtain the IP address associated with the hostname specified by the -host argument. If there is more than one address, the BOS Server records in the *CellServDB* entry the one that appears first in the list of addresses returned by the routine. The routine possibly returns addresses in a different order on different machines, which can create inconsistency.

Options

-server <*machine name*>

Identifies the server machine on which to change the */usr/afs/etc/CellServDB* file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

In cells that use the Update Server to distribute the contents of the */usr/afs/etc* directory, it is conventional to specify only the system control machine as a value for the -server argument. Otherwise, repeat the command for each file server machine. For further discussion, see "bos(8)" on page 661.

-host <*host name*>+

Specifies the fully-qualified host name (such as `db1.abc.com`) of each database server machine to register in the *`CellServDB`* file.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command adds the database server machines `db2.abc.com` and `db3.abc.com` to the */usr/afs/etc/CellServDB* file on the machine `fs1.abc.com` (the system control machine).

```
% bos addhost -server fs1.abc.com -host db2.abc.com db3.abc.com
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"CellServDB(5)" on page 415, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_listhosts(8)" on page 705, "bos_removehost(8)" on page 717

IBM AFS Quick Beginnings

Copyright

bos addkey

Name

`bos addkey` — Adds a new server encryption key to the KeyFile file

Synopsis

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

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

Description

The bos addkey command constructs a server encryption key from the text string provided, assigns it the key version number specified with the -kvno argument, and adds it to the */usr/afs/etc/KeyFile* file on the machine specified with the -server argument. Be sure to use the kas setpassword or kas setkey command to add the same key to the `afs` entry in the Authentication Database.

Do not use the -key argument, which echoes the password string visibly on the screen. If the argument is omitted, the BOS Server prompts for the string and does not echo it visibly:

```
Input key:
Retype input key:
```

The BOS Server prohibits reuse of any key version number already listed in the */usr/afs/etc/KeyFile* file. This ensures that users who still have tickets sealed with the current key are not prevented from communicating with a server process because the current key is overwritten with a new key. Use the bos listkeys command to display the key version numbers in the */usr/afs/etc/KeyFile* file.

Options

-server <*machine name*>

Indicates the server machine on which to change the */usr/afs/etc/KeyFile* file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

In cells that use the Update Server to distribute the contents of the */usr/afs/etc* directory, it is conventional to specify only the system control machine as a value for the -server argument. Otherwise, repeat the command for each file server machine. For further discussion, see "bos(8)" on page 661.

-key <*key*>

Specifies 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. Omit this argument to have the BOS Server prompt for the string instead.

-kvno <*key version number*>

Defines the new key's key version number. It must be an integer in the range from `0` (zero) through `255`. For the sake of simplicity, use the number one higher than the current highest key version number; use the bos listkeys command to display key version numbers.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If the strings typed at the `Input key` and `Retype input key` prompts do not match, the following message appears, and the command exits without adding a new key:

```
Input key mismatch
```

Examples

The following command adds a new server encryption key with key version number 14 to the KeyFile file kept on the machine `fs1.abc.com` (the system control machine). The issuer omits the -key argument, as recommended, and provides the password at the prompts.

```
% bos addkey -server fs1.abc.com -kvno 14
Input key:
Retype input key:
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_listkeys(8)" on page 707, "bos_removekey(8)" on page 719

Copyright

bos adduser

Name

`bos adduser` — Adds a privileged user to the UserList file

Synopsis

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

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

Description

The bos adduser command adds each user name specified with the -user argument to the */usr/afs/etc/UserList* file on the machine named by the -server argument. It is the issuer's responsibility to verify that an entry for the user exists in the Authentication and Protection Databases.

Options

-server <*machine name*>

Indicates the server machine on which to change the */usr/afs/etc/UserList* file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

In cells that use the Update Server to distribute the contents of the */usr/afs/etc* directory, it is conventional to specify only the system control machine as a value for the -server argument. Otherwise, repeat the command for each file server machine. For further discussion, see "bos(8)" on page 661.

-user <*user names*>+

Specifies each user name to insert into the */usr/afs/etc/UserList* file.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication.

Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command adds the user names `pat` and `smith` to the */usr/afs/etc/UserList* file on the machine `fs1.abc.com` (the system control machine).

```
% bos adduser -server fs1.abc.com -user pat smith
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_listusers(8)" on page 711, "bos_removeuser(8)" on page 721

Copyright

bos apropos

Name

`bos apropos` — Displays each help entry containing a keyword string

Synopsis

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

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

Description

The bos apropos command displays the first line of the online help entry for any bos command that has in its name or short description the string specified by the -topic argument.

To display the syntax for a command, use the bos help command.

Options

-topic <*help string*>

Specifies the keyword string to match, in lowercase letters only. If the string is more than a single word, surround it with double quotes ("") or other delimiters.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of a command's online help entry names it and briefly describes its function. This command displays the first line for any bos command where the string specified with the -topic argument is part of the command name or first line.

Examples

The following command lists all bos commands that include the word `restart` in their names or short descriptions:

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

Privilege Required

None

See Also

"bos(8)" on page 661, "bos_help(8)" on page 699

Copyright

bos create

Name

`bos create` — Defines a new process in the BosConfig file and starts it

Synopsis

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

Description

The bos create command creates a server process entry in the */usr/afs/local/BosConfig* file on the server machine named by the -server argument, sets the process's status to `Run` in the *BosConfig* file and in memory, and starts the process.

A server process's entry in the *BosConfig* file defines its name, its type, the command that initializes it, and optionally, the name of a notifier program that runs when the process terminates.

Options

-server <*machine name*>

Indicates the server machine on which to define and start the new process. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-instance <*server process name*>

Names the process to define and start. Any name is acceptable, but for the sake of simplicity it is best to use the last element of the process's binary file pathname, and to use the same name on every server machine. The conventional names, as used in all AFS documentation, are:

buserver

The Backup Server process.

fs

The process that combines the File Server, Volume Server, and Salvager processes (fileserver, volserver, and salvager).

kaserver

The Authentication Server process.

ptserver

The Protection Server process.

runntp

The controller process for the Network Time Protocol Daemon.

upclientbin

The client portion of the Update Server process that retrieves binary files from the */usr/afs/bin* directory of the binary distribution machine for this machine's CPU/operating system type. (The name of the binary is upclient, but the `bin` suffix distinguishes this process from `upclientetc`.)

upclientetc

The client portion of the Update Server process that retrieves configuration files from the */usr/afs/etc* directory of the system control machine. (The name of the binary is upclient, but the `etc` suffix distinguishes this process from `upclientbin`.)

upserver

The server portion of the Update Server process.

vlserver

The Volume Location (VL) Server process.

-type <*server type*>

Specifies the process's type. The acceptable values are:

cron

Use this value for cron-type processes that the BOS Server starts only at a defined daily or weekly time, rather than whenever it detects that the process has terminated. AFS does not define any such processes by default, but makes this value available for administrator use. Define the time for command execution as part of the -cmd argument to the bos create command.

fs

Use this value only for the fs process, which combines the File Server, Volume Server and Salvager processes. If one of the component processes terminates, the BOS Server shuts down and restarts the processes in the appropriate order.

simple

Use this value for all processes listed as acceptable values to the -instance argument, except for the fs process. There are no interdependencies between simple processes, so the BOS Server can stop and start them independently as necessary.

-cmd <*command lines*>+

Specifies each command the BOS Server runs to start the process. Specify no more than six commands (which can include the command's options, in which case the entire string is surrounded by double quotes); any additional commands are ignored.

For a simple process, provide the complete pathname of the process's binary file on the local disk (for example, `/usr/afs/bin/ptserver` for the Protection Server). If including any of the initialization command's options, surround the entire command in double quotes (""). The upclient process has a required argument, and the commands for all other processes take optional arguments.

For the fs process, provide the complete pathname of the local disk binary file for each of the component processes: fileserver, volserver, and salvager, in that order. The standard binary directory is `/usr/afs/bin`. If including any of an initialization command's options, surround the entire command in double quotes ("").

For a cron process, provide two parameters:

- The complete local disk pathname of either an executable file or a command from one of the AFS suites (complete with all of the necessary arguments). Surround this parameter with double quotes ("") if it contains spaces.
- A specification of when the BOS Server executes the file or command indicated by the first parameter. There are three acceptable values:
  - The string `now`, which directs the BOS Server to execute the file or command immediately and only once. It is usually simpler to issue the command directly or issue the bos exec command.
  - A time of day. The BOS Server executes the file or command daily at the indicated time. Separate the hours and minutes with a colon (*hh:MM*), and use either 24-hour format, or a value in the range from `1:00` through `12:59` with the addition of `am` or `pm`. For example, both `14:30` and `"2:30 pm"` indicate 2:30 in the afternoon. Surround this parameter with double quotes ("") if it contains a space.
  - A day of the week and time of day, separated by a space and surrounded with double quotes (""). The BOS Server executes the file or command weekly at the indicated day and time. For the day, provide either the whole name or the first three letters, all in lowercase letters (`sunday` or `sun`, `thursday` or `thu`, and so on). For the time, use the same format as when specifying the time alone.

-notifier <*notifier program*>

Specifies the complete pathname on the local disk of a program that the BOS Server invokes when the process terminates. The AFS distribution does not include any notifier programs, but this argument is available for administrator use. See "NOTES" on page .

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command defines and starts the simple process kaserver on the machine fs3.abc.com:

```
% bos create -server fs3.abc.com -instance kaserver -type simple \
             -cmd /usr/afs/bin/kaserver
```

The following command defines and starts the simple process upclientbin on the machine fs4.abc.com. It references fs1.abc.com as the source for updates to binary files, checking for changes to the */usr/afs/bin* directory every 120 seconds.

```
% bos create -server fs4.abc.com -instance upclientbin -type simple \
             -cmd "/usr/afs/bin/upclient fs1.abc.com -clear -t 120 \
             /usr/afs/bin"
```

The following command creates the fs process fs on the machine fs4.abc.com. Type the command on a single line.

```
% bos create -server fs4.abc.com -instance fs -type fs \
             -cmd /usr/afs/bin/fileserver /usr/afs/bin/volserver \
             /usr/afs/bin/salvager
```

The following command creates a cron process called userbackup on the machine fs5.abc.com, so that the BOS Server issues the indicated vos backupsys command each day at 3:00 a.m. (the command creates a backup version of every volume in the file system whose name begins with user). Note that the issuer provides the complete pathname to the vos command, includes the -localauth flag on it, and types the entire bos create command on one line.

```
% bos create -server fs5.abc.com -instance userbackup -type cron  \
    -cmd "/usr/afs/bin/vos backupsys -prefix user -localauth" 03:00
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

Notes

If the -notifier argument is included when this command is used to define and start a process, the BOS Server invokes the indicated *notifier program* when the process exits. The intended use of a notifier program is to inform administrators when a process exits unexpectedly, but it can be used to perform any appropriate actions. The following paragraphs describe the bnode and bnode_proc structures in which the BOS Server records information about the exiting process.

The BOS Server constructs and sends on the standard output stream one bnode and one bnode_proc structure for each exiting process associated with the notifier program. It brackets each structure with appropriate `BEGIN` and `END` statements (`BEGIN bnode` and `END bnode`, `BEGIN bnode_proc` and `END bnode_proc`), which immediately follow the preceding newline character with no intervening spaces or other characters. If the notifier program does not need information from a structure, it can scan ahead in the input stream for the `END` statement.

In general, each field in a structure is a string of ASCII text terminated by the newline character. The format of the information within a structure possibly varies slightly depending on the type of process associated with the notifier program.

The C code for the bnode and bnode_proc structures follows. Note that the structures sent by the BOS Server do not necessarily include all of the fields described here, because some are used only for internal record keeping. The notifier process must robustly handle the absence of expected fields, as well as the presence of unexpected fields, on the standard input stream.

For proper performance, the notifier program must continue processing the input stream until it detects the end-of-file (EOF). The BOS Server closes the standard input file descriptor to the notifier process when it has completed delivery of the data, and it is the responsibility of the notifier process to terminate properly.

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-&gt;name);
   printf("rsTime: %ld\n", tp-&gt;rsTime);
   printf("rsCount: %ld\n", tp-&gt;rsCount);
   printf("procStartTime: %ld\n", tp-&gt;procStartTime);
   printf("procStarts: %ld\n", tp-&gt;procStarts);
   printf("lastAnyExit: %ld\n", tp-&gt;lastAnyExit);
   printf("lastErrorExit: %ld\n", tp-&gt;lastErrorExit);
   printf("errorCode: %ld\n", tp-&gt;errorCode);
   printf("errorSignal: %ld\n", tp-&gt;errorSignal);
   printf("lastErrorName: %s\n", tp-&gt;lastErrorName);
   printf("goal: %d\n", tp-&gt;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-&gt;comLine);
   printf("coreName: %s\n", tp-&gt;coreName);
   printf("pid: %ld\n", tp-&gt;pid);
   printf("lastExit: %ld\n", tp-&gt;lastExit);
   printf("lastSignal: %ld\n", tp-&gt;lastSignal);
```

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "buserver(8)" on page 763, "fileserver(8)" on page 771, "kaserver(8)" on page 859, "ptserver(8)" on page 875, "salvager(8)" on page 879, "upclient(8)" on page 885, "upserver(8)" on page 889, "vlserver(8)" on page 915, "volserver(8)" on page 925, "vos_backupsys(1)" on page 301

Copyright

bos delete

Name

`bos delete` — Deletes a server process from the BosConfig file

Synopsis

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

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

Description

The bos delete command removes the */usr/afs/local/BosConfig* entry for each process indicated by the -instance argument, on the server machine named by the -server argument.

Before issuing this command, issue the bos stop command to stop the process and set its status flag in the *BosConfig* file to `NotRun`. The bos delete command fails with an error message if a process's status flag is `Run`.

Options

-server <*machine name*>

Indicates the server machine on which to delete the server process entry from the */usr/afs/local/BosConfig* file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-instance <*server process name*>+

Names each process to delete. Use the name assigned with the -instance argument to the bos create command; process names appear in the output of the bos status command.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

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

```
% bos delete -server db3.abc.com \
    -instance buserver kaserver ptserver vlserver
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_status(8)" on page 751

Copyright

bos exec

Name

`bos exec` — Executes a command on a remote server machine

Synopsis

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

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

Description

The bos exec command executes the indicated command on the file server machine named by the -server argument. Its intended use is to reboot the machine, using the `/sbin/reboot` command or equivalent.

Options

-server <*machine name*>

Indicates the server machine on which to execute the command. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-cmd <*command to execute*>

Specifies the complete local disk pathname of the command to execute (for example, `/sbin/reboot`). Surround this argument with double quotes ("") if the command contains one or more spaces.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local `/usr/afs/etc/KeyFile` file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command reboots the machine `fs2.abc.com`. The issuer has previously issued the bos shutdown command to shutdown all processes cleanly.

```
% bos exec -server fs2.abc.com -cmd /sbin/shutdown -r now
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"bos(8)" on page 661

Copyright

bos getdate

Name

`bos getdate` — Displays the time stamps on an AFS binary file

Synopsis

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

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

Description

The bos getdate command displays the time stamps on the current version, `.BAK` version (if any) and `.OLD` version (if any) of each binary file named by the -file argument. (The BOS Server automatically creates `.BAK` and `.OLD` versions when new binaries are installed with the bos install command.) The files must reside in the */usr/afs/bin* directory on the server machine named by the -server argument unless the -dir argument indicates an alternate directory.

To revert to the `.BAK` version of a binary, use the bos uninstall command. To remove obsolete binary files from the */usr/afs/bin* directory, use the bos prune command.

Options

-server <*machine name*>

Indicates the server machine from which to list binary files. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

All server machines of the same AFS system type show the same timestamps if the binaries were installed properly on the binary distribution machine for this machine's system type, and if all other machines of that type are running the appropriate `upclientbin` process.

-file <*files to check*>+

Names each binary file to list.

-dir <*destination dir*>

Specifies the complete pathname of the local disk directory containing each file named by the -file argument. It is necessary only if the files are not in the */usr/afs/bin* directory.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Output

For each file specified with the -file argument, the output displays the time stamp on the current (unmarked), `.BAK`, and `.OLD` version. The output explicitly reports that a version does not exist, rather than simply omitting it.

Examples

The following command examines the time stamps on the files with basename `kaserver` on the machine `fs2.abc.com`:

```
% bos getdate -server fs2.abc.com -file kaserver
File /usr/afs/bin/kaserver dated Mon Jan 4 10:00:36 1999.
.BAK file dated Wed Dec 9 18:55:04 1998, no .OLD file.
```

Privilege Required

None

See Also

"KeyFile(5)" on page 423, "bos(8)" on page 661, "bos_install(8)" on page 701, "bos_prune(8)" on page 713, "bos_uninstall(8)" on page 757

Copyright

bos getlog

Name

`bos getlog` — Prints a server process's log file

Synopsis

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

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

Description

The bos getlog command displays on the standard output stream the specified log file from the machine named by the -server argument. The BOS Server fetches the log file from the */usr/afs/logs* directory unless an alternate pathname is provided as part of the -file argument.

Cautions

Log files can grow quite large, especially for the database server processes. To keep them to a manageable size, periodically either use the UNIX rm command to truncate each log file, or use the bos restart command to restart each process.

It can take up to five minutes after the file is removed or process restarted for the space occupied by a log file to become available.

Options

-server <*machine name*>

Indicates the server machine from which to retrieve the log file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-file <*log file to examine*>

Names the log file to display. If a filename only is provided, the BOS Server fetches the log file from the */usr/afs/logs* directory; the standard values are:

AuthLog

The Authentication Server (kaserver) log file.

BackupLog

The Backup Server (buserver) log file.

BosLog

The BOS Server (bosserver) log file.

FileLog

The File Server (fileserver) log file.

SalvageLog

The Salvager (salvager) log file.

VLLog

The Volume Location (VL) Server (vlserver) log file.

VolserLog

The Volume Server (volserver) log file.

If a pathname and filename are provided, the log file is retrieved from the indicated directory. Partial pathnames are interpreted relative to the */usr/afs/logs* directory.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output is preceded by the line

```
Fetching log file '&lt;filename&gt;'...
```

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

Examples

The following example displays the FileLog file from the machine `fs3.abc.com`:

```
% bos getlog -server fs3.abc.com -file FileLog
Fetching log file 'FileLog'...
Sun Nov 8 04:00:34 1998 File server starting
Sun Nov 8 04:00:39 1998 Partition /vicepa:  attached 21 volumes;
                        0 volumes not attached
Sun Nov 8 04:00:40 1998 File Server started Sun Nov 8 04:00:40
                        1998
Mon Nov 9 21:45:06 1998 CB: RCallBack (zero fid probe in host.c)
                        failed for host 28cf37c0.22811
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"bos(8)" on page 661

Copyright

bos getrestart

Name

`bos getrestart` — Displays the automatic restart times for server processes

Synopsis

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

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

Description

The bos getrestart command displays two restart times from the */usr/afs/local/BosConfig* file on the server machine named by the -server argument:

- The *general restart* time at which the BOS Server process automatically restarts itself and all processes marked with status Run in the *BosConfig* file. The default is Sunday at 4:00 a.m.
- The *binary restart* time at which the BOS Server automatically restarts any process for which the time stamp on the binary file in the */usr/afs/bin* directory is later than the last restart time for the process. The default is 5:00 a.m. Use the bos getdate command to list a binary file's timestamp, and the -long flag to the bos status command to display a process's most recent restart time.

Use the bos setrestart command to set the restart times.

Options

-server <*machine name*>

Indicates the server machine for which to display the restart times. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication.

Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output consists of two lines:

```
Server &lt;machine_name&gt; restarts at &lt;time&gt;
Server &lt;machine_name&gt; restarts for new binaries at &lt;time&gt;
```

Possible values for <time> include:

- `never`, indicating that the BOS Server never performs that type of restart.
- `now`, indicating that the BOS Server performs that type of restart only each time it restarts.
- 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`.

Examples

The following example displays the restart times for the machine `db2.abc.com`:

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

In the following example, the issuer abbreviates the machine name `fs1.abc.com` to `fs1`, relying on the cell's name server to resolve the name. The output echoes the abbreviated form.

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

Privilege Required

None

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "bos(8)" on page 661, "bos_getdate(8)" on page 689, "bos_setrestart(8)" on page 739, "bos_status(8)" on page 751

Copyright

bos help

Name

`bos help` — Displays help for bos commands

Synopsis

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

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

Description

The bos help command displays the complete online help entry (short description and syntax statement) for each command operation code specified by the -topic argument. If the -topic argument is omitted, the output includes the first line (name and short description) of the online help entry for every bos command.

To list every bos command whose name or short description includes a specified keyword, use the bos apropos command.

Options

-topic <*help string*>+

Indicates each command for which to display the complete online help entry. Omit the bos part of the command name, providing only the operation code (for example, specify status, not bos status). If this argument is omitted, the output briefly describes every bos command.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The first line names the command and briefly describes its function.
- The second line lists aliases for the command, if any.
- The final line, which begins with the string `Usage`, lists the command's options in the prescribed order. Online help entries use the same symbols (for example, brackets) as the reference pages in this document.

Examples

The following command displays the online help entry for the bos status command:

```
% bos help status
bos status: show server instance status
Usage: bos status -server &lt;machine name&gt; [-instance &lt;server
process name&gt;+] [-long] [-cell &lt;cell name&gt;] [-noauth]
[-localauth] [-help]
```

Privilege Required

None

See Also

"bos(8)" on page 661, "bos_apropos(8)" on page 675

Copyright

bos install

Name

`bos install` — Installs a new version of a binary file

Synopsis

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

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

Description

The bos install command copies each binary file specified with the -file argument to the local disk of the server machine named by the -server argument, which is normally the binary distribution machine for its CPU / operating system type. The destination directory is */usr/afs/bin* unless the -dir argument indicates an alternate directory. The source file's UNIX mode bits are preserved in the transfer.

If there is already a file of the same name in the destination directory, the BOS Server automatically saves it by adding a `.BAK` extension. 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` version becomes the `.OLD` version automatically. The bos getdate command displays the timestamps on the current versions of the file.

To start using the new binary immediately, issue the bos restart command. Otherwise, the BOS Server automatically restarts the process at the time defined in the */usr/afs/local/BosConfig* file; use the bos getrestart command to display the time and the bos setrestart time to set it.

Options

-server <*machine name*>

Indicates the binary distribution machine on which to install the new binaries. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

If the machine is not a binary distribution machine and is running an `upclientbin` process, then the files are overwritten the next time the `upclientbin` process fetches the corresponding file from the distribution machine (by default within five minutes).

-file <*files to install*>+

Specifies the complete pathname of each binary file to copy into the destination directory. Each source directory can be on the local disk or in AFS, in which case the issuer of the bos install command must have the necessary AFS access rights and the local machine must run the Cache Manager. For the BOS Server to create `.BAK` and `.OLD` versions, the last

element in the pathname (the filename) must match the name of a file in the destination directory. "bos_create(8)" on page 677 lists the standard binary file names.

-dir <*destination dir*>

Provides the complete pathname of the local disk directory in which to install binary files. It is necessary only if the destination directory is not */usr/afs/bin*.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command copies the file */afs/abc.com/rs_aix42/usr/afs/bin/vlserver* to the file */usr/afs/bin/vlserver* on the machine fs3.abc.com, which is the binary distribution machine for server machines running AIX 4.2 in the abc.com cell. The current version of the */usr/afs/bin/vlserver* file is moved to */usr/afs/bin/vlserver.BAK*.

```
% bos install -server fs3.abc.com \
              -file /afs/abc.com/rs_aix42/usr/afs/bin/vlserver
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser root if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_getdate(8)" on page 689, "bos_getrestart(8)" on page 695, "bos_restart(8)" on page 723, "bos_setrestart(8)" on page 739

Copyright

bos listhosts

Name

`bos listhosts` — Displays the contents of the CellServDB file

Synopsis

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

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

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

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

Description

The bos listhosts command formats and displays the list of a cell's database server machines from the */usr/afs/etc/CellServDB* file on the server machine named by the -server argument.

To alter the list of machines, use the bos addhost and bos removehost commands.

Options

-server <*machine name*>

Indicates the server machine from which to display the */usr/afs/etc/CellServDB* file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

For consistent performance in the cell, the output must be the same on every server machine. The bos addhost reference page explains how to keep the machines synchronized.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of the output names the cell to which the 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 IP address. The BOS Server assigned it as part of performing the bos addhost command.

Examples

The following command displays the database server machines listed in the */usr/afs/etc/CellServDB* file on the machine `fs7.abc.com`.

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

Privilege Required

None

See Also

"CellServDB(5)" on page 415, "KeyFile(5)" on page 423, "bos(8)" on page 661, "bos_addhost(8)" on page 665, "bos_removehost(8)" on page 717

Copyright

bos listkeys

Name

`bos listkeys` — Displays the server encryption keys from the KeyFile file

Synopsis

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

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

Description

The bos listkeys command formats and displays the list of server encryption keys from the `/usr/afs/etc/KeyFile` file on the server machine named by the -server argument.

To edit the list of keys, use the bos addkey and bos removekey commands.

Cautions

Displaying actual keys on the standard output stream (by including the -showkey flag) is a security exposure. Displaying a checksum is sufficient for most purposes.

Options

-server <*machine name*>

Indicates the server machine from which to display the KeyFile file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

For consistent performance in the cell, the output must be the same on every server machine. The bos addkey reference page explains how to keep the machines synchronized.

-showkey

Displays the octal digits that constitute each key.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

If the -showkey flag is included, the output displays the actual string of eight octal numbers that constitute the key. Each octal number is a backslash and three decimal digits.

If the -showkey flag is not included, the output represents each key as a checksum, which is a decimal number derived by encrypting a constant with the key.

Following the list of keys or checksums, the string `Keys last changed` indicates when a key was last added to the *KeyFile* file. The words `All done` indicate the end of the output.

For mutual authentication to work properly, the output from the command `kas examine afs` must match the key or checksum with the same key version number in the output from this command.

Examples

The following example shows the checksums for the keys stored in the *KeyFile* file on the machine `fs3.abc.com`.

```
% bos listkeys fs3.abc.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 Mon Apr 12 11:24:46 1999.
All done.
```

The following example shows the actual keys from the *KeyFile* file on the machine `fs6.abc.com`.

```
% bos listkeys fs6.abc.com -showkey
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\253\326\236\261\211'
Keys last changed on Wed Mar 31 11:24:46 1999.
All done.
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos_addkey(8)" on page 669, "bos_removekey(8)" on page 719, "bos_setauth(8)" on page 733, "kas_examine(8)" on page 817

Copyright

bos listusers

Name

`bos listusers` — Lists the privileged users from the UserList file

Synopsis

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

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

Description

The bos listusers command lists the user names from the */usr/afs/etc/UserList* file on the file server machine named by the -server argument. The users are authorized to issue privileged bos and vos commands.

To edit the list of users, use the bos adduser and bos removeuser commands.

Options

-server <*machine name*>

Indicates the server machine from which to display the UserList file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

For consistent performance in the cell, the output must be the same on every server machine. The bos adduser reference page explains how to keep the machines synchronized.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output lists the user name of each user entitled to issue privileged bos and vos commands.

Examples

The following example lists the users from UserList file on the machine `fs4.abc.com`.

```
% bos listusers fs4.abc.com
SUsers are: pat smith jones terry
```

Privilege Required

None

See Also

"KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_adduser(8)" on page 673, "bos_removeuser(8)" on page 721

Copyright

bos prune

Name

`bos prune` — Removes obsolete files from /usr/afs/bin and /usr/afs/logs

Synopsis

bos prune -server <*machine name*> [-bak] [-old] [-core] [-all] [-cell <*cell name*>] [-noauth] [-localauth] [-help]

bos p -s <*machine name*> [-b] [-o] [-co] [-a] [-ce <*cell name*>] [-n] [-l] [-h]

Description

The bos prune command removes files from the local disk of the server machine named by the -server argument, as specified by one or more of the following flags provided on the command line:

- The -bak flag removes all files from the */usr/afs/bin* directory that have a `.BAK` extension.
- The -old flag removes all files from the */usr/afs/bin* directory that have a `.OLD` extension.
- The -core flag removes all files from the */usr/afs/logs* directory that have a `core.` prefix.
- The -all flag removes all three types of files at once.

(If none of these flags are included, the command appears to succeed, but removes no files at all.)

To display the timestamp on the current, `.BAK`, and `.OLD` versions of one or more files, use the bos getdate command.

Options

-server <*machine name*>

Indicates the server machine from which to remove files. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-bak

Removes all files from the */usr/afs/bin* directory that have a `.BAK` extension. Do not combine this flag and the -all flag.

-old

Removes all files from the */usr/afs/bin* directory that have a `.OLD` extension. Do not combine this flag and the -all flag.

-core

Removes all files from the */usr/afs/logs* directory that have a `core.` prefix. Do not combine this flag and the -all flag.

-all

Combines the effect of the -bak, -old, and -core flags. Do not combine this flag with any of those three.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example removes all files from the */usr/afs/bin* directory on the machine `fs3.abc.com` that have a `.BAK` or `.OLD` extension.

```
% bos prune -server fs3.abc.com -bak -old
```

The following example removes all files from the */usr/afs/bin* directory on the machine `db2.abc.com` that have a `.BAK` or `.OLD` extension, and all files from the */usr/afs/logs* directory that have a `core.` prefix.

```
% bos prune -server db2.abc.com -all
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_getdate(8)" on page 689

Copyright

bos removehost

Name

`bos removehost` — Removes a database server machine from the CellServDB file

Synopsis

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

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

Description

The bos removehost command removes the entry for each database server machine specified with the -host argument from the */usr/afs/etc/CellServDB* file on the server machine named by the -server argument.

Cautions

After executing this command (and waiting for the Update Server to propagate the changes, if it is used), restart the database server processes on all database server machines to force election of a quorum that includes the new set of machines listed in the */usr/afs/etc/CellServDB* file. The *IBM AFS Quick Beginnings* explains in more detail how to add and remove database server machines.

Options

-server <*machine name*>

Indicates the server machine on which to change the */usr/afs/etc/CellServDB* file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

In cells that use the Update Server to distribute the contents of the */usr/afs/etc* directory, it is conventional to specify only the system control machine as a value for the -server argument. Otherwise, repeat the command for each file server machine. For further discussion, see "bos(8)" on page 661.

-host <*host name*>+

Specifies the fully-qualified host name (such as `fs2.abc.com`) of each database server machine to remove from the CellServDB file.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command removes the former database server machine `db2.abc.com` from the *CellServDB* file on the system control machine `fs1.abc.com`.

```
% bos removehost -server fs1.abc.com -host db2.abc.com
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_addhost(8)" on page 665, "bos_listhosts(8)" on page 705

IBM AFS Quick Beginnings

Copyright

bos removekey

Name

`bos removekey` — Removes a server encryption key from the KeyFile file

Synopsis

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

bos removek -s <*machine name*> -k <*key version number*>+ [-c <*cell name*>] [-n] [-l] [-h]

Description

The bos removekey command removes each specified encryption key from the `/usr/afs/etc/KeyFile` file on the machine named by the -server argument. Use the -kvno argument to identify each key by its key version number; use the bos listkeys command to display the key version numbers.

Cautions

Before removing a obsolete key, verify that the cell's maximum ticket lifetime has passed since the current key was defined using the kas setpassword and bos addkey commands. This ensures that no clients still possess tickets encrypted with the obsolete key.

Options

-server <*machine name*>

Indicates the server machine on which to change the `/usr/afs/etc/KeyFile` file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

In cells that use the Update Server to distribute the contents of the `/usr/afs/etc` directory, it is conventional to specify only the system control machine as a value for the -server argument. Otherwise, repeat the command for each file server machine. For further discussion, see "bos(8)" on page 661.

-kvno <*key version number*>+

Specifies the key version number of each key to remove.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command removes the keys with key version numbers 5 and 6 from the *KeyFile* file on the system control machine `fs1.abc.com`.

```
% bos removekey -server fs1.abc.com -kvno 5 6
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_addkey(8)" on page 669, "bos_listkeys(8)" on page 707

Copyright

bos removeuser

Name

`bos removeuser` — Removes a privileged user from the UserList file

Synopsis

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

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

Description

The bos removeuser command removes each user name specified with the -user argument from the */usr/afs/etc/UserList* file on the machine named by the -server argument.

Options

-server <*machine name*>

Indicates the server machine on which to change the */usr/afs/etc/UserList* file. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

In cells that use the Update Server to distribute the contents of the */usr/afs/etc* directory, it is conventional to specify only the system control machine as a value for the -server argument. Otherwise, repeat the command for each file server machine. For further discussion, see "bos(8)" on page 661.

-user <*user names*>+

Specifies each user name to remove.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example removes the users `pat` and `jones` from the *`UserList`* file on the system control machine `fs1.abc.com`.

```
% bos removeuser -server fs1.abc.com -user pat jones
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_addkey(8)" on page 669, "bos_listkeys(8)" on page 707

Copyright

bos restart

Name

`bos restart` — Restarts a server process

Synopsis

bos restart -server <*machine name*> [-instance <*instances*>+] [-bosserver] [-all] [-cell <*cell name*>] [-noauth] [-localauth] [-help]

bos res -s <*machine name*> [-i <*instances*>+] [-b] [-a] [-c <*cell name*>] [-n] [-l] [-h]

Description

The bos restart command stops and immediately restarts server processes on the server machine named by the -server argument. Indicate which process or processes to restart by providing one of the following arguments:

- The -instance argument names each AFS server process to stop and restart immediately, regardless of its status flag in the */usr/afs/local/BosConfig* file. Do not include bosserver in the list of processes; use the -bosserver flag instead.
- The -bosserver flag stops all AFS server processes running on the machine, including the BOS Server. A new BOS Server starts immediately, and it starts a new instance of each process that is marked with the `Run` status flag in the *BosConfig* file.
- The -all flag stops all AFS server processes running on the machine, except the BOS Server, and immediately restarts the processes that are marked with the `Run` status flag in the *BosConfig* file.

This command does not change a process's status flag in the *BosConfig* file.

Options

-server <*machine name*>

Indicates the server machine on which to restart each process. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-instance <*instances*>+

Names each process to stop and then restart immediately regardless of its status flag setting. Use the process name assigned with the -instance argument to the bos create command. The output from the bos status command lists the names. Provide this flag or one of the -bosserver or -all options, but do not combine them.

-bosserver

Stops all AFS server processes running on the machine, including the BOS Server. A new BOS Server instance immediately starts, and starts all processes marked with the `Run` status flag in the *`BosConfig`* file. Provide this flag or one of the -instance or -all options, but do not combine them.

-all

Stops all AFS server processes running on the machine other than the BOS Server, and immediately restarts the processes marked with the Run status flag in the *`BosConfig`* file. Provide this flag or one of the -instance or -bosserver options, but do not combine them.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command stops and restarts all processes running on the machine `fs3.abc.com`, including the BOS Server.

```
% bos restart -server fs3.abc.com -bosserver
```

The following command stops and restarts all processes running on the machine `fs5.abc.com`, excluding the BOS Server.

```
% bos restart -server fs5.abc.com -all
```

The following command stops and restarts the Protection Server and Volume Location (VL) Server processes on the machine `db3.abc.com`:

```
% bos restart -server db3.abc.com -instance ptserver vlserver
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_status(8)" on page 751

Copyright

bos salvage

Name

`bos salvage` — Restores internal consistency to a file system or volume

Synopsis

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*>] [-orphans (ignore | remove | attach)] [-cell <*cell name*>] [-noauth] [-localauth] [-help]

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*>] [-o (ignore | remove | attach)] [-c <*cell name*>] [-n] [-l] [-h]

Description

The bos salvage command salvages (restores internal consistency to) one or more volumes on the file server machine named by the -server argument. When processing one or more partitions, the command restores consistency to corrupted read / write volumes where possible. For read-only or backup volumes, it inspects only the volume header:

- If the volume header is corrupted, the Salvager removes the volume completely and records the removal in its log file, `/usr/afs/logs/SalvageLog`. Issue the vos release or vos backup command to create the read-only or backup volume again.
- If the volume header is intact, the Salvager skips the volume (does not check for corruption in the contents). However, if the File Server notices corruption as it initializes, it sometimes refuses to attach the volume or bring it online. In this case, it is simplest to remove the volume by issuing the vos remove or vos zap command. Then issue the vos release or vos backup command to create it again.

Use the indicated arguments to salvage a specific number of volumes:

- To process all volumes on a file server machine, provide the -server argument and the -all flag. No volumes on the machine are accessible to Cache Managers during the salvage operation, because the BOS Server stops the File Server and Volume Server processes while the Salvager runs. The BOS Server automatically restarts them when the operation completes.
- To process all volumes on one partition, provide the -server and -partition arguments. As for a salvage of the entire machine, no volumes on the machine are accessible to Cache Managers during the salvage operation. The BOS Server automatically restarts the File Server and Volume Server when the operation completes.
- To salvage only one read / write volume, combine the -server, -partition, and -volume arguments. Only that volume is inaccessible to Cache Managers, because the BOS Server does not shutdown the File Server and Volume Server processes during the salvage of a

single volume. Do not name a read-only or backup volume with the -volume argument. Instead, remove the volume, using the vos remove or vos zap command. Then create a new copy of the volume with the vos release or vos backup command.

During the salvage of an entire machine or partition, the bos status command reports the `fs` process's auxiliary status as `Salvaging file system`.

The Salvager always writes a trace to the */usr/afs/logs/SalvageLog* file on the file server machine where it runs. To record the trace in another file as well (either in AFS or on the local disk of the machine where the bos salvage command is issued), name the file with the -file argument. To display the trace on the standard output stream as it is written to the */usr/afs/logs/SalvageLog* file, include the -showlog flag.

By default, multiple Salvager subprocesses run in parallel: one for each partition up to four, and four subprocesses for four or more partitions. To increase or decrease the number of subprocesses running in parallel, provide a positive integer value for the -parallel argument.

If there is more than one server partition on a physical disk, the Salvager by default salvages them serially to avoid the inefficiency of constantly moving the disk head from one partition to another. However, this strategy is often not ideal if the partitions are configured as logical volumes that span multiple disks. To force the Salvager to salvage logical volumes in parallel, provide the string `all` as the value for the -parallel argument. Provide a positive integer to specify the number of subprocesses to run in parallel (for example, `-parallel 5all` for five subprocesses), or omit the integer to run up to four subprocesses, depending on the number of logical volumes being salvaged.

The Salvager creates temporary files as it runs, by default writing them to the partition it is salvaging. The number of files can be quite large, and if the partition is too full to accommodate them, the Salvager terminates without completing the salvage operation (it always removes the temporary files before exiting). Other Salvager subprocesses running at the same time continue until they finish salvaging all other partitions where there is enough disk space for temporary files. To complete the interrupted salvage, reissue the command against the appropriate partitions, adding the -tmpdir argument to redirect the temporary files to a local disk directory that has enough space.

The -orphans argument controls how the Salvager handles orphaned files and directories that it finds on server partitions it is salvaging. An *orphaned* element is completely inaccessible because it is not referenced by the vnode of any directory that can act as its parent (is higher in the filespace). Orphaned objects occupy space on the server partition, but do not count against the volume's quota.

Cautions

Running this command can result in data loss if the Salvager process can repair corruption only by removing the offending data. Consult the *IBM AFS Administration Guide* for more information.

Options

-server <*machine name*>

Indicates the file server machine on which to salvage volumes. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-partition <*salvage partition*>

Specifies a single partition on which to salvage all volumes. Provide the complete partition name (for example */vicepa*) or one of the following abbreviated forms:

```
/vicepa     =     vicepa      =      a      =      0
/vicepb     =     vicepb      =      b      =      1
```

After */vicepz* (for which the index is 25) comes

```
/vicepaa    =     vicepaa     =      aa     =      26
/vicepab    =     vicepab     =      ab     =      27
```

and so on through

```
/vicepiv    =     vicepiv     =      iv     =      255
```

-volume <*salvage volume id or name*>

Specifies the name or volume ID number of a read / write volume to salvage. The -partition argument must be provided along with this one.

-file <*salvage log output file*>

Specifies the complete pathname of a file into which to write a trace of the salvage operation, in addition to the */usr/afs/logs/SalvageLog* file on the server machine. If the file pathname is local, the trace is written to the specified file on the local disk of the machine where the bos salvage command is issued. If the -volume argument is included, the file can be in AFS, though not in the volume being salvaged. Do not combine this argument with the -showlog flag.

-all

Salvages all volumes on all of the partitions on the machine named by the -server argument.

-showlog

Displays the trace of the salvage operation on the standard output stream, as well as writing it to the */usr/afs/logs/SalvageLog* file. Do not combine this flag with the -file argument.

-parallel <*# of max parallel partition salvaging*>

Specifies the maximum number of Salvager subprocesses to run in parallel. Provide one of three values:

- An integer from the range `1` to `32`. A value of `1` means that a single Salvager process salvages the partitions sequentially.
- The string `all` to run up to four Salvager subprocesses in parallel on partitions formatted as logical volumes that span multiple physical disks. Use this value only with such logical volumes.
- The string all followed immediately (with no intervening space) by an integer from the range `1` to `32`, to run the specified number of Salvager subprocesses in parallel on partitions formatted as logical volumes. Use this value only with such logical volumes.

The BOS Server never starts more Salvager subprocesses than there are partitions, and always starts only one process to salvage a single volume. If this argument is omitted, up to four Salvager subprocesses run in parallel.

-tmpdir <*directory to place tmp files*>

Specifies the full pathname of a local disk directory to which the Salvager process writes temporary files as it runs. If this argument is omitted, or specifies an ineligible or nonexistent directory, the Salvager process writes the files to the partition it is currently salvaging.

-orphans (ignore | remove | attach)

Controls how the Salvager handles orphaned files and directories. Choose one of the following three values:

ignore

Leaves the orphaned objects on the disk, but prints a message to the */usr/afs/logs/SalvageLog* file reporting how many orphans were found and the approximate number of kilobytes they are consuming. This is the default if the -orphans argument is omitted.

remove

Removes the orphaned objects, and prints a message to the */usr/afs/logs/SalvageLog* file reporting how many orphans were removed and the approximate number of kilobytes they were consuming.

attach

Attaches the orphaned objects by creating a reference to them in the vnode of the volume's root directory. Since each object's actual name is now lost, the Salvager assigns each one a name of the following form:

- `__ORPHANFILE__.`*index* for files.
- `__ORPHANDIR__.`*index* for directories.

where *index* is a two-digit number that uniquely identifies each object. The orphans are charged against the volume's quota and appear in the output of the ls command issued against the volume's root directory.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command salvages all volumes on the */vicepd* partition of the machine `db3.abc.com`:

```
% bos salvage -server db3.abc.com -partition /vicepd
```

The following command salvages the volume with volume ID number 536870988 on partition */vicepb* of the machine `fs2.abc.com`:

```
% bos salvage -server fs2.abc.com -partition /vicepb -volume 536870988
```

The following command salvages all volumes on the machine `fs4.abc.com`. Six Salvager processes run in parallel rather than the default four.

```
% bos salvage -server fs4.abc.com -all -parallel 6
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "SalvageLog(5)" on page 433, "UserList(5)" on page 437, "bos(8)" on page 661, "salvager(8)" on page 879, "vos_backup(1)" on page 299, "vos_release(1)" on page 355, "vos_remove(1)" on page 359, "vos_zap(1)" on page 389

IBM AFS Administration Guide

Copyright

bos setauth

Name

`bos setauth` — Sets authorization checking requirements for all server processes

Synopsis

bos setauth -server <*machine name*> -authrequired (on | off) [-cell <*cell name*>] [-noauth] [-localauth] [-help]

bos seta -s <*machine name*> -a (on | off) [-c <*cell name*>] [-n] [-l] [-h]

Description

The bos setauth command enables or disables authorization checking on the server machine named by the -server argument. When authorization checking is enabled (the normal case), the AFS server processes running on the machine verify that the issuer of a command meets its privilege requirements. When authorization checking is disabled, server processes perform any action for anyone, including the unprivileged user `anonymous`; this security exposure precludes disabling of authorization checking except during installation or emergencies.

To indicate to the server processes that authorization checking is disabled, the BOS Server creates the zero-length file `/usr/afs/local/NoAuth` on its local disk. All AFS server processes constantly monitor for the `NoAuth` file's presence and do not check for authorization when it is present. The BOS Server removes the file when this command is used to reenable authorization checking.

Cautions

Do not create the `NoAuth` file directly, except when directed by instructions for dealing with emergencies (doing so requires being logged in as the local superuser `root`). Use this command instead.

Options

-server <*machine name*>

Indicates the server machine on which to enable or disable authorization checking. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-authrequired (on | off)

Enables authorization checking if the value is `on`, or disables it if the value is `off`.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example disables authorization checking on the machine `fs7.abc.com`:

```
% bos setauth -server fs7.abc.com -authrequired off
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"KeyFile(5)" on page 423, "NoAuth(5)" on page 429, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_restart(8)" on page 723

Copyright

bos setcellname

Name

`bos setcellname` — Sets the cell's name in ThisCell and CellServDB

Synopsis

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

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

Description

The bos setcellname command establishes the cell's name and makes the server machine named by the -server argument a member of it, by recording the value of the -name argument in two files which it creates on the local disk:

- */usr/afs/etc/ThisCell*
- */usr/afs/etc/CellServDB*. The cell name appears on the first line in the file, preceded by the required > symbol. The machine name specified with the -server argument appears on the second line along with its IP address as obtained from the cell's naming service. The machine is thus designated as the cell's first database server machine.

Cautions

Issue this command only when the installing the cell's first AFS server machine. The *IBM AFS Quick Beginnings* explains how to copy over the *ThisCell* and *CellServDB* files from this or another appropriate machine during installation of additional server machines.

Be sure to choose a satisfactory cell name when issuing this command, because changing a cell's name is very complicated; for one thing, it requires changing every password in the Authentication Database. Consult the *IBM AFS Administration Guide* for advice on choosing a cell name.

Options

-server <*machine name*>

Indicates the server machine on which to set the cell name in the *ThisCell* and *CellServDB* file. It is always the first machine installed in a cell. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-name <*cell name*>

Defines the cell name, using standard Internet domain name format (the actual domain name is usually appropriate). Examples are `abc.com` for the ABC Corporation and `stateu.edu` for the State University. It must match the value of the -cell argument, if that is provided.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command defines the cell name `abc.com` in the *ThisCell* and *CellServDB* files on the machine `fs1.abc.com` as it is installed as the cell's first server machine.

```
% bos setcellname -server fs1.abc.com -name abc.com
```

Privilege Required

Authorization checking is normally turned off during installation, which is the only recommended time to use this command; in this case no privilege is required. If authorization checking is turned on, the issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged in as the local superuser `root` if the -localauth flag is included.

See Also

"CellServDB(5)" on page 415, "KeyFile(5)" on page 423, "ThisCell(5)" on page 435, "UserList(5)" on page 437, "bos(8)" on page 661

IBM AFS Quick Beginnings

Copyright

bos setrestart

Name

`bos setrestart` — Sets when the BOS Server restarts processes

Synopsis

bos setrestart -server <*machine name*> -time <*time to restart server*> [-general] [-newbinary] [-cell <*cell name*>] [-noauth] [-localauth] [-help]

bos setr -s <*machine name*> -t <*time to restart server*> [-g] [-ne] [-c <*cell name*>] [-no] [-l] [-h]

Description

The bos setrestart command records in the */usr/afs/local/BosConfig* file the times at which the BOS Server running on the server machine named by the -server argument performs two types of restarts:

- A *general restart*. By default, once per week the BOS Server restarts itself and then any AFS process marked with the `Run` status flag in the *`BosConfig`* file (equivalent in effect to issuing the bos restart command with the -bosserver flag). The default setting is 4:00 a.m. each Sunday morning.
- A *binary restart*. By default, once per day the BOS Server restarts any currently running process for which the timestamp on the binary file in the */usr/afs/bin* directory is later than the time the process last started or restarted. The default is 5:00 a.m. each day.

Cautions

Restarting a process makes it unavailable for a period of time. The fs process has potentially the longest outage, depending on how many volumes the file server machine houses (the File Server and Volume Server reattach each volume when they restart). The default settings are designed to coincide with periods of low usage, so that the restarts disturb the smallest possible number of users.

If the setting specified with the -time argument is within one hour of the current time, the BOS Server does not restart any processes until the next applicable opportunity (the next day for binary restarts, or the next week for general restarts).

The command changes only one type of restart setting at a time; issue the command twice to change both settings.

Options

-server <*machine name*>

Indicates the server machine on which to set a new restart time. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-time <*time to restart server*>

Specifies the restart time. By convention the general restart is defined as weekly (specifies both a day and a time), and the binary restart is defined as daily (specifies only a time). However, it is acceptable to define a daily general restart or weekly binary restart.

There are four acceptable values for either type of restart setting:

- The string `never`, which directs the BOS Server never to perform the indicated type of restart.
- The string `now`, which directs the BOS Server to perform the restart immediately and never again.
- A time of day (the conventional type of value for the binary restart time). Separate the hours and minutes with a colon (*hh:MM*), an use either 24-hour format, or a value in the range from `1:00` through `12:59` with the addition of `am` or `pm`. For example, both `14:30` and `"2:30 pm"` indicate 2:30 in the afternoon. Surround this parameter with double quotes (`""`) if it contains a space.
- A day of the week and time of day, separated by a space and surrounded with double quotes (`""`). This is the conventional type of value for the general restart. For the day, provide either the whole name or the first three letters, all in lowercase letters (`sunday` or `sun`, `thursday` or `thu`, and so on). For the time, use the same format as when specifying the time alone.

If desired, precede a time or day and time definition with the string `every` or `at`. These words do not change the meaning, but possibly make the output of the bos getrestart command easier to understand.

-general

Sets the general restart time.

-newbinary

Sets the binary restart time.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command sets the general restart time on the machine `fs4.abc.com` to Saturday at 3:30 am.

```
% bos setrestart -server fs4.abc.com -time "sat 3:30" -general
```

The following command sets the binary restart time on the machine `fs6.abc.com` to 11:45 pm.

```
% bos setrestart -server fs6.abc.com -time 23:45 -newbinary
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_getrestart(8)" on page 695, "bos_restart(8)" on page 723

Copyright

bos shutdown

Name

`bos shutdown` — Stops a process without changing its status flag

Synopsis

bos shutdown -server <*machine name*> [-instance <*instances*>+] [-wait] [-cell <*cell name*>] [-noauth] [-localauth] [-help]

bos sh -s <*machine name*> [-i <*instances*>+] [-w] [-c <*cell name*>] [-n] [-l] [-h]

Description

The bos shutdown command stops, on the server machine named by the -server argument, either

- All of the currently running AFS server processes, except the BOS Server.
- Only the processes specified by the -instance argument.

This command does not change a process's status flag in the */usr/afs/local/BosConfig* file, but only in the BOS Server's memory. To stop a process and change its *BosConfig* status flag, use the bos stop command instead.

Once stopped with this command, a process does not run again until an administrator starts it by using the bos start, bos startup, or bos restart command, or until the BOS Server restarts (assuming that the process's *BosConfig* status flag is `Run`).

Options

-server <*machine name*>

Indicates the server machine on which to stop processes. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-instance <*instances*>+

Names each process to stop. Use the process name assigned with the -instance argument to the bos create command. The output from the bos status command lists the names. Omit this argument to stop all processes other than the BOS Server.

-wait

Delays the return of the command shell prompt until all processes actually stop. If this argument is omitted, the prompt returns almost immediately even if all processes are not stopped.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command stops all processes other than the BOS Server on the machine `fs3.abc.com`.

```
% bos shutdown fs3.abc.com
```

The following command stops the `upserver` process (server portion of the Update Server) on the machine `fs5.abc.com`.

```
% bos shutdown -server fs5.abc.com -instance upserver
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_restart(8)" on page 723, "bos_start(8)" on page 747, "bos_startup(8)" on page 749, "bos_status(8)" on page 751

Copyright

bos start

Name

`bos start` — Starts a process after setting its status flag

Synopsis

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

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

Description

The bos start command sets the status flag for each process specified by the -instance argument to `Run` in the */usr/afs/local/BosConfig* file and in the BOS Server's memory on the server machine named by the -server argument, then starts it. If the process is already running, the command's only effect is to guarantee that the status flag is `Run`; it does not restart the process.

To start a process without changing its status flag in the *BosConfig* file, use the bos startup command instead.

Options

-server <*machine name*>

Indicates the server machine on which to start processes. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-instance <*server process name*>+

Names each process to start. Use the process name assigned with the -instance argument to the bos create command. The output from the bos status command lists the names.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication.

Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command changes the status flag for the `upclientbin` and `upclientetc` processes to `Run` in the *`BosConfig`* file on the machine `fs6.abc.com` and starts them running.

```
% bos start -server fs6.abc.com -instance upclientbin upclientetc
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_startup(8)" on page 749, "bos_status(8)" on page 751

Copyright

bos startup

Name

`bos startup` — Starts a process without changing its status flag

Synopsis

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

bos startu -s <*machine name*> [-i <*instances*>+] [-c <*cell name*>] [-n] [-l] [-h]

Description

The bos startup command starts, on the server machine named by the -server argument, either:

- All AFS server processes not currently running but marked with the `Run` status flag in the */usr/afs/local/BosConfig* file.
- Each process specified by -instance argument, even if its status flag in the BosConfig file is `NotRun`.

To start a process and set its *BosConfig* status flag to `Run`, use the bos start command instead.

Options

-server <*machine name*>

Indicates the server machine on which to start processes. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-instance <*instances*>+

Names each process to start. Use the process name assigned with the -instance argument to the bos create command. The output from the bos status command lists the names.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command starts all processes marked with status flag Run in the *BosConfig* file on the machine fs3.abc.com that are not currently running.

```
% bos startup fs3.abc.com
```

The following command starts the buserver, kaserver, ptserver, and vlserver processes running on the machine db2.abc.com, even if their status flags in the *BosConfig* file are NotRun.

```
% bos startup -server db2.abc.com \
    -instance buserver kaserver ptserver vlserver
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser root if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_start(8)" on page 747, "bos_status(8)" on page 751

Copyright

bos status

Name

`bos status` — Displays the status of server processes

Synopsis

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

bos stat -s <*machine name*> [-i <*server process name*>+] [-lon] [-c <*cell name*>] [-n] [-loc] [-h]

Description

The bos status command reports the status of processes on the server machine named by the -server argument, either

- All of the AFS server processes listed in the */usr/afs/local/BosConfig* file
- Only these processes named by the -instance argument

Options

-server <*machine name*>

Indicates the server machine for which to report server process status. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-instance

Names each process for which to report status. Use the process name assigned with the -instance argument to the bos command. The output from the bos status command lists the names.

-long

Produces more detailed status information.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity anonymous to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output for a process includes at least one line, which reports one of the following as the process's current status:

- `currently running normally`. The process's status flag in the BosConfig file is `Run`. For cron entries, this message indicates only that the command is scheduled to run, not necessarily that it was executing when the bos status command was issued.
- `disabled`. The process is not running, and its BosConfig status flag is `NotRun`.
- `temporarily disabled`. The process is not running although its status flag in the BosConfig file is `Run`. Either an administrator used the bos shutdown command to stop it, or the
- BOS Server stopped trying to restart it after numerous failed attempts. In the second case, the auxiliary message is `stopped for too many errors`.
- `temporarily enabled`. The process is running although its status flag in the BosConfig file is `NotRun`. An administrator has used the bos startup command to start it.

If one of the following special circumstances applies to the process, the indicated message appears in its entry:

- `has core file`. The process failed and created a core file in the */usr/afs/logs* directory. If the BOS Server was able to restart the process after the failure, the primary status is `currently running normally`.
- `stopped for too many errors`. The reason for the primary status `temporarily disabled` is that the BOS Server's attempts to restart the process all failed.

The entry for the fs process always includes a second line to report the process's `Auxiliary status`, which is one of the following:

- `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 entry for a cron process includes an `Auxiliary status` that reports when the command will next execute.

If the -long flag is used, each entry includes the following additional information:

- The process's type (`simple`, `fs`, or `cron`).
- The day and time the process last started or restarted.
- The number of `proc starts`, which is how many times the BOS Server has started or restarted the process since it started itself.
- The `Last exit` time when the process (or one of the component processes in the fs process) last terminated. This line does not appear if the process has not terminated since the BOS Server started.
- The `Last error exit` time when the process (or one of the component processes in the fs process) last failed due to an error. A further explanation such as `due to shutdown request` sometimes appears. This line does not appear if the process has not failed since the BOS Server started.
- Each command that the BOS Server invokes to start the process, as specified by the -cmd argument to the bos create command.
- The pathname of the notifier program that the BOS Server invokes when the process terminates (if any), as specified by the -notifier argument to the bos create command.

If the -long flag is provided and the BOS Server discovers that the mode bits on files and subdirectories in the local */usr/afs* directory differ from the expected values, it prints the following warning message:

```
Bosserver reports inappropriate access on server directories
```

The following chart summarizes the expected mode bit settings. A question mark indicates that the BOS Server does not check that bit.

Examples

The following example command displays the status of processes on the machine fs3.abc.com:

```
% bos status fs3.abc.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 example command displays a detailed status report for the fs and ptserver processes on the machine fs1.abc.com.

```
% bos status -server fs1.abc.com -instance fs ptserver -long
Instance fs, (type is fs), currently running normally.
   Auxiliary status is: file server running.
   Process last started at Wed Jan 7 5:34:49 1998 (3 proc starts)
```

```
    Last exit at Wed Jan 7 5:34:49 1998
    Last error exit at Wed Jan 7 5:34:49 1998, 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 Tue Jan 6 8:29:19 1998 (1 proc starts)
    Command 1 is '/usr/afs/bin/ptserver'
```

Privilege Required

None

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_shutdown(8)" on page 743, "bos_startup(8)" on page 749, "bos_status(8)" on page 751

Copyright

bos stop

Name

`bos stop` — Stops a process after changing its status flag

Synopsis

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

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

Description

The bos stop command sets the status flag for each process specified with the -instance argument to `NotRun` in the */usr/afs/local/BosConfig* file on the server machine named by the -server argument, then stops it.

To stop a process without changing its *BosConfig* status flag, use the bos shutdown command instead.

Options

-server <*machine name*>

Indicates the server machine on which to stop processes. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

-instance <*server process name*>+

Names each process to stop. Use the process name assigned with the -instance argument to the bos create command. The output from the bos status command lists the names.

-wait

Delays the return of the command shell prompt until all processes actually stop. If this argument is omitted, the prompt returns almost immediately even if all processes are not stopped.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example command stops the upserver and runntp processes on the machine `fs7.abc.com`.

```
% bos stop -server fs7.abc.com -instance upserver runntp
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_shutdown(8)" on page 743, "bos_status(8)" on page 751

Copyright

bos uninstall

Name

`bos uninstall` — Reverts to the former version of a process's binary file

Synopsis

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

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

Description

The bos uninstall command replaces each binary file specified by the -file argument with its `.BAK` version on the server machine named by the -server argument, which is normally the binary distribution machine for its CPU / operating system type. It also changes the extension on the current `.OLD` version (if any) to `.BAK`. Each binary file must reside in the local */usr/afs/bin* directory unless the -dir argument names an alternate directory.

To start using the reverted binary immediately, issue the bos restart command. Otherwise, the BOS Server automatically restarts the process at the time defined in the */usr/afs/local/BosConfig* file; use the bos getrestart command to display the time and the bos setrestart time to set it.

Options

-server <*machine name*>

Indicates the binary distribution machine on which to revert to the `.BAK` version of binaries. Identify the machine by IP address or its host name (either fully-qualified or abbreviated unambiguously). For details, see "bos(8)" on page 661.

If the machine is not a binary distribution machine and is running an `upclientbin` process, then the files are overwritten the next time the `upclientbin` process fetches the corresponding file from the distribution machine (by default within five minutes).

-file <*files to uninstall*>+

Names each binary file to replace with its `.BAK` version.

-dir <*destinatino dir*>

Provides the complete pathname of the local disk directory containing each file named by the -file argument. It is necessary only if the binaries are not in the */usr/afs/bin* directory.

-cell <*cell name*>

Names the cell in which to run the command. Do not combine this argument with the -localauth flag. For more details, see "bos(8)" on page 661.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. Do not combine this flag with the -localauth flag. For more details, see "bos(8)" on page 661.

-localauth

Constructs a server ticket using a key from the local */usr/afs/etc/KeyFile* file. The bos command interpreter presents the ticket to the BOS Server during mutual authentication. Do not combine this flag with the -cell or -noauth options. For more details, see "bos(8)" on page 661.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example command overwrites the */usr/afs/bin/kaserver* file on the machine `fs4.abc.com` with its `.BAK` version, and the current `.BAK` version by the `.OLD` version.

```
% bos uninstall -server fs4.abc.com -file kaserver
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on the machine named by the -server argument, or must be logged onto a server machine as the local superuser `root` if the -localauth flag is included.

See Also

"BosConfig(5)" on page 409, "KeyFile(5)" on page 423, "UserList(5)" on page 437, "bos(8)" on page 661, "bos_getrestart(8)" on page 695, "bos_restart(8)" on page 723, "bos_setrestart(8)" on page 739, "upclient(8)" on page 885

Copyright

bosserver

Name

`bosserver` — Initializes the BOS Server

Synopsis

bosserver [-noauth] [-log] [-enable_peer_stats] [-enable_process_stats] [-help]

Description

The bosserver command initializes the Basic OverSeer (BOS) Server (bosserver process). In the conventional configuration, the binary file is located in the */usr/afs/bin* directory on a file server machine.

The BOS Server must run on every file server machine and helps to automate file server administration by performing the following tasks:

- Monitors the other AFS server processes on the local machine, to make sure they are running correctly.
- Automatically restarts 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.
- Processes commands from the bos suite that administrators issue to verify the status of server processes, install and start new processes, stop processes either temporarily or permanently, and restart halted processes.
- Manages system configuration information: the files that list the cell's server encryption keys, database server machines, and users privileged to issue commands from the bos and vos suites.

The BOS Server logs a default set of important events in the file */usr/afs/logs/BosLog*. To record the name of any user who performs a privileged bos command (one that requires being listed in the */usr/afs/etc/UserList* file), add the -log flag. To display the contents of the BosLog file, use the bos getlog command.

The first time that the BOS Server initializes on a server machine, it creates several files and subdirectories in the local */usr/afs* directory, and sets their mode bits to protect them from unauthorized access. Each time it restarts, it checks that the mode bits still comply with the settings listed in the following chart. A question mark indicates that the BOS Server initially turns off the bit (sets it to the hyphen), but does not check it at restart.

If the mode bits do not comply, the BOS Server writes the following warning to the *BosLog* file:

```
Bosserver reports inappropriate access on server directories
```

However, the BOS Server does not reset the mode bits, so the administrator can set them to alternate values if desired (with the understanding that the warning message then appears at startup).

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Options

-noauth

Assigns the unprivileged identity `anonymous` to the issuer, which is useful only when authorization checking is disabled on the server machine (for instance, during the installation of a file server machine.)

-log

Records in the */usr/afs/logs/BosLog* file the names of all users who successfully issue a privileged bos command (one that requires being listed in the */usr/afs/etc/UserList* file).

-enable_peer_stats

Activates the collection of Rx statistics and allocates memory for their storage. For each connection with a specific UDP port on another machine, a separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received. To display or otherwise access the records, use the Rx Monitoring API.

-enable_process_stats

Activates the collection of Rx statistics and allocates memory for their storage. A separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received, aggregated over all connections to other machines. To display or otherwise access the records, use the Rx Monitoring API.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command initializes the BOS Server and logs the names of users who issue privileged bos commands.

```
% bosserver -log &
```

Privilege Required

The issuer most be logged onto a file server machine as the local superuser `root`.

See Also

"BosConfig(5)" on page 409, "BosLog(5)" on page 413, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_exec(8)" on page 687, "bos_getlog(8)" on page 691, "bos_getrestart(8)" on page 695, "bos_restart(8)" on page 723, "bos_shutdown(8)" on page 743, "bos_start(8)" on page 747, "bos_startup(8)" on page 749, "bos_status(8)" on page 751, "bos_stop(8)" on page 755

Copyright

buserver

Name

buserver — Initializes the Backup Server

Synopsis

buserver [-database <*database directory*>] [-cellservdb <*cell configuration directory*>] [-resetdb] [-noauth] [-smallht] [-servers <*list of ubik database servers*>+] [-enable_peer_stats] [-enable_process_stats] [-help]

Description

The buserver command initializes the Backup Server, which runs on database server machines and maintains the Backup Database. In the conventional configuration, the binary file is located in the */usr/afs/bin* directory on a file server machine.

The buserver command is not normally issued at the command shell prompt, but rather placed into a database server machine's */usr/afs/local/BosConfig* file with the bos create command. If it is ever issued at the command shell prompt, the issuer must be logged onto a file server machine as the local superuser root.

As it initializes, the Backup Server process creates the two files that constitute the Backup Database, *bdb.DB0* and *bdb.DBSYS1*, in the */usr/afs/db* directory if they do not already exist. The Backup Database houses information about volume sets and entries, the dump hierarchy, Tape Coordinators, and previously performed dump sets. Use the commands in the backup suite to administer the database.

The Backup Server records a trace of its activity in the */usr/afs/logs/BackupLog* file. Use the bos getlog command to display the contents of the file.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Cautions

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.

Options

-database <*database directory*>

Specifies the pathname of an alternate directory for the Backup Database files, ending in a final slash (/). If this argument is not provided, the default is the */usr/afs/db* directory.

-cellservdb <*cell configuration directory*>

Specifies the pathname of the directory from which the Backup Server reads in an alternate version of the `CellServDB` file. This argument is mandatory for correct functioning when the Backup Server is running on a subset of the cell's database server machines that is not a majority of the machines listed in the standard `/usr/afs/etc/CellServDB` file (which the Backup Server consults if this argument is not provided). It is not appropriate in any other circumstances.

-resetdb

Removes all of the information in the Backup Database files in the `/usr/afs/db` directory, leaving zero-length versions of them. The backup operator must recreate the configuration entries in the database (for volume sets, the dump hierarchy and so on) before performing backup operations.

-noauth

Establishes an unauthenticated connection between the issuer and the Backup Server, in which the Backup Server treats the issuer as the unprivileged user `anonymous`. 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 refuses to perform such an action even if the -noauth flag is used.

-smallht

Directs the Backup Server to use smaller internal hash tables for the Backup Database, which reduces memory requirements but can make data access take longer.

-servers <*list of ubik database servers*>+

Specifies the database server machines on which to start the Backup Server. Use this argument if running 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.

-enable_peer_stats

Activates the collection of Rx statistics and allocates memory for their storage. For each connection with a specific UDP port on another machine, a separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received. To display or otherwise access the records, use the Rx Monitoring API.

-enable_process_stats

Activates the collection of Rx statistics and allocates memory for their storage. A separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received, aggregated over all connections to other machines. To display or otherwise access the records, use the Rx Monitoring API.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example bos create command creates a `buserver` process on the file server machine `fs3.abc.com`. It appears here on two lines only for legibility.

```
% bos create -server fs3.abc.com -instance buserver \
             -type simple -cmd /usr/afs/bin/buserver
```

Privilege Required

The issuer must be logged in as the superuser `root` on a file server machine to issue the command at a command shell prompt. It is conventional instead to create and start the process by issuing the bos create command.

See Also

"BackupLog(5)" on page 407, "BosConfig(5)" on page 409, "CellServDB(5)" on page 415, "bdb.DB0(5)" on page 451, "backup(8)" on page 531, "bos_create(8)" on page 677, "bos_getlog(8)" on page 691

Copyright

butc

Name

butc — Initializes the Tape Coordinator process

Synopsis

butc [-port <*port offset*>] [-debuglevel (0 | 1 | 2)] [-cell <*cell name*>] [-noautoquery] [-localauth] [-help]

butc [-p <*port offset*>] [-d (0 | 1 | 2)] [-c <*cell name*>] [-n] [-l] [-h]

Description

The butc command initializes a Tape Coordinator process on a Tape Coordinator machine, enabling an operator to direct Backup System requests to the associated tape device or backup data file. (The Tape Coordinator controls a backup data file if the `FILE YES` instruction appears in the */usr/afs/backup/CFG_device_name* file that corresponds to the Tape Coordinator's entry in the */usr/afs/backup/tapeconfig* file. For the sake of simplicity, the following discusses tape devices only.)

It is conventional to start and run the Tape Coordinator in the foreground. In this case, it runs on its own connection, which is unavailable for any other use and must remain open the entire time the Tape Coordinator is to accept backup requests and while it is executing them. (When using a window manager, the connection corresponds to a separate command shell window.) The Tape Coordinator can run in the background if the *CFG_device_name* file is configured to eliminate any need for the Tape Coordinator to prompt the operator. In both the foreground and background, the Tape Coordinator writes operation traces and other output to the standard output stream on the connection over which it was started. Use the -debuglevel argument to control the amount of information that appears. The Tape Coordinator also writes traces and error messages to two files in the local */usr/afs/backup* directory:

- The *TE_device_name* file records problems that the Tape Coordinator encounters as it executes backup operations.
- The *TL_device_name* file records a trace of operations as well as the same errors written to the *TE_device_name* file.

The Tape Coordinator creates the files automatically as it initializes. If there are existing files, the Tape Coordinator renames them with a `.old` extension, overwriting the existing `.old` files if they exist. It derives the *device_name* part of the file names by stripping off the device name's */dev/* prefix and replacing any other slashes with underscores. For example, the files are called *TE_rmt_4m* and *TL_rmt_4m* for a device called */dev/rmt/4m*.

By default, at the beginning of each operation the Tape Coordinator prompts for the operator to insert the first tape into the drive and press Return. To suppress this prompt, include the -noautoquery flag on the command line or the instruction `AUTOQUERY NO` in the */usr/afs/backup/CFG_device_name* file. When the prompt is suppressed, the first required tape must be in the drive before a backup command is issued. For subsequent tapes, the Tape

Coordinator uses its normal tape acquisition routine: if the */usr/afs/backup/CFG_device_name* file includes a `MOUNT` instruction, the Tape Coordinator invokes the indicated command; otherwise, it prompts the operator for the next tape.

To stop the Tape Coordinator process, enter an interrupt signal such as Ctrl-C over the dedicated connection (in the command shell window).

To cancel a backup operation that involves a tape before it begins (assuming the initial tape prompt has not been suppressed), enter the letter `a` (for `abort`) and press Return at the Tape Coordinator's prompt for the first tape.

Tape Coordinator operation depends on the correct configuration of certain files, as described in the following list:

- The local */usr/afs/backup/tapeconfig* file must include an entry for the Tape Coordinator that specifies its device name and port offset number, among other information; for details, "tapeconfig(5)" on page 493.
- The port offset number recorded in the Tape Coordinator's entry in the Backup Database must match the one in the *tapeconfig* file. Create the Backup Database entry by using the backup addhost command.
- The optional */usr/afs/backup/CFG_device_name* file can contain instructions for mounting and unmounting tapes automatically (when using a tape stacker or jukebox, for instance) or automating other aspects of the backup process. The *device_name* part of the name is derived as described previously for the *TE_device_name* and *TL_device_name* files.

Cautions

If the Tape Coordinator machine is an AIX machine, use the SMIT utility to set the device's block size to 0 (zero), indicating variable block size. Otherwise, tape devices attached to machines running other operating systems sometimes cannot read tapes written on AIX machines. For instructions, see the *IBM AFS Administration Guide* chapter about configuring the Backup System.

Options

-port <*port offset*>

Specifies the port offset number of the Tape Coordinator to initialize.

-debuglevel

Controls the amount and type of messages the Tape Coordinator displays on the standard output stream. Provide one of three acceptable values:

- `0` to display the minimum level of detail required to describe Tape Coordinator operations, including prompts for tapes, messages that indicate the beginning and end of operations, and error messages. This is the default value.

- `1` to display the names of the volumes being dumped or restored as well as the information displayed at level `0`.
- `2` to display all messages also being written to the *`TL_device_name`* log file.

-cell <*cell name*>

Names the cell in which the Tape Coordinator operates (the cell to which the file server machines that house affected volumes belong). If this argument is omitted, the Tape Coordinator runs in the local cell as defined in the local */usr/vice/etc/ThisCell* file. Do not combine this flag with the -localauth argument.

-noautoquery

Suppresses the Tape Coordinator's prompt for insertion of the first tape needed for an operation. The operator must insert the tape into the drive before issuing the backup command that initializes the operation.

-localauth

Constructs a server ticket using the server encryption key with the highest key version number in the local */usr/afs/etc/KeyFile*. The butc command interpreter presents the ticket, which never expires, to the Volume Server and Volume Location Server to use in mutual authentication.

Do not combine this argument with the -cell flag, and use it only when logged on to a server machine as the local superuser `root`; client machines do not have */usr/afs/etc/KeyFile* file.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command starts the Tape Coordinator with port offset `7` at debug level `1`, meaning the Tape Coordinator reports the names of volumes it is dumping or restoring.

```
% butc -port 7 -debuglevel 1
```

Privilege Required

The issuer must be listed in the */usr/afs/etc/UserList* file on every machine where the Backup Server or Volume Location (VL) Server is running, and on every file server machine that houses a volume to be backed up. If the -localauth flag is included, the issuer must instead be logged on to the Tape Coordinator machine as the local superuser `root`. In addition, the issuer must be able to read and write to the log and configuration files in the local */usr/afs/backup* directory.

See Also

"KeyFile(5)" on page 423, "ThisCell(5)" on page 435, "UserList(5)" on page 437, "butc(5)" on page 453, "butc_logs(5)" on page 465, "tapeconfig(5)" on page 493, "backup_addhost(8)" on page 539

Copyright

fileserver

Name

`fileserver` — Initializes the File Server component of the fs process

Synopsis

fileserver [-d <*debug level*>] [-p <*number of processes*>] [-spare <*number of spare blocks*>] [-pctspare <*percentage spare*>] [-b <*buffers*>] [-l <*large vnodes*>] [-s <*small nodes*>] [-vc <*volume cachesize*>] [-w <*call back wait interval*>] [-cb <*number of call backs*>] [-banner] [-novbc] [-implicit <*admin mode bits: rlidwka*>] [-hr <*number of hours between refreshing the host cps*>] [-busyat <*redirect clients when queue > n*>] [-rxpck <*number of rx extra packets*>] [-rxdbg] [-rxdbge] [-m <*min percentage spare in partition*>] [-lock] [-L] [-S] [-k <*stack size*>] [-realm <*Kerberos realm name*>] [-udpsize <*size of socket buffer in bytes*>] [-enable_peer_stats] [-enable_process_stats] [-help]

Description

The fileserver command initializes the File Server component of the `fs` process. In the conventional configuration, its binary file is located in the */usr/afs/bin* directory on a file server machine.

The fileserver command is not normally issued at the command shell prompt, but rather placed into a database server machine's */usr/afs/local/BosConfig* file with the bos create command. If it is ever issued at the command shell prompt, the issuer must be logged onto a file server machine as the local superuser `root`.

The File Server creates the */usr/afs/logs/FileLog* log file as it initializes, if the file does not already exist. It does not write a detailed trace by default, but use the -d option to increase the amount of detail. Use the bos getlog command to display the contents of the log file.

The command's arguments enable the administrator to control many aspects of the File Server's performance, as detailed in "OPTIONS" on page . By default the fileserver command sets values for many arguments that are suitable for a medium-sized file server machine. To set values suitable for a small or large file server machine, use the -S or -L flag respectively. The following list describes the parameters and corresponding argument for which the fileserver command sets default values, and the table below summarizes the setting for each of the three machine sizes.

- The maximum number of lightweight processes (LWPs) the File Server uses to handle requests for data; corresponds to the -p argument. The File Server always uses a minimum of 32 KB for these processes.
- The maximum number of directory blocks the File Server caches in memory; corresponds to the -b argument. Each cached directory block (buffer) consumes 2,092 bytes of memory.
- The maximum number of large vnodes the File Server caches in memory for tracking directory elements; corresponds to the -l argument. Each large vnode consumes 292 bytes of memory.

- The maximum number of small vnodes the File Server caches in memory for tracking file elements; corresponds to the -s argument. Each small vnode consumes 100 bytes of memory.
- The maximum volume cache size, which determines how many volumes the File Server can cache in memory before having to retrieve data from disk; corresponds to the -vc argument.
- The maximum number of callback structures the File Server caches in memory; corresponds to the -cb argument. Each callback structure consumes 16 bytes of memory.
- The maximum number of Rx packets the File Server uses; corresponds to the -rxpck argument. Each packet consumes 1544 bytes of memory.

The default values are:

Parameter (Argument)	Small (-S)	Medium	Large (-L)
Number of LWPs (-p)	6	9	12
Number of cached dir blocks (-b)	70	90	120
Number of cached large vnodes (-l)	200	400	600
Number of cached small vnodes (-s)	200	400	600
Maximum volume cache size (-vc)	200	400	600
Number of callbacks (-cb)	20,000	60,000	64,000
Number of Rx packets (-rxpck)	100	150	200

To override any of the values, provide the indicated argument (which can be combined with the -S or -L flag).

The amount of memory required for the File Server varies. The approximate default memory usage is 751 KB when the -S flag is used (small configuration), 1.1 MB when all defaults are used (medium configuration), and 1.4 MB when the -L flag is used (large configuration). If additional memory is available, increasing the value of the -cb and -vc arguments can improve File Server performance most directly.

By default, the File Server allows a volume to exceed its quota by 1 MB when an application is writing data to an existing file in a volume that is full. The File Server still does not allow users to create new files in a full volume. To change the default, use one of the following arguments:

- Set the -spare argument to the number of extra kilobytes that the File Server allows as overage. A value of `0` allows no overage.
- Set the -pctspare argument to the percentage of the volume's quota the File Server allows as overage.

By default, the File Server implicitly grants the `a` (administer) and `l` (lookup) permissions to system:administrators on the access control list (ACL) of every directory in the volumes stored on its file server machine. In other words, the group's members can exercise those two permissions even when an entry for the group does not appear on an ACL. To change the set of default permissions, use the -implicit argument.

The File Server maintains a *host current protection subgroup* (*host CPS*) for each client machine from which it has received a data access request. Like the CPS for a user, a host CPS lists all of the Protection Database groups to which the machine belongs, and the File Server compares the host CPS to a directory's ACL to determine in what manner users on the machine are authorized to access the directory's contents. When the pts adduser or pts removeuser

command is used to change the groups to which a machine belongs, the File Server must recompute the machine's host CPS in order to notice the change. By default, the File Server contacts the Protection Server every two hours to recompute host CPSs, implying that it can take that long for changed group memberships to become effective. To change this frequency, use the -hr argument.

The File Server generates the following message when a partition is nearly full:

```
No space left on device
```

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Cautions

Do not use the -k and -w arguments, which are intended for use by the AFS Development group only. Changing them from their default values can result in unpredictable File Server behavior. In any case, on many operating systems the File Server uses native threads rather than the LWP threads, so using the -k argument to set the number of LWP threads has no effect.

Do not specify both the -spare and -pctspare arguments. Doing so causes the File Server to exit, leaving an error message in the `/usr/afs/logs/FileLog` file.

Options that are available only on some system types, such as the -m and -lock options, appear in the output generated by the -help option only on the relevant system type.

Options

-d <*debug level*>

Sets the detail level for the debugging trace written to the `/usr/afs/logs/FileLog` file. Provide one of the following values, each of which produces an increasingly detailed trace: `0`, `1`, `5`, `25`, and `125`. The default value of `0` produces only a few messages.

-p <*number of processes*>

Sets the number of threads to run. Provide a positive integer. The File Server creates and uses five threads for special purposes, in addition to the number specified (but if this argument specifies the maximum possible number, the File Server automatically uses five of the threads for its own purposes).

The maximum number of threads can differ in each release of AFS. Consult the *IBM AFS Release Notes* for the current release.

-spare <*number of spare blocks*>

Specifies the number of additional kilobytes an application can store in a volume after the 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 the -pctspare argument.

-pctspare <*percentage spare*>

Specifies the amount by which the File Server allows 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 the -spare argument.

-b <*buffers*>

Sets the number of directory buffers. Provide a positive integer.

-l <*large vnodes*>

Sets the number of large vnodes available in memory for caching directory elements. Provide a positive integer.

-s <*small nodes*>

Sets the number of small vnodes available in memory for caching file elements. Provide a positive integer.

-vc <*volume cachesize*>

Sets the number of volumes the File Server can cache in memory. Provide a positive integer.

-w <*call back wait interval*>

Sets the interval at which the daemon spawned by the File Server performs its maintenance tasks. Do not use this argument; changing the default value can cause unpredictable behavior.

-cb <*number of callbacks*>

Sets the number of callbacks the File Server can track. Provide a positive integer.

-banner

Prints the following banner to */dev/console* about every 10 minutes.

```
File Server is running at I&lt;time&gt;.
```

-novbc

Prevents the File Server from breaking the callbacks that Cache Managers hold on a volume that the File Server is reattaching after the volume was offline (as a result of the vos restore command, for example). Use of this flag is strongly discouraged.

-implicit <*admin mode bits*>

Defines the set of permissions granted by default to the system:administrators group on the ACL of every directory in a volume stored on the file server machine. Provide one or more of the standard permission letters (`rlidwka`) and auxiliary permission letters (`ABCDEFGH`), or one of the shorthand notations for groups of permissions (`all`, `none`, `read`, and `write`). To review the meaning of the permissions, see the fs setacl reference page.

-hr <*number of hours between refreshing the host cps*>

Specifies how often the File Server refreshes its knowledge of the machines that belong to protection groups (refreshes the host CPSs for machines). 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.

-busyat <*redirect clients when queue > n*>

Defines the number of incoming RPCs that can be waiting for a response from the File Server before the File Server returns the error code `VBUSY` to the Cache Manager that sent the latest RPC. In response, the Cache Manager retransmits the RPC after a delay. This argument prevents the accumulation of so many waiting RPCs that the File Server can never process them all. Provide a positive integer. The default value is `600`.

-rxpck <*number of rx extra packets*>

Controls the number of Rx packets the File Server uses to store data for incoming RPCs that it is currently handling, that are waiting for a response, and for replies that are not yet complete. Provide a positive integer.

-rxdbg

Writes a trace of the File Server's operations on Rx packets to the file `/usr/afs/logs/rx_dbg`.

-rxdbge

Writes a trace of the File Server's operations on Rx events (such as retransmissions) to the file `/usr/afs/logs/rx_dbg`.

-m <*min percentage spare in partition*>

Specifies the percentage of each AFS server partition that the AIX version of the File Server creates as a reserve. Specify an integer value between `0` and `30`; the default is 8%. A value of `0` means that the partition can become completely full, which can have serious negative consequences.

-lock

Prevents any portion of the fileserver binary from being paged (swapped) out of memory on a file server machine running the IRIX operating system.

-L

Sets values for many arguments in a manner suitable for a large file server machine. Combine this flag with any option except the -S flag; omit both flags to set values suitable for a medium-sized file server machine.

-S

Sets values for many arguments in a manner suitable for a small file server machine. Combine this flag with any option except the -L flag; omit both flags to set values suitable for a medium-sized file server machine.

-k <*stack size*>

Sets the LWP stack size in units of 1 kilobyte. Do not use this argument, and in particular do not specify a value less than the default of 24.

-realm <*Kerberos realm name*>

Defines the Kerberos realm name for the File Server to use. If this argument is not provided, it uses the realm name corresponding to the cell listed in the local */usr/afs/etc/ThisCell* file.

-udpsize <*size of socket buffer in bytes*>

Sets the size of the UDP buffer, which is 64 KB by default. Provide a positive integer, preferably larger than the default.

-enable_peer_stats

Activates the collection of Rx statistics and allocates memory for their storage. For each connection with a specific UDP port on another machine, a separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received. To display or otherwise access the records, use the Rx Monitoring API.

-enable_process_stats

Activates the collection of Rx statistics and allocates memory for their storage. A separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received, aggregated over all connections to other machines. To display or otherwise access the records, use the Rx Monitoring API.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following bos create command creates an fs process on the file server machine fs2.abc.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 -server fs2.abc.com -instance fs -type fs \
             -cmd "/usr/afs/bin/fileserver -pctspare 10 \
             -L" /usr/afs/bin/volserver /usr/afs/bin/salvager
```

Privilege Required

The issuer must be logged in as the superuser root on a file server machine to issue the command at a command shell prompt. It is conventional instead to create and start the process by issuing the bos create command.

See Also

"BosConfig(5)" on page 409, "FileLog(5)" on page 421, "bos_create(8)" on page 677, "bos_getlog(8)" on page 691, "fs_setacl(1)" on page 117, "salvager(8)" on page 879, "volserver(8)" on page 925

Copyright

fms

Name

`fms` — Determine a tape's capacity and a tape device's filemark size

Synopsis

fms -tape <*tape special file*> [-help]

fms -t <*tape special file*> [-h]

Description

The fms command determines the capacity of the tape currently in the tape device identified by the -tape argument, along with the size of the filemark for the device. The filemark is also referred to as the device's end-of-file (EOF) marker, and can differ for each combination of tape and tape device.

As the Tape Coordinator writes a dump, it writes a filemark between the data included from each volume and also tracks the amount of space left before the end of the tape (EOT). For some tape devices, the filemark is large enough (multiple megabytes) that failure to consider it leads the Tape Coordinator significantly to overestimate the available space.

The intended use of this command is to determine tape capacity and filemark size values that can be specified in a tape device's entry in the `/usr/afs/backup/tapeconfig` file. For certain types of tape drives, the Tape Coordinator operates more efficiently when the `tapeconfig` file lists accurate values. For further discussion, see the *IBM AFS Administration Guide* chapter on configuring the Backup System.

Insert a tape in the drive before issuing this command.

Cautions

Do not use this command on compressing tape devices in compression mode or with tape devices that handle tapes of multigigabyte (or multiterabyte) capacity. It does not produce accurate results in those cases. For alternate suggestions on the values to record in the tapeconfig file for compressing drives, see the *IBM AFS Administration Guide* chapter on configuring the Backup System.

Running the command completely overwrites the tape, so use a blank one or one that can be recycled.

Because it writes filemarks to the complete length of the tape, the command can take from several hours to more than a day to complete.

Options

-tape <*tape special file*>

Specifies the UNIX device name of the tape device for which to determine filemark size and the capacity of the tape it currently contains. The format varies on different system types, but usually begins with */dev*; an example is */dev/sd0a*.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The command generates output both on the standard output stream and in the *fms.log* file that it creates in the current working directory. The output reports the capacity of the tape in the device and the device's filemark size.

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. The last two lines of both screen and file output provide the following information:

- `Tape capacity is` *number* `bytes`: specifies the size, in bytes, of the tape in the device.
- `File marks are` *number* `bytes`: specifies the device's filemark size in bytes.

The following message indicates that the fms command interpreter cannot access the tape device. The command halts.

```
Can't open tape drive I&lt;device&gt;
```

The following message indicates that the command interpreter cannot create the *fms.log* log file. Again, the command halts.

```
Can't open log file
```

Examples

The following command illustrates the output for the device called */dev/rmt1h*:

```
% fms /dev/rmt1h
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
```

The following appears in 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

The issuer must be able to insert and write to files in the currently working directory, if the *fms.log* file does not already exist. If it already exists, the issuer need only be able to write to it.

See Also

"fms.log(5)" on page 469, "tapeconfig(5)" on page 493

Copyright

fstrace

Name

`fstrace` — Introduction to the fstrace command suite

Description

The commands in the fstrace command suite are the interface that system administrators employ to trace Cache Manager operations for debugging purposes. Examples of Cache Manager operations are fetching file data or the status information used to produce output for the UNIX ls command.

The fstrace command interpreter defines an extensive set of Cache Manager operations as the `cm` *event set*. When the event set is activated, the Cache Manager writes a message to the `cmfx` *trace log* in kernel memory each time it performs one of the defined operations. The log expands only to a defined size (by default, 60 KB), after which it is overwritten in a circular fashion (new trace messages overwrite the oldest ones). If an operation of particular interest occurs, the administrator can afterward display the log on the standard output stream or write it to a file for later study. For more specific procedural instructions, see the *IBM AFS Administration Guide*.

There are several categories of commands in the fstrace command suite:

- Commands to administer or display information about the trace log: fstrace clear, fstrace lslog, fstrace setlog.
- Commands to set or display the status of the event set: fstrace lsset and fstrace setset.
- A command to display the contents of the trace log: fstrace dump.
- Commands to obtain help: fstrace apropos and fstrace help.

Options

All fstrace commands accept the following optional flag. It is listed in the command descriptions and described in detail here:

-help

Prints a command's online help message on the standard output stream. Do not combine this flag with any of the command's other options; when it is provided, the command interpreter ignores all other options, and only prints the help message.

Privilege Required

To issue most fstrace commands, the issuer must be logged on as the local superuser `root` on the machine that is generating the trace log.

See Also

"fstrace_apropos(8)" on page 785, "fstrace_clear(8)" on page 787, "fstrace_dump(8)" on page 789, "fstrace_help(8)" on page 793, "fstrace_lslog(8)" on page 795, "fstrace_lsset(8)" on page 797, "fstrace_setlog(8)" on page 799, "fstrace_setset(8)" on page 801

Copyright

fstrace apropos

Name

`fstrace apropos` — Displays each help entry containing a keyword string

Synopsis

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

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

Description

The fstrace apropos command displays the first line of the online help entry for any fstrace command that contains in its name or short description the string specified with the -topic argument.

To display a command's complete syntax, use the fstrace help command.

Options

-topic <*help string*>

Specifies the keyword string to match, in lowercase letters only. If the string is more than a single word, surround it with double quotes ("") or other delimiters.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of a command's online help entry names it and briefly describes its function. This command displays the first line for any fstrace command where the string specified with the -topic argument is part of the command name or first line.

Examples

The following command lists all fstrace commands that include the word `set` in their names or short descriptions:

```
% fstrace apropos set
clear: clear logs by logname or by event set
lsset: list available event sets
setlog: set the size of a log
setset: set state of event sets
```

Privilege Required

None

See Also

"fstrace(8)" on page 783, "fstrace_help(8)" on page 793

Copyright

fstrace clear

Name

`fstrace clear` — Clears the trace log

Synopsis

fstrace clear [-set <*set name*>+] [-log <*log name*>+] [-help]

fstrace c [-s <*set name*>+] [-l <*log name*>+] [-h]

Description

The fstrace clear command erases the contents of the trace log from kernel memory, but leaves kernel memory allocated for the log.

Options

-set <*set name*>+

Names the event set for which to clear the associated trace log. The only acceptable value is `cm` (for which the associated trace log is `cmfx`). Provide either this argument or the -log argument, or omit both to clear the `cmfx` log by default.

-log <*log name*>+

Names the trace log to clear. The only acceptable value is `cmfx`. Provide either this argument or the -set argument, or omit both to clear the `cmfx` log by default.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command clears the `cmfx` trace log on the local machine:

```
# fstrace clear
```

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"fstrace(8)" on page 783, "fstrace_lslog(8)" on page 795, "fstrace_lsset(8)" on page 797

Copyright

fstrace dump

Name

`fstrace dump` — Dumps a trace log

Synopsis

fstrace dump [-set <*set name*>+] [-follow <*log name*>] [-file <*output filename*>] [-sleep <*seconds between reads*>] [-help]

fstrace d [-se <*set name*>+] [-fo <*log name*>] [-fi <*output filename*>] [-sl <*seconds between reads*>] [-h]

Description

The fstrace dump command displays the current contents of the `cmfx` trace log on the standard output stream or writes it to the file named by the -file argument.

To write the log continuously to the standard output stream or to a file, use the -follow argument. By default, the log's contents are written out every ten seconds and then automatically cleared. To change the interval between writes, use the -sleep argument.

Cautions

This command produces output only if the `cm` event set is active. To display or set the event set's state, use the fstrace lsset or fstrace setset command respectively.

To make the output from this command maximally readable, the message catalog file called *`afszcm.cat`* must reside in the local */usr/vice/etc/C* directory. If necessary, copy the file to that directory from the AFS Binary Distribution before activating tracing.

When the `cm` event set is active, a defined amount of kernel memory (by default, 60 KB) is allocated for the `cmfx` trace log. As described in "fstrace(8)" on page 783, when the buffer is full, messages are overwritten in a circular fashion (new messages overwrite the oldest ones). To allocate more kernel memory for the log, use the fstrace setlog command; to display the log buffer's current size, use the fstrace lslog command with the -long argument.

Options

-set <*set name*>+

Names the event set for which to write out the associated trace log. The only acceptable value is `cm` (for which the associated trace log is `cmfx`). Provide either this argument or the -log argument, or omit both to write out the `cmfx` log by default.

-follow <*log name*>

Names the trace log to write out continuously at a specified interval (by default, every ten seconds; use the -sleep argument to change the interval). The log is cleared after each write operation.

The only acceptable value is `cmfx`. Provide either this argument or the -set argument, or omit both to write out the `cmfx` log by default.

-file <*output filename*>

Specifies the pathname of the file to which to write the trace log's contents. It can be in AFS or on the local disk. Partial pathnames are interpreted relative to the current working directory. If this argument is omitted, the trace log appears on the standard output stream.

-sleep <*seconds between reads*>

Sets the number of seconds between writes of the trace log's contents when it is dumped continuously. Provide the -follow argument along with this one. If this argument is omitted, the default interval is ten seconds.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output begins with a header specifying the date and time at which the write operation began. If the -follow argument is not included, the header also reports the number of logs being dumped; it is always `1`, since there is only the `cmfx` trace log. The format of the header is as follows:

```
AFS Trace Dump -
  Date: I&lt;starting_timestamp&gt;
Found 1 logs.
Contents of log cmfx:
```

Each subsequent message describes a Cache Manager operation in the following format:

```
time &lt;timestamp&gt;, pid &lt;pid&gt;:&lt;event_message&gt;
```

where

<timestamp>

Specifies the time at which the Cache Manager performed the operation, as the number of seconds since the dump began.

<pid>

Specifies the process ID of the process or thread associated with the message.

<event_message>

Is the message itself. They are generally meaningful only to someone familiar with the AFS source code.

In addition, every 1024 seconds the fstrace command interpreter writes a message that records the current clock time, in the following format:

```
time &lt;timestamp&gt;, pid &lt;pid&gt;: Current time: &lt;unix_time&gt;
```

where

<timestamp>

Is the number of seconds from the start of trace logging.

<pid>

Is the process ID number.

<unix_time>

Is the machine's clock time, represent in the standard UNIX time format as the number of seconds since midnight on January 1, 1970.

Use this message to determine the actual clock time associated with each log message. Determine the actual time as follows:

- Locate the message of interest.
- Search backward through the trace file for the closest current time message.
- If the current time message's timestamp is smaller than the log message's timestamp, subtract former from the latter. If the current time message's timestamp is larger than the log message's timestamp, add 1024 to the latter and subtract the former from the result.
- Add the resulting number to the current time message's <unix_time> to determine the log message's actual time.

  If any of the data in the kernel trace buffer has been overwritten since tracing was activated, the following message appears at the appropriate place in the output:

  ```
  Log wrapped; data missing.
  ```

  To reduce the likelihood of overwriting, use the fstrace setlog command to increase the kernel buffer's size. To display the current defined buffer size, use the fstrace lslog command with the -long argument.

  The following message at the end of the log dump indicates that it is completed:

  ```
  AFS Trace Dump - Completed
  ```

Examples

The following command dumps the log associated with the cm event set to the standard output stream.

```
# fstrace dump -set cm
AFS Trace Dump -
   Date: Tue Apr  7 10:54:57 1998
Found 1 logs.
time 32.965783, pid 0: Tue Apr  7 10:45:52 1998
time 32.965783, pid 33657: Close 0x5c39ed8 flags 0x20
time 32.965897, pid 33657: Gn_close vp 0x5c39ed8 flags 0x20 (returns 0x0)
time 35.159854, pid 10891: Breaking callback for 5bd95e4 states 1024 (volume 0)
time 35.407081, pid 10891: Breaking callback for 5c0fadc states 1024 (volume 0)
                                .
                                .
                                .
time 71.440456, pid 33658: Lookup adp 0x5bbdcf0 name g3oCKs \
     fid (756 4fb7e:588d240.2ff978a8.6)
time 71.440569, pid 33658: Returning code 2 from 19
time 71.440619, pid 33658: Gn_lookup vp 0x5bbdcf0 name g3oCKs (returns 0x2)
time 71.464989, pid 38267: Gn_open vp 0x5bbd000 flags 0x0 (returns 0x0)
AFS Trace Dump - Completed
```

The following command dumps the trace log associated with the cm event set on the local machine to the file `cmfx.dump.file.1`, using the default interval of 10 seconds between successive dumps:

```
# fstrace dump -follow cmfx -file cmfx.dump.file.1
```

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"afszcm.cat(5)" on page 449, "fstrace(8)" on page 783, "fstrace_lslog(8)" on page 795, "fstrace_setlog(8)" on page 799, "fstrace_lsset(8)" on page 797

Copyright

fstrace help

Name

`fstrace help` — Displays help for fstrace commands

Synopsis

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

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

Description

The fstrace help command displays the complete online help entry (short description and syntax statement) for each command operation code specified by the -topic argument. If the -topic argument is omitted, the output includes the first line (name and short description) of the online help entry for every fstrace command.

To list every fstrace command whose name or short description includes a specified keyword, use the fstrace apropos command.

Options

-topic <*help string*>+

Indicates each command for which to display the complete online help entry. Omit the fstrace part of the command name, providing only the operation code (for example, specify clear, not fstrace clear). If this argument is omitted, the output briefly describes every fstrace command.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The first line names the command and briefly describes its function.
- The second line lists aliases for the command, if any.
- The final line, which begins with the string `Usage`, lists the command's options in the prescribed order. Online help entries use the same symbols (for example, brackets) as the reference pages in this document.

Examples

The following command displays the online help entry for the fstrace setset command:

```
% fstrace help -topic setset
fstrace setset: set state of event sets
Usage: fstrace setset [-set &lt;set_name&gt;+] [-active] [-inactive]
[-dormant] [-help]
```

Privilege Required

None

See Also

"fstrace(8)" on page 783, "fstrace_apropos(8)" on page 785

Copyright

fstrace lslog

Name

`fstrace lslog` — Displays information about a log

Synopsis

fstrace lslog [-set <*set name*>+] [-log <*log name*>] [-long] [-help]

fstrace lsl [-s <*set name*>+] [-log <*log name*>] [-lon] [-h]

Description

The fstrace lslog command reports whether the `cmfx` log is available for use. If the -long argument is included, the output reports the log's defined size, and whether that amount of space is currently allocated in kernel memory or not.

To change the `cmfx` trace log's size, use the fstrace setlog command. To display or set whether space is allocated for it in kernel memory, use the fstrace lsset or fstrace setset command to display or set the state of the corresponding `cm` event set, respectively.

Options

-set <*set name*>+

Names the event set for which to display information about the corresponding trace log. The only acceptable value is `cm` (for which the associated trace log is `cmfx`). Provide either this argument or the -log argument, or omit both to display information about the `cmfx` log by default.

-log <*log name*>

Names the trace log about which to report. The only acceptable value is `cmfx`. Provide either this argument or the -set argument, or omit both to report on the `cmfx` log by default.

-long

Reports the defined size of the log in kilobyte units and whether that amount of space is currently allocated in kernel memory.

-help

Prints the online help for this command. All other valid options are ignored.

Output

By default, the fstrace lslog command displays only the name of the available log, `cmfx`, in the following format:

```
Available logs:
cmfx
```

When the -long flag is included, the output also reports the defined size of the log in kilobytes, and whether or not that amount of space is currently allocated in kernel memory, in the following format:

```
Available logs:
cmfx : &lt;log_size&gt; kbytes (allocated | unallocated)
```

The `allocated` state indicates that the indicated number of kilobytes is reserved for the `cmfx` trace log in kernel memory. The `cm` event set's state is either `active` or `inactive`, as reported by the fstrace lsset command, and set by the fstrace setset command's -active or -inactive flags respectively.

The `unallocated` state indicates that no kernel memory is currently reserved for the `cmfx` trace log. The cm event set's state is `dormant`, as reported by the fstrace lsset command and set by the fstrace setset command's -dormant flag. If the event set's state is later changed to active or inactive, the number of kilobytes indicated as <log_size> are again allocated in kernel memory.

Examples

The following example uses the -long flag to display information about the `cmfx` log:

```
# fstrace lslog -log cmfx -long
Available logs:
cmfx : 60 kbytes (allocated)
```

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"fstrace(8)" on page 783, "fstrace_lsset(8)" on page 797, "fstrace_setlog(8)" on page 799

Copyright

fstrace lsset

Name

`fstrace lsset` — Reports the status of an event set

Synopsis

fstrace lsset [-set <*set name*>+] [-help]

fstrace lss [-s <*set name*>+] [-h]

Description

The fstrace lsset command displays a list of the available event sets and reports their current status (active, inactive, or dormant).

To change an event set's status, use the fstrace setset command.

Options

-set <*set name*>+

Names the event set for which to display the status. The only acceptable value is `cm`, which is also the default if this argument is omitted.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output lists the available event sets and the status of each, in the following format:

```
Available sets:
cm (active | inactive | dormant)
```

where

active

Indicates that tracing is enabled for the event set, and kernel memory allocated for the corresponding trace log.

inactive

Indicates that tracing is temporarily disabled for the event set, but kernel memory still allocated for the corresponding trace log.

dormant

Indicates that tracing is disabled for the event set, and no kernel memory allocated for the corresponding trace log.

Examples

The following example displays the available event set and its status:

```
# fstrace lsset
Available sets:
cm active
```

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"fstrace(8)" on page 783, "fstrace_setset(8)" on page 801

Copyright

fstrace setlog

Name

`fstrace setlog` — Sets the size of a trace log

Synopsis

fstrace setlog [-log <*log name*>+] -buffersize <*1-kilobyte units*> [-help]

fstrace setl [-l <*log name*>+] -b <*1-kilobyte units*> [-h]

fstrace sl [-l <*log name*>+] -b <*1-kilobyte units*> [-h]

Description

The fstrace setlog command defines the number of kilobytes of kernel memory allocated for the `cmfx` trace log. If kernel memory is currently allocated, the command clears the current log and creates a new log buffer of the specified size.

To display the current defined size of the log buffer, issue the fstrace lslog command with the -long argument. To control whether the indicated amount of space is actually allocated, use the fstrace setset command to set the status of the `cm` event set; to display the event set's status, use the fstrace lsset command.

Options

-log <*log name*>+

Names trace log for which to set the size. The only acceptable value is `cmfx`, which is also the default if this argument is omitted.

-buffersize <*1-kilobyte units*>

Specifies the number of 1-kilobyte blocks of kernel memory to allocate for the trace log.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command allocated 80 KB of kernel memory for the `cmfx` trace log:

```
# fstrace setlog -log cmfx -buffersize 80
```

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"fstrace(8)" on page 783, "fstrace_lslog(8)" on page 795, "fstrace_lsset(8)" on page 797, "fstrace_setset(8)" on page 801

Copyright

fstrace setset

Name

`fstrace setset` — Sets the status of an event set

Synopsis

fstrace setset [-set <*set name*>+] [-active] [-inactive] [-dormant] [-help]

fs set [-s <*set name*>+] [-a] [-i] [-d] [-h]

Description

The fstrace setset command sets the status of the `cm` kernel event set on the local machine, which determines whether trace messages are recorded in the log buffer in kernel memory.

Options

-set <*set name*>+

Names the event set for which to set the status. The only acceptable value `cm`, which is also the default if this argument is omitted.

-active

Enables tracing for the event set and allocates kernel memory for the associated trace log buffer. Provide one of this flag, the -inactive flag, or the -dormant flag.

-inactive

Temporarily disables tracing for the event set, but does not change the allocation of kernel memory for the associated trace log buffer. Provide one of this flag, the -active flag, or the -dormant flag.

-dormant

Disables tracing for the event set and frees the kernel memory previously allocated for the associated trace log buffer. Provide one of this flag, the -active flag, or the -inactive flag.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example sets the cm event set's status to inactive:

```
# fstrace setset -set cm -inactive
```

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"fstrace(8)" on page 783, "fstrace_lsset(8)" on page 797, "fstrace_setlog(8)" on page 799

Copyright

kadb_check

Name

kadb_check — Checks the integrity of the Authentication Database

Synopsis

kadb_check -database <*kadb file*> [-uheader] [-kheader] [-entries] [-verbose] [-rebuild <*out file*>] [-help]

kadb_check -d <*kadb file*> [-u] [-k] [-e] [-v] [-r <*out file*>] [-h]

Description

The kadb_check command checks the integrity of the Protection Database, reporting any errors or corruption it finds. If there are problems, do not issue any kas commands until the database is repaired.

Cautions

The results can be unpredictable if the Authentication Server makes changes to the Authentication Database while this command is running. Use the bos shutdown command to shutdown the local kaserver process before running this command, or before creating a second copy of the *kaserver.DB0* file (with a different name) on which to run the command.

Options

-database <*kadb file*>

Names the Authentication Database (copy of the *kaserver.DB0* file) to check. If the current working directory is not the location of the file, provide a pathname, either full or relative to the current working directory.

-uheader

Displays information which Ubik maintains in the database's header.

-kheader

Displays information which the Authentication Server maintains in the database's header.

-entries

Outputs every entry in the database, providing information similar to that returned by the kas examine command.

-verbose

Reports additional information about the database, including the number of free (allocated but unused) entries in the database.

-rebuild <*out file*>

Names the file in which to record a list of kas commands which, if issued in the command shell, recreate the current state of the database being verified. Partial pathnames are interpreted relative to the current working directory.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If there are errors in the database, the output always reports them on the standard error stream. If any options other than -database or -help are provided, the output written to the standard output stream includes additional information as described for each option in "OPTIONS" on page . The output is intended for debugging purposes and is meaningful to someone familiar with the internal structure of the Authentication Database.

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"kaserver.DB0(5)" on page 471, "bos_shutdown(8)" on page 743, "kas_examine(8)" on page 817, "kaserver(8)" on page 859

Copyright

kas

Name

kas — Introduction to the kas command suite

Description

The commands in the kas command suite are the administrative interface to the Authentication Server, which runs on each database server machine in a cell, maintains the Authentication Database, and provides the authentication tickets that client applications must present to AFS servers in order to obtain access to AFS data and other services.

There are several categories of commands in the kas command suite:

- Commands to create, modify, examine and delete entries in the Authentication Database, including passwords: kas create, kas delete, kas examine, kas list, kas setfields, kas setkey, kas setpassword, and kas unlock.
- Commands to create, delete, and examine tokens and server tickets: kas forgetticket, kas listtickets, kas noauthentication, and kas stringtokey.
- A command to enter interactive mode: kas interactive.
- A command to trace Authentication Server operations: kas statistics.
- Commands to obtain help: kas apropos and kas help.

Because of the sensitivity of information in the Authentication Database, the Authentication Server authenticates issuers of kas commands directly, rather than accepting the standard token generated by the Ticket Granting Service. Any kas command that requires administrative privilege prompts the issuer for a password. The resulting ticket is valid for six hours unless the maximum ticket lifetime for the issuer or the Authentication Server's Ticket Granting Service is shorter.

To avoid having to provide a password repeatedly when issuing a sequence of kas commands, enter *interactive mode* by issuing the kas interactive command, typing kas without any operation code, or typing kas followed by a user and cell name, separated by an at-sign (@; an example is `kas smith.admin@abc.com`). After prompting once for a password, the Authentication Server accepts the resulting token for every command issued during the interactive session. See "kas_interactive(8)" on page 827 for a discussion of when to use each method for entering interactive mode and of the effects of entering a session.

The Authentication Server maintains two databases on the local disk of the machine where it runs:

- The Authentication Database (*/usr/afs/db/kaserver.DB0*) stores the information used to provide AFS authentication services to users and servers, including the password scrambled as an encryption key. The reference page for the kas examine command describes the information in a database entry.
- An auxiliary file (*/usr/afs/local/kaauxdb* by default) that tracks how often the user has provided an incorrect password to the local Authentication Server. The reference page for the

kas setfields command describes how the Authentication Server uses this file to enforce the limit on consecutive authentication failures. To designate an alternate directory for the file, use the kaserver command's -localfiles argument.

Options

The following arguments and flags are available on many commands in the kas suite. (Some of them are unavailable on commands entered in interactive mode, because the information they specify is established when entering interactive mode and cannot be changed except by leaving interactive mode.) The reference page for each command also lists them, but they are described here in greater detail.

-admin_username <*user name*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. If this argument is omitted, the kas command interpreter requests authentication for the identity under which the issuer is logged onto the local machine. Do not combine this argument with the -noauth flag.

-cell <*cell name*>

Names the cell in which to run the command. It is acceptable to abbreviate the cell name to the shortest form that distinguishes it from the other entries in the `/usr/vice/etc/CellServDB` file on the local machine. If the -cell argument is omitted, the command interpreter determines the name of the local cell by reading the following in order:

- The value of the AFSCELL environment variable.
- The local `/usr/vice/etc/ThisCell` file.

The -cell argument is not available on commands issued in interactive mode. The cell defined when the kas command interpreter enters interactive mode applies to all commands issued during the interactive session.

-help

Prints a command's online help message on the standard output stream. Do not combine this flag with any of the command's other options; when it is provided, the command interpreter ignores all other options, and only prints the help message.

-noauth

Establishes an unauthenticated connection to the Authentication Server, in which the Authentication Server treats the issuer as the unprivileged user anonymous. It is useful only when authorization checking is disabled on the server machine (during the installation of a server machine or when the bos setauth command has been used during other unusual circumstances). In normal circumstances, the Authentication Server allows only privileged users to issue most kas commands, and refuses to perform such an action even if the -noauth flag is provided. Do not combine this flag with the -admin_username and -password_for_admin arguments.

-password_for_admin <*password*>

Specifies the password of the command's issuer. It is best to omit this argument, which echoes the password visibly in the command shell, instead enter the password at the prompt. Do not combine this argument with the -noauth flag.

-servers <*machine name*>+

Establishes a connection with the Authentication Server running on each specified database server machine, instead of on each machine listed in the local `/usr/vice/etc/CellServDB` file. In either case, the kas command interpreter then chooses one of the machines at random to contact for execution of each subsequent command. The issuer can abbreviate the machine name to the shortest form that allows the local name service to identify it uniquely.

Privilege Required

To issue most kas commands, the issuer must have the `ADMIN` flag set in his or her Authentication Database entry (use the kas setfields command to turn the flag on).

See Also

"CellServDB(5)" on page 415, "kaserver.DB0(5)" on page 471, "kaserverauxdb(5)" on page 473, "kas_apropos(8)" on page 809, "kas_create(8)" on page 811, "kas_delete(8)" on page 815, "kas_examine(8)" on page 817, "kas_forgetticket(8)" on page 823, "kas_help(8)" on page 825, "kas_interactive(8)" on page 827, "kas_list(8)" on page 831, "kas_listtickets(8)" on page 835, "kas_noauthentication(8)" on page 837, "kas_quit(8)" on page 839, "kas_setfields(8)" on page 841, "kas_setpassword(8)" on page 847, "kas_statistics(8)" on page 851, "kas_stringtokey(8)" on page 855, "kas_unlock(8)" on page 857, "kaserver(8)" on page 859

Copyright

kas apropos

Name

`kas apropos` — Displays each help entry containing a keyword string

Synopsis

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

kas a -t <*help string*> [-h]

Description

The kas apropos command displays the first line of the online help entry for any kas command that has the string specified by the -topic argument in its name or short description.

To display the syntax for a command, use the kas help command.

Options

-topic <*help string*>

Specifies the keyword string to match, in lowercase letters only. If the string is more than a single word, surround it with double quotes ("") or other delimiters.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of a command's online help entry names it and briefly describes its function. This command displays the first line for any kas command where the string specified with the -topic argument is part of the command name or first line.

Examples

The following command lists all kas commands that include the word `key` in their names or short descriptions:

```
% kas apropos key
setkey: set a user's key
stringtokey: convert a string to a key
```

Privilege Required

None, and no password is required.

See Also

"kas(8)" on page 805, "kas_help(8)" on page 825

Copyright

kas create

Name

`kas create` — Creates an entry in the Authentication Database

Synopsis

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 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

The kas create command creates an entry in the Authentication Database for the user named by the -name argument.

To avoid having the account's initial password echo visibly at the shell prompt, omit the -initial_password argument; the command interpreter prompts for the password and does not echo it visibly. Whether or not -initial_password is omitted, the Authentication Server converts the password into a form suitable for use as an encryption key, and records it in the entry's key field.

To alter settings in an Authentication Database entry, use the kas setfields command. To examine an entry, use the kas examine command. To list every entry in the database, use the kas list command.

Options

-name <*name of user*>

Names the new Authentication Database entry. Because it is the name under which the user logs in, it must obey the restrictions that many operating systems impose on user names (usually, to contain no more than eight lowercase letters).

-initial_password <*initial password*>

Sets the user's password; provide a character string that can include uppercase and lowercase letters, numerals and punctuation. The Authentication Server scrambles the string into an octal string suitable for use as an encryption key before placing it in the entry's key field. If this argument is omitted, the command interpreter prompts for the string and does not echo it visibly.

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example shows the prompts that appear when an administrator logged in as admin creates an Authentication Database entry for the user smith, and does not include either the -initial_password or -password_for_admin arguments.

```
% kas create smith
Password for admin:
initial_password:
Verifying, please re-enter initial_password:
```

Privilege Required

The issuer must have the ADMIN flag set on his or her Authentication Database entry.

See Also

"kas(8)" on page 805, "kas_examine(8)" on page 817, "kas_list(8)" on page 831, "kas_setfields(8)" on page 841

Copyright

kas delete

Name

`kas delete` — Deletes an entry from the Authentication Database

Synopsis

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]

kas d -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

The kas delete command removes from the Authentication Database the user entry named by the -name argument. The indicated user becomes unable to log in, or the indicated server becomes unreachable (because the Authentication Server's Ticket Granting Service module no longer has a key with which to seal tickets for the server).

Options

-name <*name of user*>

Names the Authentication Database entry to delete.

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>+

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example shows the administrative user admin entering interactive mode to delete three accounts.

```
% kas
Password for admin:
ka&gt; delete smith
ka&gt; delete pat
ka&gt; delete terry
```

Privilege Required

The issuer must have the ADMIN flag set on his or her Authentication Database entry.

See Also

"kas(8)" on page 805, "kas_create(8)" on page 811

Copyright

kas examine

Name

`kas examine` — Displays information from an Authentication Database entry

Synopsis

kas examine -name <*name of user*> [-showkey] [-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 e -na <*name of user*> [-sh] [-a <*admin principal to use for authentication*>] [-p <*admin password*>] [-c <*cell name*>] [-se <*explicit list of authentication servers*>+] [-no] [-h]

Description

The kas examine command formats and displays information from the Authentication Database entry of the user named by the -name argument.

To alter the settings displayed with this command, issue the kas setfields command.

Cautions

Displaying actual keys on the standard output stream by including the -showkey flag constitutes a security exposure. For most purposes, it is sufficient to display a checksum.

Options

-name <*name of user*>

Names the Authentication Database entry from which to display information.

-showkey

Displays the octal digits that constitute the key. The issuer must have the `ADMIN` flag on his or her Authentication Database entry.

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>+

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity `anonymous` to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output includes:

- The entry name, following the string `User data for`.
- One or more status flags in parentheses; they appear only if an administrator has used the kas setfields command to change them from their default values. A plus sign (+) separates the flags if there is more than one. The nondefault values that can appear, and their meanings, are as follows:

  ADMIN

  Enables the user to issue privileged kas commands (default is `NOADMIN`).

  NOTGS

  Prevents the user from obtaining tickets from the Authentication Server's Ticket Granting Service (default is `TGS`).

  NOSEAL

  Prevents the Ticket Granting Service from using the entry's key field as an encryption key (default is `SEAL`).

  NOCPW

  Prevents the user from changing his or her password (default is `CPW`).

- The key version number, in parentheses, following the word `key`, then one of the following.
  - A checksum equivalent of the key, following the string `cksum is`, if the -showkey flag is not included. The checksum is a decimal number derived by encrypting a constant with the key. In the case of the `afs` entry, this number must match the checksum with the corresponding key version number in the output of the bos listkeys command; if not,

follow the instructions in the *IBM AFS Administration Guide* for creating a new server encryption key.

- The actual key, following a colon, if the -showkey flag is included. The key consists of eight octal numbers, each represented as a backslash followed by three decimal digits.

- The date the user last changed his or her own password, following the string `last cpw` (which stands for "last change of password").
- The string `password will never expire` indicates that the associated password never expires; the string `password will expire` is followed by the password's expiration date. After the indicated date, the user cannot authenticate, but has 30 days after it in which to use the kpasswd or kas setpassword command to set a new password. After 30 days, only an administrator (one whose account is marked with the `ADMIN` flag) can change the password by using the kas setpassword command. To set the password expiration date, use the kas setfields command's -pwexpires argument.
- The number of times the user can fail to provide the correct password before the account locks, followed by the string `consecutive unsuccessful authentications are permitted`, or the string `An unlimited number of unsuccessful authentications is permitted` to indicate that there is no limit. To set the limit, use the kas setfields command's -attempts argument. To unlock a locked account, use the kas unlock command. The kas setfields reference page discusses how the implementation of the lockout feature interacts with this setting.
- The number of minutes for which the Authentication Server refuses the user's login attempts after the limit on consecutive unsuccessful authentication attempts is exceeded, following the string `The lock time for this user is`. Use the kas command's -locktime argument to set the lockout time. This line appears only if a limit on the number of unsuccessful authentication attempts has been set with the the kas setfields command's -attempts argument.
- An indication of whether the Authentication Server is currently refusing the user's login attempts. The string `User is not locked` indicates that authentication can succeed, whereas the string `User is locked until` *time* indicates that the user cannot authenticate until the indicated time. Use the kas unlock command to enable a user to attempt authentication. This line appears only if a limit on the number of unsuccessful authentication attempts has been set with the kas setfields command's -attempts argument.
- The date on which the Authentication Server entry expires, or the string `entry never expires` to indicate that the entry does not expire. A user becomes unable to authenticate when his or her entry expires. Use the kas setfields command's -expiration argument to set the expiration date.
- The maximum possible lifetime of the tokens that the Authentication Server grants the user. This value interacts with several others to determine the actual lifetime of the token, as described in "klog(1)" on page 153. Use the kas setfields command's -lifetime argument to set this value.
- The date on which the entry was last modified, following the string `last mod on` and the user name of the administrator who modified it. The date on which a user changed his or her own password is recorded on the second line of output as `last cpw` instead.

- An indication of whether the user can reuse one of his or her last twenty passwords when issuing the kpasswd, kas setpassword, or kas setkey commands. Use the kas setfields command's -reuse argument to set this restriction.

Examples

The following example command shows the user smith displaying her own Authentication Database entry. Note the `ADMIN` flag, which shows that `smith` is privileged.

```
% kas examine smith
Password for smith:
User data for smith (ADMIN)
 key (0) cksum is 3414844392,  last cpw: Thu Mar 25 16:05:44 1999
 password will expire:  Fri Apr 30 20:44:36 1999
 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 Tue Jan 5 08:22:29 1999 by admin
 permit password reuse
```

In the following example, the user `pat` examines his Authentication Database entry to determine when the account lockout currently in effect will end.

```
% kas examine pat
Password for pat:
User data for pat
 key (0) cksum is 73829292912,  last cpw: Wed Apr 7 11:23:01 1999
 password will expire:  Fri  Jun 11 11:23:01 1999
 5 consecutive unsuccessful authentications are permitted.
 The lock time for this user is 25.5 minutes.
 User is locked until Tue Sep 21 12:25:07 1999
 entry expires on never. Max ticket lifetime 100.00 hours.
 last mod on Thu Feb 4 08:22:29 1999 by admin
 permit password reuse
```

In the following example, an administrator logged in as `admin` uses the -showkey flag to display the octal digits that constitute the key in the `afs` entry.

```
% kas examine -name afs -showkey
Password for admin: I&lt;admin_password&gt;
User data for afs
 key (12): \357\253\304\352\234\236\253\352, last cpw: no date
 entry never expires. Max ticket lifetime 100.00 hours.
 last mod on Thu Mar 25 14:53:29 1999 by admin
 permit password reuse
```

Privilege Required

A user can examine his or her own entry. To examine others' entries or to include the -showkey flag, the issuer must have the `ADMIN` flag set in his or her Authentication Database entry.

See Also

"bos_addkey(8)" on page 669, "bos_listkeys(8)" on page 707, "bos_setauth(8)" on page 733, "kas(8)" on page 805, "kas_setfields(8)" on page 841, "kas_setpassword(8)" on page 847, "kas_unlock(8)" on page 857, "klog(1)" on page 153, "kpasswd(1)" on page 163

Copyright

kas forgetticket

Name

`kas forgetticket` — Discards all tickets for the issuer

Synopsis

kas forgetticket [-all] [-help]

kas f [-a] [-h]

Description

The kas forgetticket command discards all of the issuer's tickets stored in the local machine's kernel memory. This includes the AFS server ticket from each cell in which the user has authenticated, and any tickets that the user have acquired during the current kas session (either when entering the session or by using the kas getticket command).

Options

-all

Discards all tickets. This argument explicitly invokes the command's default behavior.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command discards all of the issuer's tickets.

```
% kas forgetticket
```

Privilege Required

None, and no password is required.

See Also

"kas(8)" on page 805

Copyright

kas help

Name

`kas help` — Displays help for kas commands

Synopsis

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

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

Description

The kas help command displays the complete online help entry (short description and syntax statement) for each command operation code specified by the -topic argument. If the -topic argument is omitted, the output includes the first line (name and short description) of the online help entry for every kas command.

To list every kas command whose name or short description includes a specified keyword, use the kas apropos command.

Options

-topic <*help string*>+

Indicates each command for which to display the complete online help entry. Omit the kas part of the command name, providing only the operation code (for example, specify setpassword, not kas setpassword). If this argument is omitted, the output briefly describes every kas command.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The first line names the command and briefly describes its function.
- The second line lists aliases for the command, if any.
- The final line, which begins with the string `Usage`, lists the command's options in the prescribed order. Online help entries use the same symbols (for example, brackets) as the reference pages in this document.

Examples

The following command 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 &lt;name of user&gt;
[-new_password &lt;new password&gt;] [-kvno &lt;key version number&gt;]
[-admin_username &lt;admin principal to use for authentication&gt;]
[-password_for_admin &lt;password&gt;] [-cell &lt;cell name&gt;]
[-servers &lt;explicit list of authentication servers&gt;+] [-help]
```

Privilege Required

None, and no password is required.

See Also

"kas(8)" on page 805, "kas_apropos(8)" on page 809

Copyright

kas interactive

Name

`kas interactive` — Enters interactive mode

Synopsis

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 i [-a <*admin principal to use for authentication*>] [-p <*admin password*>] [-c <*cell name*>] [-s <*explicit list of authentication servers*>+] [-n] [-h]

Description

The kas interactive command establishes an interactive session for the issuer of the command. By default, the command interpreter establishes an authenticated connection for the user logged into the local file system with all of the Authentication Servers listed in the local */usr/vice/etc/CellServDB* file for the cell named in the local */usr/vice/etc/ThisCell* file. To specify an alternate identity, cell name, or list of Authentication Servers, include the -admin_username, -cell, or -servers arguments respectively. Interactive mode lasts for six hours unless the maximum ticket lifetime for the issuer or the Authentication Server's Ticket Granting Service is shorter.

There are two other ways to enter interactive mode, in addition to the kas interactive command:

- Type the kas command at the shell prompt without any operation code. If appropriate, include one or more of the -admin_username, -password_for_admin, -cell, and -servers arguments.
- Type the kas command followed by a user name and cell name, separated by an `@` sign (for example: kas admin@abc.com), to establish a connection under the specified identity with the Authentication Servers listed in the local */usr/vice/etc/CellServDB* file for the indicated cell. If appropriate, provide the -servers argument to specify an alternate list of Authentication Server machines that belong to the indicated cell.

There are several consequences of entering interactive mode:

- The `ka>` prompt replaces the system (shell) prompt. When typing commands at this prompt, provide only the operation code (omit the command suite name, kas).
- The command interpreter does not prompt for the issuer's password.

  The issuer's identity and password, the relevant cell, and the set of Authentication Server machines specified when entering interactive mode apply to all commands issued during the session. They cannot be changed without leaving the session, except by using the kas noauthentication command to replace the current authenticated connections with

unauthenticated ones. The -admin_username, -password_for_admin, -cell, and -servers arguments are ignored if provided on a command issued during interactive mode.

To establish an unauthenticated connection to the Authentication Server, include the -noauth flag or provide an incorrect password. Unless authorization checking is disabled on each Authentication Server machine involved, however, it is not possible to perform any privileged operations within such a session.

To end the current authenticated connection and establish an unauthenticated one, issue the kas noauthentication command. To leave interactive mode and return to the regular shell prompt, issue the kas quit command.

Options

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>+

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example shows a user entering interactive mode as the privileged user admin.

```
% kas interactive admin
Password for admin: I&lt;admin_password&gt;
ka&gt;
```

Privilege Required

None

See Also

"kas(8)" on page 805, "kas_noauthentication(8)" on page 837, "kas_quit(8)" on page 839

Copyright

kas list

Name

`kas list` — Displays all entries in the Authentication Database

Synopsis

kas list [-long] [-showadmin] [-showkey] [-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 ls [-l] [-showa] [-showk] [-a <*admin principal to use for authentication*>] [-p <*admin password*>] [-c <*cell name*>] [-se <*explicit list of authentication servers*>+] [-n] [-h]

Description

The kas list command either displays all entries from the Authentication Database by name, or displays the full database entry for a defined set of entries, as determined by the flag provided:

- To display every entry in the Authentication Database in full, include the -long flag.
- To display only those entries in full that have the `ADMIN` flag set, include the -showadmin flag.
- To list only the name of each Authentication Database entry, omit both the -long and -showadmin flags.

By default, full entries include a checksum for the encryption key, rather than the actual octal digits that constitute the key. To display the octal digits, include the -showkey flag with the -long or -showadmin flag.

Options

-long

Displays every Authentication Database entry in full. Provide this flag or the -showadmin flag, or omit both to display just the name of every database entry.

-showadmin

Displays in full only the Authentication Database entries that have the `ADMIN` flag set. Provide this flag or the -long flag, or omit both to display just the name of every database entry.

-showkey

Displays the octal digits that constitute the key in each full entry. Provide either the -long or -showadmin flag along with this one.

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>+

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If neither the -long or -showadmin flag is provided, the output lists the name of each entry in the Authentication Database on its own line.

If the -long flag is included, the output includes every Authentication Database entry in full. If the -showadmin flag is included, the output includes in full only the Authentication Database entries that have the `ADMIN` flag set. If the -showkey is provided along with either one, the output includes the octal digits that constitute the encryption key in each entry.

A full Authentication Database entry includes the same information displayed by the kas examine command; for details, see that command's reference page.

Privilege Required

The issuer must have the `ADMIN` flag set on his or her Authentication Database entry.

See Also

"kas(8)" on page 805, "kas_examine(8)" on page 817

Copyright

kas listtickets

Name

`kas listtickets` — Displays all of the issuer's tickets (tokens)

Synopsis

kas listtickets [-name <*name of server*>] [-long] [-help]

kas listt [-n <*name of server*>] [-l] [-h]

Description

The kas listtickets command displays the associated user ID (AFS UID), cell name, and expiration date of some or all of the issuer's tickets (tokens), depending on which options are provided:

- To display all tokens, provide neither the -name argument nor -long flag. The output is similar to that of the tokens command.
- To display a single token, provide the -name argument to specify name of the Authentication Database entry for the entity that accepts the token. All AFS server processes accept tokens sealed with the key from the `afs` entry.
- To display in addition the octal numbers that constitute the token and session key, provide the -long flag.

Options

-name <*name of server*>

Names the Authentication Database entry of the entity (usually a server process) that accepts the token to display.

-long

Displays the octal numbers that constitute the session key and ticket.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output reports the AFS UID of the user who owns the token, the service (usually, `afs`) and cell for which it is valid, and its expiration date, using the following format. If the message does not specify a cell, the ticket is for the local cell.

```
User’s (AFS ID &lt;AFS UID&gt;) tokens for &lt;service&gt;[@&lt;cellname&gt;] \
    [Expires &lt;date&gt;]
```

If the -long flag is provided, the output also includes the octal numbers making up the session key and token, along with the key version number and the number of bytes in the token (if the number of bytes is not 56, there is an error).

If the marker `[>> POSTDATED <]` appears instead of an expiration date, 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 `abc.com` cell and AFS UID 35 in the `test.abc.com` cell. He is working on a machine in the first cell and is authenticated in both cells.

```
% kas listtickets
User’s (AFS ID 1020) tokens for afs [Expires Wed Mar 31 9:30:54 1999]
User’s (AFS ID 35@test.abc.com) tokens for afs@test.abc.com \
          [Expires Wed Mar 31 13:54:26 1999]


% kas listtickets -name afs -long
User’s (AFS ID 1020) tokens for afs [Expires Wed Mar 31 9:30:54 1999]
SessionKey: \375\205\351\227\032\310\263\013
Ticket: (kvno = 0, len = 56): \033\005\221\156\203\278\312\058\016\133...
```

Privilege Required

None, and no password is required.

See Also

"kas(8)" on page 805, "tokens(1)" on page 269

Copyright

kas noauthentication

Name

`kas noauthentication` — Discards an authenticated identity in interactive mode

Synopsis

noauthentication [-help]

n [-h]

Description

The kas noauthentication command closes the (presumably authenticated) connection that the issuer established with one or more Authentication Server processes when entering interactive mode. It opens a new unauthenticated connection to each server, assigning the issuer the unprivileged identity anonymous. It does not actually discard the user's tokens from the Cache Manager's memory (as the unlog or kas forgetticket command does). Unless authorization checking is disabled on each Authentication Server machine, it becomes impossible to perform any privileged operations within the session established by this command.

This command is operative only during interactive mode, so omit the kas command suite name from the command line.

Options

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example command discards the authentication information with which the user entered interactive mode.

```
ka&gt; noauthentication
```

Privilege Required

None, and no password is required.

See Also

"kas(8)" on page 805, "kas_forgetticket(8)" on page 823, "kas_interactive(8)" on page 827, "unlog(1)" on page 283

Copyright

kas quit

Name

`kas quit` — Leaves interactive mode

Synopsis

quit [-help]

q [-h]

Description

The kas quit command ends interactive mode, severing the authenticated connection to one or more Authentication Server processes and returning the issuer to the normal shell prompt.

This command is operative only during interactive mode, so omit the kas command suite name from the command line.

Options

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example demonstrates how the normal command shell prompt returns when the issuer leaves interactive mode.

```
ka&gt; quit
%
```

Privilege Required

None, and no password is required.

See Also

"kas(8)" on page 805, "kas_interactive(8)" on page 827

Copyright

kas setfields

Name

`kas setfields` — Sets fields in an Authentication Database entry

Synopsis

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]

kas setf -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]

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

The kas setfields command changes the Authentication Database entry for the user named by the -name argument in the manner specified by the various optional arguments, which can occur singly or in combination:

- To set the flags that determine whether the user has administrative privileges to the Authentication Server, can obtain a ticket, can change his or her password, and so on, include the -flags argument.
- To set when the Authentication Database entry expires, include the -expiration argument.
- To set the maximum ticket lifetime associated with the entry, include the -lifetime argument. "klog(1)" on page 153 explains how this value interacts with others to determine the actual lifetime of a token.
- To set when the user's password expires, include the -pwexpires argument.
- To set whether the user can reuse any of the previous twenty passwords when creating a new one, include the -reuse argument.
- To set the maximum number of times the user can provide an incorrect password before the Authentication Server refuses to accept any more attempts (locks the issuer out), include the -attempts argument. After the sixth failed authentication attempt, the Authentication Server

logs a message in the UNIX system log file (the *syslog* file or equivalent, for which the standard location varies depending on the operating system).

- To set how long the Authentication Server refuses to process authentication attempts for a locked-out user, set the -locktime argument.

The kas examine command displays the settings made with this command.

Cautions

The password lifetime set with the -pwexpires argument begins at the time the user's password was last changed, rather than when this command is issued. It can therefore be retroactive. If, for example, a user changed her password 100 days ago and the password lifetime is set to 100 days or less, the password effectively expires immediately. To avoid retroactive expiration, instruct the user to change the password just before setting a password lifetime.

Administrators whose authentication accounts have the ADMIN flag enjoy complete access to the sensitive information in the Authentication Database. To prevent access by unauthorized users, use the -attempts argument to impose a fairly strict limit on the number of times that a user obtaining administrative tokens can provide an incorrect password. Note, however, that there must be more than one account in the cell with the ADMIN flag. The kas unlock command requires the ADMIN privilege, so it is important that the locked-out administrator (or a colleague) can access another ADMIN-privileged account to unlock the current account.

In certain circumstances, the mechanism used to enforce the number of failed authentication attempts can cause a lockout even though the number of failed attempts is less than the limit set by the -attempts argument. Client-side authentication programs such as klog and an AFS-modified login utility normally choose an Authentication Server at random for each authentication attempt, and in case of a failure are likely to choose a different Authentication Server for the next attempt. The Authentication Servers running on the various database server machines do not communicate with each other about how many times a user has failed to provide the correct password to them. Instead, each Authentication Server maintains its own separate copy of the auxiliary database file *kaserverauxdb* (located in the */usr/afs/local* directory by default), which records the number of consecutive authentication failures for each user account and the time of the most recent failure. This implementation means that on average each Authentication Server knows about only a fraction of the total number of failed attempts. The only way to avoid allowing more than the number of attempts set by the -attempts argument is to have each Authentication Server allow only some fraction of the total. More specifically, if the limit on failed attempts is *f*, and the number of Authentication Servers is *S*, then each Authentication Server can only permit a number of attempts equal to *f* divided by *S* (the Ubik synchronization site for the Authentication Server tracks any remainder, *f* mod *S*).

Normally, this implementation does not reduce the number of allowed attempts to less than the configured limit (*f*). If one Authentication Server refuses an attempt, the client contacts another instance of the server, continuing until either it successfully authenticates or has contacted all of the servers. However, if one or more of the Authentication Server processes is unavailable, the limit is effectively reduced by a percentage equal to the quantity *U* divided by *S*, where *U* is the number of unavailable servers and *S* is the number normally available.

To avoid the undesirable consequences of setting a limit on failed authentication attempts, note the following recommendations:

- Do not set the -attempts argument (the limit on failed authentication attempts) too low. A limit of nine failed attempts is recommended for regular user accounts, to allow three failed attempts per Authentication Server in a cell with three database server machines.
- Set fairly short lockout times when including the -locktime argument. Although guessing passwords is a common method of attack, it is not a very sophisticated one. Setting a lockout time can help discourage attackers, but excessively long times are likely to be more of a burden to authorized users than to potential attackers. A lockout time of 25 minutes is recommended for regular user accounts.
- Do not assign an infinite lockout time on an account (by setting the -locktime argument to `0` [zero]) unless there is a highly compelling reason. Such accounts almost inevitably become locked at some point, because each Authentication Server never resets the account's failure counter in its copy of the *kaauxdb* file (in contrast, when the lockout time is not infinite, the counter resets after the specified amount of time has passed since the last failed attempt to that Authentication Server). Furthermore, the only way to unlock an account with an infinite lockout time is for an administrator to issue the kas unlock command. It is especially dangerous to set an infinite lockout time on an administrative account; if all administrative accounts become locked, the only way to unlock them is to shut down all instances of the Authentication Server and remove the *kaauxdb* file on each.

Options

-name <*name of user*>

Names the Authentication Database account for which to change settings.

-flags <*hex flag or flag name expression*>

Sets one or more of four toggling flags, adding them to any flags currently set. Either specify one or more of the following strings, or specify a hexidecimal number that combines the indicated values. To return all four flags to their defaults, provide a value of `0` (zero). To set more than one flag at once using the strings, connect them with plus signs (example: `NOTGS+ADMIN+CPW`). To remove all the current flag settings before setting new ones, precede the list with an equal sign (example: `=NOTGS+ADMIN+CPW`).

ADMIN

The user is allowed to issue privileged kas commands (hexadecimal equivalent is `0x004`, default is `NOADMIN`).

NOTGS

The Authentication Server's Ticket Granting Service (TGS) refuses to issue tickets to the user (hexadecimal equivalent is `0x008`, default is `TGS`).

NOSEAL

The Ticket Granting Service cannot use the contents of this entry's key field as an encryption key (hexadecimal equivalent is `0x020`, default is `SEAL`).

NOCPW

The user cannot change his or her own password or key (hexadecimal equivalent is `0x040`, default is `CPW`).

-expiration <*date of account expiration*>

Determines when the entry itself expires. When a user entry expires, the user becomes unable to log in; when a server entry such as `afs` expires, all server processes that use the associated key become inaccessible. Provide one of the three acceptable values:

never

The account never expires (the default).

mm/dd/yyyy

Sets the expiration date to 12:00 a.m. on the indicated date (month / day / year). Examples: `01/23/1999`, `10/07/2000`.

"*mm/dd/yyyy hh:MM*"

Sets the expiration date to the indicated time (hours:minutes) on the indicated date (month / day / year). Specify the time 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: `"01/23/1999 22:30"`, `"10/07/2000 3:45"`.

Acceptable values for the year range from `1970` (1 January 1970 is time 0 in the standard UNIX date representation) through `2037` (2037 is the maximum because the UNIX representation cannot accommodate dates later than a value in February 2038).

-lifetime <*maximum ticket lifetime*>

Specifies the maximum lifetime that the Authentication Server's Ticket Granting Service (TGS) can assign to a ticket. If the account belongs to a user, this value is the maximum lifetime of a token issued to the user. If the account corresponds to a server such as `afs`, this value is the maximum lifetime of a ticket that the TGS issues to clients for presentation to the server during mutual authentication.

Specify an integer that represents a number of seconds (3600 equals one hour), or include a colon in the number to indicate a number of hours and minutes (`10:00` equals 10 hours). If this argument is omitted, the default setting is 100:00 hours (360000 seconds).

-pwexpires <*number of days password is valid*>

Sets the number of days after the user's password was last changed that it remains valid. Provide an integer from the range `1` through `254` to specify the number of days until expiration, or the value `0` to indicate that the password never expires (the default).

When the password expires, the user is unable to authenticate, but has 30 days after the expiration date in which to use the kpasswd command to change the password (after that, only an administrator can change it by using the kas setpassword command). Note that the clock starts at the time the password was last changed, not when the kas setfields command is issued. To avoid retroactive expiration, have the user change the password just before issuing a command that includes this argument.

-reuse (yes | no)

Specifies whether or not the user can reuse any of his or her last 20 passwords. The acceptable values are `yes` to allow reuse of old passwords (the default) and `no` to prohibit reuse of a password that is similar to one of the previous 20 passwords.

-attempts <*maximum successive failed login tries*>

Sets the number of consecutive times the user can provide an incorrect password during authentication (using the klog command or a login utility that grants AFS tokens). When the user exceeds the limit, the Authentication Server rejects further attempts (locks the user out) for the amount of time specified by the -locktime argument. Provide an integer from the range `1` through `254` to specify the number of failures allowed, or `0` to indicate that there is no limit on authentication attempts (the default value).

-locktime <*failure penalty*>

Specifies how long the Authentication Server refuses authentication attempts from a user who has exceeded the failure limit set by the -attempts argument.

Specify a number of hours and minutes (*hh:mm*) or minutes only (*mm*), from the range `01` (one minute) through `36:00` (36 hours). The kas command interpreter automatically reduces any larger value to `36:00` and also rounds up any non-zero value to the next higher multiple of 8.5 minutes. A value of `0` (zero) sets an infinite lockout time; an administrator must issue the kas unlock command to unlock the account.

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>+

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

In the following example, an administrator using the admin account grants administrative privilege to the user smith, and sets the Authentication Database entry to expire at midnight on 31 December 2000.

```
% kas setfields -name smith -flags ADMIN -expiration 12/31/2000
Password for admin:
```

In the following example, an administrator using the admin account sets the user pat's password to expire in 60 days from when it last changed, and prohibits reuse of passwords.

```
% kas setfields -name pat -pwexpires 60 -reuse no
Password for admin:
```

Privilege Required

The issuer must have the ADMIN flag set on his or her Authentication Database entry.

See Also

"kaserverauxdb(5)" on page 473, "kas(8)" on page 805, "kas_examine(8)" on page 817, "kas_setpassword(8)" on page 847, "kas_unlock(8)" on page 857, "klog(1)" on page 153, "kpasswd(1)" on page 163

Copyright

kas setpassword

Name

`kas setpassword` — Changes the key field in an Authentication Database entry

Synopsis

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]

kas setpasswd -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]

kas setp -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]

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

The kas setpassword command accepts a character string of unlimited length, scrambles it into a form suitable for use as an encryption key, places it in the key field of the Authentication Database entry named by the -name argument, and assigns it the key version number specified by the -kvno argument.

To avoid making the password string visible at the shell prompt, omit the -new_password argument. Prompts then appear at the shell which do not echo the password visibly.

When changing the afs server key, also issue bos addkey command to add the key (with the same key version number) to the `/usr/afs/etc/KeyFile` file. See the *IBM AFS Administration Guide* for instructions.

The command interpreter checks the password string subject to the following conditions:

- If there is a program called kpwvalid in the same directory as the kas binary, the command interpreter invokes it to process the password. For details, see "kpwvalid(8)" on page 867.
- If the -reuse argument to the kas setfields command has been used to prohibit reuse of previous passwords, the command interpreter verifies that the password is not too similar too any of the user's previous 20 passwords. It generates the following error message at the shell:

```
Password was not changed because it seems like a reused password
```

To prevent a user from subverting this restriction by changing the password twenty times in quick succession (manually or by running a script), use the -minhours argument on the kaserver initialization command. The following error message appears if a user attempts to change a password before the minimum time has passed:

```
Password was not changed because you changed it too
recently; see your systems administrator
```

Options

-name <*name of user*>

Names the entry in which to record the new key.

-new_password <*new password*>

Specifies the character string the user types when authenticating to AFS. Omit this argument and type the string at the resulting prompts so that the password does not echo visibly. Note that some non-AFS programs cannot handle passwords longer than eight characters.

-kvno <*key version number*>

Specifies the key version number associated with the new key. Provide an integer in the range from `0` through `255`. If omitted, the default is `0` (zero), which is probably not desirable for server keys.

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>+

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

In the following example, an administrator using the `admin` account changes the password for `pat` (presumably because `pat` forgot the former password or got locked out of his account in some other way).

```
% kas setpassword pat
Password for admin:
new_password:
Verifying, please re-enter new_password:
```

Privilege Required

Individual users can change their own passwords. To change another user's password or the password (server encryption key) for server entries such as `afs`, the issuer must have the `ADMIN` flag set in his or her Authentication Database entry.

See Also

"bos_addkey(8)" on page 669, "kas(8)" on page 805, "kaserver(8)" on page 859, "kpwvalid(8)" on page 867

Copyright

kas statistics

Name

`kas statistics` — Displays statistics from an Authentication Server process

Synopsis

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 sta [-a <*admin principal to use for authentication*>] [-p <*admin password*>] [-c <*cell name*>] [-s <*explicit list of authentication servers*>+] [-n] [-h]

Description

The kas statistics command displays statistics from the Authentication Server running on one of the cell's database server machines. Use the -servers argument to name a specific machine, or the command interpreter chooses one at random from all the database server machines with which it has established connections.

Cautions

The -servers argument is not available in interactive mode, making it impossible to specify a certain machine.

Options

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>+

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The information in the output includes:

- The number of allocation and freeing operations the Authentication Server has performed, and how many password change requests it has processed.
- An indication of its hash table use.
- The server machine's IP address in hexadecimal and the date when the current instance of the Authentication Server started.
- The number of requests and aborted requests for various services: authentication, ticket granting, password setting, entry listing, and so on.
- The amount of CPU time that the Authentication Server has used to process requests since it started. The amount is not accurate on all system types, however.
- The number of entries in the Authentication Database that are marked with the `ADMIN` flag.

Examples

In the following example, an administrator using the admin account gathers statistics from the Authentication Server running on the machine `fs1.abc.com`.

```
% kas statistics -servers fs1.abc.com
56 allocs, 46 frees, 0 password changes
Hash table utilization = 0.100000%
From host bfff21a7 started at Tue Mar 23 12:42:02 1999:
  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.
3 admin accounts
```

Privilege Required

The issuer must have the `ADMIN` flag set on his or her Authentication Database entry.

See Also

"kas(8)" on page 805

Copyright

kas stringtokey

Name

`kas stringtokey` — Converts a character string into an octal key

Synopsis

kas stringtokey -string <*password string*> [-cell <*cell name*>] [-help]

kas str -s <*password string*> [-c <*cell name*>] [-h]

Description

The kas stringtokey command converts the character string specified with the -string argument into an octal string suitable for use as an encryption key.

The kas command interpreter generates the octal key by using an encryption algorithm on the combination of the specified string and the name of the local cell (as recorded in the local `/usr/vice/etc/ThisCell` file). Use the -cell argument to convert a string into a key appropriate for a cell other than the local one.

Cautions

This command writes the key to the standard output stream, on which it can possibly be intercepted by third parties. It is not very secure to use the key in an actual Authentication Database entry.

Options

-string <*password string*>

Specifies the character string to convert into an octal key.

-cell <*cell name*>

Specifies the complete Internet domain name of the cell to combine with the password string while generating the key. If this argument is omitted, the kas command interpreter determines the name of the local cell by consulting:

- First, the value of the environment variable AFSCELL.
- Second, the cellname in the `/usr/vice/etc/ThisCell` file on the local machine.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The output is of the following form:

```
Converting I&lt;password string&gt; in realm ’I&lt;cell_name&gt;’ yields key=’I&lt;key&g
```

Examples

The following example shows the octal key equivalent of the string `new_pswd` in the ABC Corporation cell.

```
% kas stringtokey new_pswd
Converting new_pswd in realm ’ABC.COM’ yields
    key=’\346\307\364\320\263\233\342\354’.
```

Privilege Required

None, and no password is required.

See Also

"ThisCell(5)" on page 435, "kas(8)" on page 805

Copyright

kas unlock

Name

`kas unlock` — Unlocks a locked user account

Synopsis

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]

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

The kas unlock command unlocks the Authentication Database entry named by the -name argument. An entry becomes locked when the user exceeds the limit on failed authentication attempts, generally by providing the wrong password to either an AFS-modified login utility or the klog command. Use the kas setfields command to set the limit and the lockout time, and the kas examine command to examine the settings.

To unlock all locked user accounts at once, shutdown the kaserver process on every database server machine, and remove the `/usr/afs/local/kaauxdb` file from each one. The kaserver process recreates the file as it restarts.

Options

-name <*authentication ID*>

Names the Authentication Database entry to unlock.

-admin_username <*admin principal*>

Specifies the user identity under which to authenticate with the Authentication Server for execution of the command. For more details, see "kas(8)" on page 805.

-password_for_admin <*admin password*>

Specifies the password of the command's issuer. If it is omitted (as recommended), the kas command interpreter prompts for it and does not echo it visibly. For more details, see "kas(8)" on page 805.

-cell <*cell name*>

Names the cell in which to run the command. For more details, see "kas(8)" on page 805.

-servers <*authentication servers*>+

Names each machine running an Authentication Server with which to establish a connection. For more details, see "kas(8)" on page 805.

-noauth

Assigns the unprivileged identity anonymous to the issuer. For more details, see "kas(8)" on page 805.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

In the following example, an administrator using the admin account unlocks the entry for jones:

```
% kas unlock -name jones -admin_username admin
Administrator's (admin) Password:
```

Privilege Required

The issuer must have the ADMIN flag set on his or her Authentication Database entry.

See Also

"kas(8)" on page 805, "kas_examine(8)" on page 817, "kas_setfields(8)" on page 841, "klog(1)" on page 153

Copyright

kaserver

Name

kaserver — Initializes the Authentication Server

Description

kaserver [-noAuth] [-fastKeys] [-database <*dbpath*>] [-localfiles <*lclpath*>] [-minhours <*n*>] [-servers <*serverlist*>] [-enable_peer_stats] [-enable_process_stats] [-help]

Description

The kaserver command initializes the Authentication Server, which runs on every database server machine. In the conventional configuration, its binary file is located in the */usr/afs/bin* directory on a file server machine.

The kaserver command is not normally issued at the command shell prompt but rather placed into a file server machine's */usr/afs/local/BosConfig* file with the bos create command. If it is ever issued at the command shell prompt, the issuer must be logged onto a database server machine as the local superuser root.

As it initializes, the Authentication Server process creates the two files that constitute the Authentication Database, *kaserver.DB0* and *kaserver.DBSYS1*, in the */usr/afs/db* directory if they do not already exist. Use the commands in the kas suite to administer the database.

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 and tokens that users and servers use to establish secure connections. Its Ticket Granting Service (TGS) component performs this function.

The Authentication Server records a trace of its activity in the */usr/afs/logs/AuthLog* file. Use the bos getlog command to display the contents of the file. Use the kdb command to read the protected files associated with the *AuthLog* file, *AuthLog.dir* and *AuthLog.pag*.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Options

-noAuth

Assigns 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 test flag for use by the AFS Development staff; it serves no functional purpose.

-database <*dbpath*>

Specifies the pathname of an alternate directory in which the Authentication Database files reside. Provide the complete pathname, ending in the base filename to which the `.DB0` and `.DBSYS1` extensions are appended. For example, the appropriate value for the default database files is */usr/afs/db/kaserver*.

Provide the -localfiles argument along with this one; otherwise, the -localfiles argument is also set to the value of this argument, which is probably inappropriate.

-localfiles <*lclpath*>

Specifies the pathname of an alternate directory in which the auxiliary Authentication Database file resides. Provide the complete pathname, ending in the base filename to which the `auxdb` suffix is appended. For example, the appropriate value for the default auxiliary database file is */usr/afs/local/kaserver*.

-minhours <*n*>

Specifies the minimum number of hours that must pass between 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 <*authentication servers*>+

Names each database server machine running an Authentication Server with which the local Authentication Server is to synchronize its copy of the Authentication Database, rather than with the machines listed in the local */usr/afs/etc/CellServDB* file.

-enable_peer_stats

Activates the collection of Rx statistics and allocates memory for their storage. For each connection with a specific UDP port on another machine, a separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received. To display or otherwise access the records, use the Rx Monitoring API.

-enable_process_stats

Activates the collection of Rx statistics and allocates memory for their storage. A separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received, aggregated over all connections to other machines. To display or otherwise access the records, use the Rx Monitoring API.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following bos create command creates a `kaserver` process on `fs3.abc.com` (the command appears on two lines here only for legibility):

```
% bos create -server fs3.abc.com -instance kaserver \
             -type simple -cmd /usr/afs/bin/kaserver
```

Privilege Required

The issuer must be logged in as the superuser `root` on a file server machine to issue the command at a command shell prompt. It is conventional instead to create and start the process by issuing the bos create command.

See Also

"AuthLog(5)" on page 405, "BosConfig(5)" on page 409, "CellServDB(5)" on page 415, "kaserver.DB0(5)" on page 471, "kaserverauxdb(5)" on page 473, "bos(8)" on page 661, "bos_create(8)" on page 677, "bos_getlog(8)" on page 691, "kas(8)" on page 805, "kdb(8)" on page 863

Copyright

kdb

Name

kdb — Displays log or privileged actions performed by the Authentication Server

Synopsis

kdb [-dbmfile <*dbmfile to use (default /usr/afs/logs/AuthLog)*>] [-key <*extract entries that match specified key*>] [-help]

Description

The kdb command displays the contents of the *AuthLog.dir* and *AuthLog.pag* files associated with the *AuthLog* file that resides on the local disk, by default in the */usr/afs/logs* directory. The files must exist in that directory, which normally implies that the Authentication Server is running on the machine. The files contain information on privileged actions performed by the Authentication Server.

Cautions

It is possible that on some operating systems that AFS otherwise supports, the Authentication Server cannot create the */usr/afs/logs/AuthLog.dir* and */usr/afs/logs/AuthLog.pag* files, making this command inoperative. See the *IBM AFS Release Notes* for details.

Options

-dbmfile <*dbmfile to use*>

Specifies the pathname of the file to display. Provide either a complete pathname, a pathname relative to the */usr/afs/logs* directory, or a filename only, in which case the file must reside in the */usr/afs/logs* directory. Omit this argument to display information from the *AuthLog.dir* and *AuthLog.pag* files in the */usr/afs/logs* directory.

-key <*extract entries that match specified key*>

Specifies each entry to be displayed from the indicated file.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of output indicates the location of the files from which the subsequent information is derived:

```
Printing all entries found in &lt;file_location&gt;
```

Each entry then includes the following two fields, separated by a colon:

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 appears; in cases where no server is directly involved, only the user appears.

service

Identifies one of the following actions or services performed by the user or server process.

- `auth`: Obtained a ticket-granting ticket.
- `chp`: Changed a user password.
- `cruser`: Created a user entry in the Authentication Database.
- `delu`: Deleted a user entry from the Authentication Database.
- `gtck`: Obtained a ticket other than a ticket-granting ticket.
- `setf`: Set fields in an Authentication Database entry.
- `unlok`: Unlocked an Authentication Database entry.

The final line of output sums the number of entries.

Examples

The following example shows the output of the kdb command in the ABC Corporation cell (`abc.com`):

```
% kdb
Printing all entries found in /usr/afs/logs/AuthLog
admin,krbtgt.ABC.COM:auth
admin,afs:gtck
admin:cruser
admin:delu
4 entries were found
```

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"AuthLog.dir(5)" on page 403, "bos_getlog(8)" on page 691, "kaserver(8)" on page 859

Copyright

kpwvalid

Name

`kpwvalid` — Checks quality of new password

Synopsis

kpwvalid

Description

The kpwvalid command checks the quality of a new password passed to it from the kpasswd or kas setpassword command. It is optional. If it exists, it must reside in the same AFS directory as the binaries for the kpasswd and kas command suites (create a symbolic link from the client machine's local disk to this directory). The directory's ACL must extend the `a` (administer) and `w` (write) permissions to the system:administrators group only. These requirements prevent unauthorized users from substituting a spurious kpwvalid binary.

The AFS distribution includes an example kpwvalid program that checks that the password is at least eight characters long; the code for it appears in "EXAMPLES" on page below.

The script or program must accept a sequence of password strings, one per line, on the standard input stream. The first is the current password and is ignored. Each subsequent string is a candidate password to be checked. The program must write the following to the standard output stream for each one:

- `0` (zero) and a newline character to indicate that the password is acceptable.
- A non-zero decimal number and a newline character to indicate that the password is not acceptable.

Further, it must write any error messages only to the standard error stream, not to the standard output stream.

Examples

The following example program, included in the AFS distribution, verifies that the requested password includes eight or more characters.

```
#include &lt;stdio.h&gt;
/* prints 0 if the password is long enough, otherwise non-zero */
main()
{
char oldpassword[512];
char password[512];


if (fgets(oldpassword, 512, stdin))
   while (fgets(password, 512, stdin)) {
```

```
        if (strlen(password) &gt; 8) { /* password includes a newline */
           fputs("0\n",stdout);
           fflush(stdout);
        }
        else {
           fputs("Passwords must contain at least 8 characters.\n",
                 stderr);
           fputs("1\n",stdout);
           fflush(stdout);
        }
return 0;
}
```

See Also

"kas_setpassword(8)" on page 847, "kpasswd(1)" on page 163

Copyright

package

Name

package — Configures files and directories on the local disk

Synopsis

package [*initcmd*] [-config <*base name of configuration file*>] [-fullconfig <*full name of configuration file, or stdin*>] [-overwrite] [-noaction] [-verbose] [-silent] [-rebootfiles] [-debug] [-help]

package [i] [-c <*base name of configuration file*>] [-f <*full name of configuration file, or stdin*>] [-o] [-n] [-v] [-s] [-r] [-d] [-h]

Description

The package command configures the machine's local disk to comply with the instructions in the configuration file named by the -config or -fullconfig argument.

By default, the package command alters any existing local disk element whose contents or configuration does not match the element defined in the configuration file. For example, if a configuration file `D` instruction defines a directory that has the same name as a symbolic link on the local disk, the package command replaces the symbolic link with the directory. The `F` and `L` instructions include an optional *update_code* field that alters this behavior.

Also by default, the package command takes no action on elements on the local disk that are not mentioned in the configuration file. Use the `D` instruction's `R` update code to remove files from the disk directory that are not mentioned in the configuration file.

Before running the package command, the administrator must create the template file and other files on the local disk. For instructions, see the *IBM AFS Administration Guide*.

It is not possible to configure a remote client machine's disk using this command.

Cautions

The package command interpreter exits without executing any instruction if there are any syntax errors or incorrect values in the configuration file.

Options

[*initcmd*]

Accommodates the command's use of the AFS command parser, and is optional.

-config <*base name of configuration file*>

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. The package command determines the machine's system type name and automatically appends it to the base name. An

example of the proper value for this argument is `staff` rather than `staff.rs_aix42`. Partial pathnames are interpreted relative to the current working directory.

Provide this argument or the -fullconfig argument.

-fullconfig <*full name of configuration file, or stdin*>

Specifies the configuration file to use. Two types of values are acceptable:

- The full pathname of the configuration file to use, complete with an extension indicating the machine type (examples: `staff.rs_aix42`, `admin.sun4x_56`).
- The string `stdin` to indicate that the issuer is providing configuration information via the standard input stream, either by piping in the contents of a file, or by typing configuration lines at the shell. In the latter case, type Ctrl-D to conclude the input.

Provide this argument or the -config argument.

-overwrite

Overwrites elements on the local disk with the source version indicated in the configuration file, even if the owner write (`w`) mode bit is turned on the disk element. Files protected by the `I` update code on an `F` line in the configuration file are not overwritten.

-noaction

Checks the sequence of operations to be performed when the command actually runs and reports any problems that the package command interpreter expects to encounter. No elements on the local disk or in AFS are changed. If the -verbose flag is also provided, the trace includes all actions to be performed as well as anticipated errors.

-silent

Suppresses some of the trace messages sent to the standard output stream by default. The output still reports major problems.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-rebootfiles

Prevents overwriting of any file marked with the `Q` update code on an `F` line in the configuration file. This effectively prevents the machine from rebooting automatically again when the package command is invoked in the machine's AFS initialization file.

-debug

Enables debugging output, which is directed to the standard output stream by default. By default, no debugging output is produced.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

This command is usually invoked in a client machine's AFS initialization file (*/etc/rc* or equivalent), rather than issued at the command shell prompt.

The following command 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 example uses the configuration file whose basename is defined in the */.package* file on the local machine. This method enables the administrator to use the same package command in every machine's AFS initialization file but still customize configuration by putting the appropriate basename in the */.package* file.

```
# /etc/package -c ‘cat /.package‘ -v
```

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"package(5)" on page 475

Copyright

prdb_check

Name

prdb_check — Checks the integrity of the Protection Database

Synopsis

prdb_check -database <*ptdb file*> [-uheader] [-pheader] [-entries] [-verbose] [-help]

prdb_check -d <*ptdb file*> [-u] [-p] [-e] [-v] [-h]

Description

The prdb_check command checks the integrity of the Protection Database, reporting any errors or corruption it finds. If there are problems, do not issue any pts commands until the database is repaired.

Cautions

The results can be unpredictable if the Protection Server makes changes to the Protection Database while this command is running. Use the bos shutdown command to shutdown the local ptserver process before running this command, or before creating a second copy of the *prdb.DB0* file (with a different name) on which to run the command.

Options

-database <*ptdb file*>

Names the Protection Database (copy of the *prdb.DB0* file) to check. If the current working directory is not the location of the file, provide a pathname, either full or relative to the current working directory.

-uheader

Displays information which Ubik maintains in the database's header.

-pheader

Displays information which the Protection Server maintains in the database's header.

-entries

Outputs every entry in the database. Some of the information is similar to that returned by the pts examine command.

-verbose

Reports additional information about the database, including the number of entries in the database and a trace of the internal database structures the command is verifying.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If there are errors in the database, the output always reports them on the standard error stream. If any options other than -database or -help are provided, the output written to the standard output stream includes additional information as described for each option in "OPTIONS" on page . The output is intended for debugging purposes and is meaningful to someone familiar with the internal structure of the Protection Database.

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"prdb.DB0(5)" on page 487, "bos_shutdown(8)" on page 743, "pts_examine(1)" on page 195, "ptserver(8)" on page 875

Copyright

ptserver

Name

`ptserver` — Initializes the Protection Server

Synopsis

ptserver [-database <*db path*>] [-p <*number of processes*>] [-rebuildDB] [-enable_peer_stats] [-enable_process_stats] [-help]

Description

The ptserver command initializes the Protection Server, which must run on every database server machine. In the conventional configuration, its binary file is located in the */usr/afs/bin* directory on a file server machine.

The ptserver command is not normally issued at the command shell prompt, but rather placed into a database server machine's */usr/afs/local/BosConfig* file with the bos create command. If it is ever issued at the command shell prompt, the issuer must be logged onto a file server machine as the local superuser `root`.

The Protection Server performs the following tasks:

- Maintains the Protection Database, which contains entries for every user and group in the cell. Use the pts commands to administer the database.
- Allocates AFS IDs for new user, machine and group entries and maps each ID to the corresponding name.
- Generates a current protection subgroup (CPS) at the File Server's request. The CPS lists all groups to which a user or machine belongs.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Options

-database <*db path*>

Specifies the pathname of an alternate directory in which the Protection Database files reside. Provide the complete pathname, ending in the base filename to which the `.DB0` and `.DBSYS1` extensions are appended. For example, the appropriate value for the default database files is */usr/afs/db/prdb*.

-p <*number of processes*>

Sets the number of server lightweight processes (LWPs) to run. Provide a positive integer from the range `3` to `16`. The default value is `3`.

-rebuildDB

Rebuilds the Protection Database at the beginning of Protection Server initialization. Use this argument only in consultation with AFS Development or Product Support.

-enable_peer_stats

Activates the collection of Rx statistics and allocates memory for their storage. For each connection with a specific UDP port on another machine, a separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received. To display or otherwise access the records, use the Rx Monitoring API.

-enable_process_stats

Activates the collection of Rx statistics and allocates memory for their storage. A separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received, aggregated over all connections to other machines. To display or otherwise access the records, use the Rx Monitoring API.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following bos create command creates a `ptserver` process on the machine `fs3.abc.com`. The command appears here on multiple lines only for legibility.

```
% bos create -server fs3.abc.com -instance ptserver \
             -type simple -cmd /usr/afs/bin/ptserver
```

Privilege Required

The issuer must be logged in as the superuser `root` on a file server machine to issue the command at a command shell prompt. It is conventional instead to create and start the process by issuing the bos create command.

See Also

"BosConfig(5)" on page 409, "prdb.DB0(5)" on page 487, "bos_create(8)" on page 677, "bos_getlog(8)" on page 691, "pts(1)" on page 173

Copyright

salvager

Name

`salvager` — Initializes the Salvager component of the fs process

Synopsis

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 <*name of dir to place tmp files*>] [-showlog] [-showsuid] [-showmounts] [-orphans (ignore | remove | attach)] [-help]

Description

The salvager command initializes the Salvager component of the `fs` process. In the conventional configuration, its binary file is located in the */usr/afs/bin* directory on a file server machine.

The Salvager restores internal consistency to corrupted read / write volumes on the local file server machine where possible. For read-only or backup volumes, it inspects only the volume header:

- If the volume header is corrupted, the Salvager removes the volume completely and records the removal in its log file, */usr/afs/logs/SalvageLog*. Issue the vos release or vos backup command to create the read-only or backup volume again.
- If the volume header is intact, the Salvager skips the volume (does not check for corruption in the contents). However, if the File Server notices corruption as it initializes, it sometimes refuses to attach the volume or bring it online. In this case, it is simplest to remove the volume by issuing the vos remove or vos zap command. Then issue the vos release or vos backup command to create it again.

Unlike other server process initialization commands, the salvager command is designed to be issued at the command shell prompt, as well as being placed into a file server machine's */usr/afs/local/BosConfig* file with the bos create command. It is also possible to invoke the Salvager remotely by issuing the bos salvage command.

Combine the command's options as indicated to salvage different numbers of read / write volumes:

- To salvage all volumes on the file server machine, provide no arguments. No volumes on the machine are accessible to Cache Managers during the salvage, because the BOS Server stops the File Server and Volume Server processes while the Salvager runs.
- To salvage all of the volumes on one partition, provide the -partition argument. As for a salvage of all volumes on the machine, no volumes on the machine are accessible to Cache Managers during the salvage operation.

- To salvage only one volume, combine the -partition and -volumeid arguments. Only that volume is inaccessible to Cache Managers, because the BOS Server does not shutdown the File Server and Volume Server processes.

The Salvager normally salvages only those read / write volumes that are marked as having been active when a crash occurred. To have it salvage all relevant read / write volumes, add the -force flag.

The Salvager normally creates new inodes as it repairs damage. If the partition is so full that there is no room for new inodes, use the -nowrite argument to bringing undamaged volumes online without attempting to salvage damaged volumes. Then use the vos move command to move one or more of the undamaged volumes to other partitions, freeing up the space that the Salvager needs to create new inodes.

By default, multiple Salvager subprocesses run in parallel: one for each partition up to four, and four subprocesses for four or more partitions. To increase or decrease the number of subprocesses running in parallel, provide a positive integer value for the -parallel argument.

If there is more than one server partition on a physical disk, the Salvager by default salvages them serially to avoid the inefficiency of constantly moving the disk head from one partition to another. However, this strategy is often not ideal if the partitions are configured as logical volumes that span multiple disks. To force the Salvager to salvage logical volumes in parallel, provide the string `all` as the value for the -parallel argument. Provide a positive integer to specify the number of subprocesses to run in parallel (for example, `-parallel 5all` for five subprocesses), or omit the integer to run up to four subprocesses, depending on the number of logical volumes being salvaged.

The Salvager creates temporary files as it runs, by default writing them to the partition it is salvaging. The number of files can be quite large, and if the partition is too full to accommodate them, the Salvager terminates without completing the salvage operation (it always removes the temporary files before exiting). Other Salvager subprocesses running at the same time continue until they finish salvaging all other partitions where there is enough disk space for temporary files. To complete the interrupted salvage, reissue the command against the appropriate partitions, adding the -tmpdir argument to redirect the temporary files to a local disk directory that has enough space.

The -orphans argument controls how the Salvager handles orphaned files and directories that it finds on server partitions it is salvaging. An *orphaned* element is completely inaccessible because it is not referenced by the vnode of any directory that can act as its parent (is higher in the filespace). Orphaned objects occupy space on the server partition, but do not count against the volume's quota.

To generate a list of all mount points that reside in one or more volumes, rather than actually salvaging them, include the -showmounts flag.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Options

[*initcmd*]

Accommodates the command's use of the AFS command parser, and is optional.

-partition <*name of partition to salvage*>

Specifies the name of the partition to salvage. Specify the full partition name using the form `/vicepx` or `/vicepxx`. Omit this argument to salvage every partition on the file server machine.

-volumeid <*volume id to salvage*>

Specifies the volume ID of a specific read / write volume to salvage. The -partition argument must be provided along with this one and specify the volume's actual site.

-debug

Allows only one Salvager subprocess to run at a time, regardless of the setting of the -parallel option. Include it when running the Salvager in a debugger to make the trace easier to interpret.

-nowrite

Brings all undamaged volumes online without attempting to salvage any damaged volumes.

-inodes

Records in the `/usr/afs/logs/SalvageLog` file a list of all AFS inodes that the Salvager modified.

-force

Inspects all volumes for corruption, not just those that are marked as having been active when a crash occurred.

-oktozap

Removes a volume that is so damaged that even issuing the vos zap command with the -force flag is ineffective. Use this argument only in consultation with AFS Development or Product Support. Combine it with the -partition and -volumeid arguments to identify the volume to remove.

-rootinodes

Records in the `/usr/afs/logs/SalvageLog` file a list of all AFS inodes owned by the local superuser `root`.

-salvagedirs

Salvages entire directory structures, even if they do not appear to be damaged. By default, the Salvager salvages a directory only if it is flagged as corrupted.

-blockreads

Forces the 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 <*# of max parallel partition salvaging*>

Specifies the maximum number of Salvager subprocesses to run in parallel. Provide one of three values:

- An integer from the range `1` to `32`. A value of `1` means that a single Salvager process salvages the partitions sequentially.
- The string `all` to run up to four Salvager subprocesses in parallel on partitions formatted as logical volumes that span multiple physical disks. Use this value only with such logical volumes.
- The string `all` followed immediately (with no intervening space) by an integer from the range `1` to `32`, to run the specified number of Salvager subprocesses in parallel on partitions formatted as logical volumes. Use this value only with such logical volumes.

The BOS Server never starts more Salvager subprocesses than there are partitions, and always starts only one process to salvage a single volume. If this argument is omitted, up to four Salvager subprocesses run in parallel.

-tmpdir <*name of dir to place tmp files*>

Names a local disk directory in which the Salvager places the temporary files it creates during a salvage operation, instead of writing them to the partition being salvaged (the default). If the Salvager cannot write to the specified directory, it attempts to write to the partition being salvaged.

-showlog

Displays on the standard output stream all log data that is being written to the */usr/afs/logs/SalvageLog* file.

-showsuid

Displays a list of the pathnames for all files that have the setuid or setgid mode bit set.

-showmounts

Records in the */usr/afs/logs/SalvageLog* file all mount points found in each volume. The Salvager does not repair corruption in the volumes, if any exists.

-orphans (ignore | remove | attach)

Controls how the Salvager handles orphaned files and directories. Choose one of the following three values:

ignore

Leaves the orphaned objects on the disk, but prints a message to the */usr/afs/logs/SalvageLog* file reporting how many orphans were found and the approximate number of kilobytes they are consuming. This is the default if the -orphans argument is omitted.

remove

Removes the orphaned objects, and prints a message to the */usr/afs/logs/SalvageLog* file reporting how many orphans were removed and the approximate number of kilobytes they were consuming.

attach

Attaches the orphaned objects by creating a reference to them in the vnode of the volume's root directory. Since each object's actual name is now lost, the Salvager assigns each one a name of the following form:

__ORPHANFILE__.*index* for files.

__ORPHANDIR__.*index* for directories.

where *index* is a two-digit number that uniquely identifies each object. The orphans are charged against the volume's quota and appear in the output of the ls command issued against the volume's root directory.

-help

Prints the online help for this command. All other valid options are ignored.

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 -partition /vicepg -volumeid 258347486
```

Privilege Required

To issue the command at the shell prompt, the issuer must be logged in as the local superuser root.

See Also

"BosConfig(5)" on page 409, "SalvageLog(5)" on page 433, "bos_create(8)" on page 677, "bos_getlog(8)" on page 691, "bos_salvage(8)" on page 727, "vos_move(1)" on page 347

Copyright

upclient

Name

upclient — Initializes the client portion of the Update Server

Synopsis

upclient <*hostname*> [-crypt] [-clear] [-t <*retry time*>] [-verbose]* <*dir*>+ [-help]

Description

The upclient command initializes the client portion of the Update Server. In the conventional configuration, its binary file is located in the */usr/afs/bin* directory on a file server machine.

The upclient command is not normally issued at the command shell prompt but rather placed into a file server machine's */usr/afs/local/BosConfig* file with the bos create command. If it is ever issued at the command shell prompt, the issuer must be logged onto a database server machine as the local superuser root.

The upclient process periodically checks that all files in each local directory named by the *dir* argument match the files in the corresponding directory on the source machine named by the *hostname* argument. If a file does not match, the upclient process requests the source copy from the upserver process running on the source machine.

By default, the upclient process requests that the upserver process encrypt the data before transferring it. Use the -clear flag to request unencrypted transfer if appropriate. (The -crypt flag explicitly sets the default.)

In the conventional configuration, separate instances of the upclient process request data from the */usr/afs/bin* and */usr/afs/etc* directories, except on machines for which the system control machine is also the binary distribution machine for the machine's system type. The conventional names for the separate instances are upclientbin and upclientetc respectively.

The upclient and upserver processes always mutually authenticate, whether or not the data they pass is encrypted; they use the key with the highest key version number in the */usr/afs/etc/KeyFile* file to construct a server ticket for mutual authentication.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Cautions

Do not use the Update Server to distribute the contents of the */usr/afs/etc* directory using the -clear option. The contents of this directory are sensitive.

Options

<*hostname*>

Names either the cell's system control machine (if the requested directory is `/usr/afs/etc`), or the binary distribution machine for the local machine's CPU and operating system type (if the requested directory is `/usr/afs/bin`).

-crypt

Requests the transfer of data from the upserver process in encrypted form. This is the default; this flag just sets the default explicitly. Do not use this flag with the -clear flag.

-clear

Requests transfer of data from the upserver process in unencrypted form. Provide this flag or the -crypt flag, but not both.

-t <*retry time*>

Specifies how often to check for changes in each specified directory, as a number of seconds. If this argument is omitted, the default is `300` (5 minutes). This argument determines the maximum amount of time it takes for a change made on the source machine to propagate to this machine.

-verbose*

Writes a trace of the upclient process's operations on the standard output stream, which usually corresponds to the machine console. Provide one, two, or three instances of the flag; each additional instance generates increasingly numerous and detailed messages.

<*dir*>+

Names each directory to check for modified files. The conventional choices are the following:

- `/usr/afs/bin`, in which case the recommended name for the process (assigned with the -instance argument to the bos create command) is `upclientbin`. The *hostname* is the binary distribution machine for the local machine's system type. You may wish to use the -clear flag for the `/usr/afs/bin` directory, since binaries are not particularly sensitive and encrypting them takes system resources.
- `/usr/afs/etc`, in which case the recommended name for the process (assigned with the -instance argument to the bos create command) is `upclientetc`. The *hostname* is the cell's system control machine. Use the -crypt flag for the `/usr/afs/etc` directory, since it contains the `KeyFile` file and other data vital to cell security.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following bos create command creates an `upclientbin` process on the machine `fs4.abc.com` that refers to the machine `fs1.abc.com` as the source for the */usr/afs/bin* directory (thus `fs1.abc.com` is the binary distribution machine for machines of `fs4.abc.com`'s type). The files in the */usr/afs/bin* directory are distributed every 120 seconds. The command requests transfer in unencrypted form.

```
% bos create  -server fs4.abc.com -instance upclientbin -type simple \
              -cmd "/usr/afs/bin/upclient fs1.abc.com -clear \
              -t 120 /usr/afs/bin"
```

Privilege Required

The issuer must be logged in as the superuser `root` on a file server machine to issue the command at a command shell prompt. It is conventional instead to create and start the process by issuing the bos create command.

See Also

"BosConfig(5)" on page 409, "bos_create(8)" on page 677, "upserver(8)" on page 889

Copyright

upserver

Name

upserver — Initializes the server portion of the Update Server

Synopsis

upserver [<*directory*>+] [-crypt <*directory*>+] [-clear <*directory*>+] [-auth <*directory*>+] [-help]

Description

The upserver command initializes the server portion of the Update Server (the upserver process). In the conventional configuration, its binary file is located in the */usr/afs/bin* directory on a file server machine.

The upserver command is not normally issued at the command shell prompt but rather placed into a file server machine's */usr/afs/local/BosConfig* file with the bos create command. If it is ever issued at the command shell prompt, the issuer must be logged onto a database server machine as the local superuser root.

The upserver command specifies which of the directories on the local disk are eligible for distribution in response to requests from the client portion of the Update Server (the upclient process) running on other machines. If no directories are specified, the upserver process distributes the contents of any directory on its local disk.

The upserver process can distribute a directory's contents in encrypted or unencrypted form. By default, it does not use encryption unless an upclient process requests it (this default is equivalent to setting the -clear flag). When the -crypt flag is provided, the upserver process only fulfills requests for encrypted transfer.

The upclient and upserver processes always mutually authenticate, whether or not the data they pass is encrypted; they use the key with the highest key version number in the */usr/afs/etc/KeyFile* file to construct a server ticket for mutual authentication.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Cautions

Do not use the Update Server to distribute the contents of the */usr/afs/etc* directory without the -crypt flag. The contents of this directory are sensitive.

Options

<*directory*>+

Names each directory to distribute in unencrypted form (because they appear before the first -crypt or -clear flag on the command line). If this argument is omitted, all directories

on the machine's local disk are eligible for distribution.

-crypt <*directory*>+

Precedes a list of one or more directories that the upserver process distributes only in encrypted form.

-clear <*directory*>+

Precedes a list of one or more directories that the upserver process distributes in unencrypted form unless the upclient process requests them in encrypted form. Use this argument only if a list of directories headed by the -crypt flag precedes it on the command line.

-auth <*directory*>+

Precedes a list of one or more directories which the upserver process distributes using a form of encryption that is intermediate in complexity and security between the unencrypted and encrypted levels set by the -clear and -crypt arguments. Do not use this argument, because the upclient process does not have a corresponding argument that it can use to request data transfer at this level.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example bos create command defines and starts an upserver process on the host machine `fs1.abc.com`. The last parameter (enclosed in quotes) instructs the upserver process to distribute the contents of the */usr/afs/bin* directory in unencrypted form and the contents of the */usr/afs/etc* directory in encrypted form.

```
% bos create  -server fs1.abc.com -instance upserver -type simple \
    -cmd "/usr/afs/bin/upserver /usr/afs/bin -crypt /usr/afs/etc"
```

Privilege Required

The issuer must be logged in as the superuser `root` on a file server machine to issue the command at a command shell prompt. It is conventional instead to create and start the process by issuing the bos create command.

See Also

"BosConfig(5)" on page 409, "bos_create(8)" on page 677, "upclient(8)" on page 885

Copyright

uss

Name

uss — Introduction to the uss command suite

Description

The commands in the uss command suite help administrators to create AFS user accounts more easily and efficiently. If uss commands are not used, creating an account requires issuing at least six separate commands to five different AFS servers.

There are three main commands in the suite:

- The uss add command creates a single complete user account, based on command line arguments and instructions in a template file.
- The uss bulk command creates multiple complete accounts at once, based on command line arguments, instructions in a template file and a bulk input file.
- The uss delete command removes most parts of a user account.

To obtain help, issue the uss apropos and uss help commands.

Options

The following arguments and flags are available on many commands in the uss suite. The reference page for each command also lists them, but they are described here in greater detail.

-admin <*administrator to authenticate*>

Specifies the AFS user name under which to establish a connection to the AFS server processes that administer the various parts of a user account. If it is omitted, the connection is established under the issuer's effective user ID (his or her identity in the local file system). Even when this argument is included, UNIX commands that run during the uss operation (for instance, the UNIX */etc/chown* command) run under the effective user ID.

-cell <*cell name*>

Names the cell in which to run the command. It is acceptable to abbreviate the cell name to the shortest form that distinguishes it from the other entries in the */usr/vice/etc/CellServDB* file on the local machine. If the -cell argument is omitted, the command interpreter determines the name of the local cell by reading the following in order:

- The value of the AFSCELL environment variable.
- The local */usr/vice/etc/ThisCell* file.

-dryrun

Reports actions that the command interpreter needs to perform when executing the uss operation, without actually performing them. Include this flag to verify that the command produces the desired account configuration. Combine it with the -verbose flag to yield even more detailed information. Note that the output does not necessarily reveal all possible problems that can prevent successful execution of the command, especially those that result from transient server or network outages.

-help

Prints a command's online help message on the standard output stream. Do not combine this flag with any of the command's other options; when it is provided, the command interpreter ignores all other options, and only prints the help message.

-skipauth

Bypasses mutual authentication with the AFS Authentication Server, allowing a site that uses Kerberos instead of the AFS Authentication Server to substitute that form of authentication.

Privilege Required

The issuer of a uss command must have all the rights required for performing the equivalent actions individually. See each uss command's reference page.

See Also

"uss(5)" on page 497, "uss_bulk(5)" on page 511, "uss_add(8)" on page 895, "uss_apropos(8)" on page 901, "uss_bulk(8)" on page 903, "uss_delete(8)" on page 907, "uss_help(8)" on page 911

Copyright

uss add

Name

uss add — Creates a user account

Synopsis

uss add -user <*login name*> [-realname <*full name in quotes*>] [-pass <*initial password*>] [-pwexpires <*password expires in [0..254] days (0 => never)*>] [-server <*file server for home volume*>] [-partition <*file server'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] [-help]

uss ad -us <*login name*> [-r <*full name in quotes*>] [-pas <*initial password*>] [-pw <*password expires in [0..254] days (0 => never)*>] [-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] [-h]

Description

The uss add command creates entries in the Protection Database and Authentication Database for the user name specified by the -user argument. By default, the Protection Server automatically allocates an AFS user ID (UID) for the new user; to specify an alternate AFS UID, include the -uid argument. If a password is provided with the -pass argument, it is stored as the user's password in the Authentication Database after conversion into a form suitable for use as an encryption key. Otherwise, the string changeme is assigned as the user's initial password.

The other results of the command depend on which instructions and which of a defined set of variables appear in the template file specified with the -template argument. Many of the command's arguments supply a value for one of the defined variables, and failure to provide an argument when the corresponding variable appears in the template file halts the account creation process at the point where the command interpreter first encounters the variable in the template file.

To create multiple accounts with a single command, use the uss bulk command. To delete accounts with a single command, use the uss delete command.

Options

-user <*login name*>

Names the user's Authentication Database and Protection Database entries. It can include up to eight alphanumeric characters, but not any of the following characters: : (colon), @ (at-sign), . (period), space, or newline. Because it becomes the username (the name under

which a user logs in), it is best not to include shell metacharacters and to obey the restrictions that many operating systems impose on usernames (usually, to contain no more than eight lowercase letters).

Corresponding variable in the template file: $USER.

-realname <*full name in quotes*>

Specifies the user's full name. If it contains spaces or punctuation, surround it with double quotes. If not provided, it defaults to the user name provided with the -user argument.

Corresponding variable in the template file: $NAME. Many operating systems include a field for the full name in a user's entry in the local password file (*/etc/passwd* or equivalent), and this variable can be used to pass a value to be used in that field.

-pass <*initial password*>

Specifies the user's initial password. Although the AFS commands that handle passwords accept strings of virtually unlimited length, it is best to use a password of eight characters or less, which is the maximum length that many applications and utilities accept. If not provided, this argument defaults to the string `changeme`.

Corresponding variable in the template file: none.

-pwexpires <*password expiration*>

Sets the number of days after a user's password is changed that it remains valid. Provide an integer from the range `1` through `254` to specify the number of days until expiration, or the value `0` to indicate that the password never expires (the default).

When the password becomes invalid (expires), the user is unable to authenticate, but has 30 more days in which to issue the kpasswd command to change the password (after that, only an administrator can change it).

Corresponding variable in the template file: $PWEXPIRES.

-server <*file server name*>

Names the file server machine on which to create the new user's volume. It is best to provide a fully qualified hostname (for example, `fs1.abc.com`), but an abbreviated form is acceptable provided that the cell's naming service is available to resolve it at the time the volume is created.

Corresponding variable in the template file: $SERVER.

-partition <*file server partition*>

Specifies the partition on which to create the user's volume; it must be on the file server machine named by the -server argument. Provide the complete partition name (for example */vicepa*) or one of the following abbreviated forms:

```
/vicepa     =     vicepa      =      a      =      0
/vicepb     =     vicepb      =      b      =      1
```

After */vicepz* (for which the index is 25) comes

```
/vicepaa    =     vicepaa     =      aa     =      26
```

```
/vicepab    =     vicepab     =      ab     =      27
```

and so on through

```
/vicepiv    =     vicepiv     =      iv     =      255
```

Corresponding variable in the template file: $PART.

-mount <*home directory mount point*>

Specifies the pathname for the user's home directory. Partial pathnames are interpreted relative to the current working directory.

Specify the read / write path to the directory, to avoid the failure that results from attempting to create a new mount point in a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, */afs/.abc.com*). For further discussion of the concept of read / write and read-only paths through the filespace, see the fs mkmount reference page.

Corresponding variable in template: $MTPT, but in the template file's `V` instruction only. Occurrences of the $MTPT variable in template instructions that follow the `V` instruction take their value from the `V` instruction's *mount_point* field. Thus the value of this command line argument becomes the value for the $MTPT variable in instructions that follow the `V` instruction only if the string $MTPT appears alone in the `V` instruction's *mount_point* field.

-uid <*uid to assign the user*>

Specifies a positive integer other than 0 (zero) to assign as the user's AFS UID. If this argument is omitted, the Protection Server assigns an AFS UID that is one greater than the current value of the `max user id` counter (use the pts listmax command to display the counter). If including this argument, it is best first to use the pts examine command to verify that no existing account already has the desired AFS UID; it one does, the account creation process terminates with an error.

Corresponding variable in the template file: $UID.

-template <*pathname of template file*>

Specifies the pathname of the template file. If this argument is omitted, the command interpreter searches the following directories in the indicated order for a file called `uss.template`:

- The current working directory.
- */afs/cellname/common/uss*, where *cellname* names the local cell.
- */etc*

If the issuer provides a filename other than `uss.template` but without a pathname, the command interpreter searches for it in the indicated directories. If the issuer provides a full or partial pathname, the command interpreter consults the specified file only; it interprets partial pathnames relative to the current working directory.

If the specified template file is empty (zero-length), the command creates Protection and Authentication Database entries only.

"uss(5)" on page 497 details the file's format.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-var <*auxilliary argument pairs*>

Specifies values for each of the number variables $1 through $9 that can appear in the template file. Use the number variables to assign values to variables in the uss template file that are not part of the standard set.

Corresponding variables in the template file: $1 through $9.

For each instance of this argument, provide two parts in the indicated order, separated by a space:

- The integer from the range `1` through `9` that matches the variable in the template file. Do not precede it with a dollar sign.
- A string of alphanumeric characters to assign as the value of the variable.

See the chapter on uss in the *IBM AFS Administration Guide* for further explanation.

-cell <*cell name*>

Specifies the cell in which to run the command. For more details, see "uss(8)" on page 893.

-admin <*administrator to authenticate*>

Specifies the AFS user name under which to establish authenticated connections to the AFS server processes that maintain the various components of a user account. For more details, see "uss(8)" on page 893.

-dryrun

Reports actions that the command interpreter needs to perform while executing the command, without actually performing them. For more details, see "uss(8)" on page 893.

-skipauth

Prevents authentication with the AFS Authentication Server, allowing a site using Kerberos to substitute that form of authentication.

-overwrite

Overwrites any directories, files and links that exist in the file system and for which there are definitions in `D`, `E`, `F`, `L`, or `S` instructions in the template file named by the -template argument. If this flag is omitted, the command interpreter prompts once for confirmation that it is to overwrite all such elements.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The combination of the following example uss add command and `V` instruction in a template file called `uss.tpl` creates Protection and Authentication Database entries named `smith`, and a volume called `user.smith` with a quota of 2500 kilobyte blocks, mounted at the pathname */afs/abc.com/usr/smith*. The access control list (ACL) on the mount point grants `smith` all rights.

The issuer of the uss add command provides only the template file's name, not its complete pathname, because it resides in the current working directory. The command and `V` instruction appear here on two lines only for legibility; there are no line breaks in the actual instruction or command.

```
V user.$USER $SERVER.abc.com /vice$PART $1 \
    /afs/abc.com/usr/$USER $UID $USER all


% uss add -user smith -realname "John Smith" -pass js_pswd \
    -server fs2 -partition b -template uss.tpl -var 1 2500
```

Privilege Required

The issuer (or the user named by the -admin argument) must belong to the system:administrators group in the Protection Database and must have the `ADMIN` flag turned on in his or her Authentication Database entry.

If the template contains a `V` instruction, the issuer must be listed in the */usr/afs/etc/UserList* file and must have at least `a` (administer) and `i` (insert) permissions on the ACL of the directory that houses the new mount point. If the template file includes instructions for creating other types of objects (directories, files or links), the issuer must have each privilege necessary to create them.

See Also

"UserList(5)" on page 437, "uss(5)" on page 497, "fs_mkmount(1)" on page 105, "uss(8)" on page 893, "uss_bulk(8)" on page 903, "uss_delete(8)" on page 907

Copyright

uss apropos

Name

`uss apropos` — Displays each help entry containing a keyword string.

Synopsis

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

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

Description

The uss apropos command displays the first line of the online help entry for any uss command that has in its name or short description the string specified by the -topic argument.

To display the syntax for a command, use the uss help command.

Options

-topic <*help string*>+

Specifies the keyword string to match, in lowercase letters only. If the string is more than a single word, surround it with double quotes ("") or other delimiters.

-help

Prints the online help for this command. All other valid options are ignored.

Output

The first line of a command's online help entry names it and briefly describes its function. This command displays the first line for any uss command where the string specified by the -topic argument is part of the command name or first line.

Examples

The following command lists all uss commands that include the word `create` in their names or short descriptions:

```
% uss apropos create
add: create a new user
```

Privilege Required

None

See Also

"uss(8)" on page 893, "uss_help(8)" on page 911

Copyright

uss bulk

Name

`uss bulk` — Executes multiple uss commands listed in a file

Synopsis

uss bulk -file <*bulk input file*> [-template <*pathname of template file*>] [-verbose] [-cell <*cell name*>] [-admin <*administrator to authenticate*>] [-dryrun] [-skipauth] [-overwrite] [-pwexpires <*password expires in [0..254] days (0* = never)>>] [-pipe] [-help]

uss b -f <*bulk input file*> [-t <*pathname of template file*>] [-v] [-c <*cell name*>] [-a <*administrator to authenticate*>] [-d] [-s] [-o] [-pw <*password expires in [0..254] days (0* = never)>>] [-pi] [-h]

Description

The uss bulk command executes the uss commands listed in the *bulk input file* specified with the -file argument. If the bulk input file includes add instructions that reference a template file, then the -template argument is required.

To create a single account, use the uss add command. To delete one or more accounts, use the uss delete command.

Options

-file <*bulk input file*>

Specifies the pathname of the bulk input file. Partial pathnames are interpreted relative to the current working directory. For details on the file's format, see "uss_bulk(5)" on page 511.

-template <*pathname of template file*>

Specifies the pathname of the template file for any uss add commands that appear in the bulk input file. Partial pathnames are interpreted relative to the current working directory. For details on the file's format, see "uss(5)" on page 497.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-cell <*cell name*>

Specifies the cell in which to run the command. For more details, see "uss(8)" on page 893.

-admin <*administrator to authenticate*>

Specifies the AFS user name under which to establish authenticated connections to the AFS server processes that maintain the various components of a user account. For more details, see "uss(8)" on page 893.

-dryrun

Reports actions that the command interpreter needs to perform while executing the command, without actually performing them. For more details, see "uss(8)" on page 893.

-skipauth

Prevents authentication with the AFS Authentication Server, allowing a site using Kerberos to substitute that form of authentication.

-overwrite

Overwrites any directories, files and links that exist in the file system and for which there are also `D`, `E`, `F`, `L`, or `S` instructions in a template file referenced by an `add` instruction in the bulk input file. If this flag is omitted, the command interpreter prompts, once for each `add` instruction in the bulk input file, for confirmation that it should overwrite such elements. Do not include this flag if the bulk input file does not contain `add` instructions.

-pwexpires <*password expiration*>

Sets the number of days after a user's password is changed that it remains valid, for each user named by an `add` instruction in the bulk input file. Provide an integer from the range `1` through `254` to specify the number of days until expiration, or the value `0` to indicate that the password never expires (the default).

When the password becomes invalid (expires), the user is unable to authenticate, but has 30 more days in which to issue the kpasswd command to change the password (after that, only an administrator can change it).

-pipe

Suppresses the Authentication Server's prompt for the password of the issuer or the user named by the -admin argument (the Authentication Server always separately authenticates the creator of an entry in the Authentication Database). Instead, the command interpreter accepts the password via the standard input stream, as piped in from another program. This enables the uss bulk command to run as part of unattended batch jobs.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following example command executes the instructions in the bulk input file called *`new_students`*, which includes `add` instructions that refer to the template file *`student.template`*. Both files reside in the current working directory.

```
% uss bulk new_students student.template
```

Privilege Required

The issuer (or the user named by the -admin argument) must have the privileges necessary to run the commands that correspond to instructions in the bulk input file.

See Also

"uss(5)" on page 497, "uss_bulk(5)" on page 511, "uss(8)" on page 893, "uss_add(8)" on page 895, "uss_delete(8)" on page 907

Copyright

uss delete

Name

`uss delete` — Deletes a user account

Synopsis

uss delete -user <*login name*> [-mountpoint <*mountpoint for user's volume*>] [-savevolume] [-verbose] [-cell <*cell name*>] [-admin <*administrator to authenticate*>] [-dryrun] [-skipauth] [-help]

uss d -u <*login name*> [-m <*mountpoint for user's volume*>] [-sa] [-v] [-c <*cell name*>] [-a <*administrator to authenticate*>] [-d] [-sk] [-h]

Description

The uss delete command removes the Authentication Database and Protection Database entries for the user named by -user argument. In addition, it can remove the user's home volume and associated VLDB entry, a mount point for the volume or both, depending on whether the -mountpoint and -savevolume options are provided.

- To remove both the volume and mount point, use the -mountpoint argument to name the user's home directory. It is best to create a tape backup of a volume before deleting it. Note that other mount points for the volume are not removed, if they exist.
- To remove the mount point only, provide both the -mountpoint and -savevolume options.
- To preserve both the volume and mount point, omit the -mountpoint argument (or both it and the -savevolume flag).

Options

-user <*login name*>

Names the entry to delete from the Protection and Authentication Databases.

-mountpoint <*mountpoint for the user's volume*>

Specifies the pathname to the user's home directory, which is deleted from the filespace. By default, the volume referenced by the mount point is also removed from the file server machine that houses it, along with its Volume Location Database (VLDB) entry. To retain the volume and VLDB entry, include the -savevolume flag. Partial pathnames are interpreted relative to the current working directory.

Specify the read / write path to the mount point, to avoid the failure that results from attempting to remove a mount point from a read-only volume. By convention, the read / write path is indicated by placing a period before the cell name at the pathname's second level (for example, `/afs/.abc.com`). For further discussion of the concept of read / write and read-only paths through the filespace, see the fs mkmount reference page.

-savevolume

Preserves the user's volume and VLDB entry.

-verbose

Produces on the standard output stream a detailed trace of the command's execution. If this argument is omitted, only warnings and error messages appear.

-cell <*cell name*>

Specifies the cell in which to run the command. For more details, see "uss(8)" on page 893.

-admin <*administrator to authenticate*>

Specifies the AFS user name under which to establish authenticated connections to the AFS server processes that maintain the various components of a user account. For more details, see "uss(8)" on page 893.

-dryrun

Reports actions that the command interpreter needs to perform while executing the command, without actually performing them. For more details, see "uss(8)" on page 893.

-skipauth

Prevents authentication with the AFS Authentication Server, allowing a site using Kerberos to substitute that form of authentication.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following command removes smith's user account from the `abc.com` cell. The -savevolume argument retains the `user.smith` volume on its file server machine.

```
% uss delete smith -mountpoint /afs/abc.com/usr/smith -savevolume
```

Privilege Required

The issuer (or the user named by -admin argument) must belong to the system:administrators group in the Protection Database, must have the `ADMIN` flag turned on in his or her Authentication Database entry, and must have at least `a` (administer) and `d` (delete) permissions on the access control list (ACL) of the mount point's parent directory. If the -savevolume flag is not included, the issuer must also be listed in the */usr/afs/etc/UserList* file.

See Also

"UserList(5)" on page 437, "fs_mkmount(1)" on page 105, "uss(8)" on page 893

Copyright

uss help

Name

`uss help` — Displays help for uss commands

Synopsis

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

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

Description

The uss help command displays the complete online help entry (short description and syntax statement) for each command operation code specified by the -topic argument. If the -topic argument is omitted, the output includes the first line (name and short description) of the online help entry for every uss command.

To list every uss command whose name or short description includes a specified keyword, use the uss apropos command.

Options

-topic <*help string*>+

Indicates each command for which to display the complete online help entry. Omit the uss part of the command name, providing only the operation code (for example, specify bulk, not uss bulk). If this argument is omitted, the output briefly describes every uss command.

-help

Prints the online help for this command. All other valid options are ignored.

Output

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

- The first line names the command and briefly describes its function.
- The second line lists aliases for the command, if any.
- The final line, which begins with the string `Usage`, lists the command's options in the prescribed order. Online help entries use the same symbols (for example, brackets) as the reference pages in this document.

Examples

The following command displays the online help entry for the uss bulk command:

```
% uss help bulk
uss bulk: bulk input mode
Usage: uss bulk -file &lt;bulk input file&gt; [-template &lt;pathname
of template file&gt;] [-verbose] [-cell &lt;cell name&gt;] [-admin
&lt;administrator to authenticate&gt;] [-dryrun] [-skipauth] [-overwrite]
[-pwexpires &lt;password expires in [0..254] days (0 =&gt; never)&gt;] [-pipe]
[-help]
```

Privilege Required

None

See Also

"uss(8)" on page 893, "uss_apropos(8)" on page 901

Copyright

vldb_check

Name

`vldb_check` — Checks the integrity of the VLDB

Synopsis

vldb_check -database <*vldb file*> [-uheader] [-vheader] [-servers] [-entries] [-verbose] [-help]

vldb_check -d <*vldb file*> [-u] [-vh] [-s] [-e] [-ve] [-h]

Description

The vldb_check command checks the integrity of the Volume Location Database (VLDB), reporting any errors or corruption it finds. If there are problems, do not issue any vos commands until the database is repaired.

Cautions

The results can be unpredictable if the Volume Location (VL) Server makes changes to the VLDB while this command is running. Use the bos shutdown command to shutdown the local vlserver process before running this command, or before creating a second copy of the *`vldb.DB0`* file (with a different name) on which to run the command.

Options

-database <*vldb file*>

Names the VLDB (copy of the *`vldb.DB0`* file) to check. If the current working directory is not the location of the file, provide a pathname, either full or relative to the current working directory.

-uheader

Displays information which Ubik maintains in the database's header.

-pheader

Displays information which the VL Server maintains in the database's header.

-servers <*authentication servers*>+

Outputs the server entries from the VLDB, which list the IP addresses registered for each file server machine in the cell.

-entries

Outputs every volume entry in the database. The information includes the volume's name and the volume ID number for each of its versions.

-verbose

Reports additional information about the database, including the number of entries for each type of volume.

-help

Prints the online help for this command. All other valid options are ignored.

Output

If there are errors in the database, the output always reports them on the standard error stream. If any options other than -database or -help are provided, the output written to the standard output stream includes additional information as described for each option in "OPTIONS" on page . The output is intended for debugging purposes and is meaningful to someone familiar with the internal structure of the VLDB.

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"vldb.DB0(5)" on page 519, "bos_shutdown(8)" on page 743, "vlserver(8)" on page 915

Copyright

vlserver

Name

`vlserver` — Initializes the Volume Location Server

Synopsis

vlserver [-p <*lwp processes*>] [-nojumbo] [-enable_peer_stats] [-enable_process_stats] [-help]

Description

The vlserver command initializes the Volume Location (VL) Server, which runs on every database server machine. In the conventional configuration, its binary file is located in the */usr/afs/bin* directory on a file server machine.

The vlserver command is not normally issued at the command shell prompt but rather placed into a file server machine's */usr/afs/local/BosConfig* file with the bos create command. If it is ever issued at the command shell prompt, the issuer must be logged onto a database server machine as the local superuser `root`.

As it initializes, the VL Server process creates the two files that constitute the Volume Location Database (VLDB), */vldb.DB0* and */vldb.DBSYS1*, in the */usr/afs/db* directory if they do not already exist. Use the commands in the vos suite to administer the database.

The VL Server maintains the record of volume locations in the Volume Location Database (VLDB). When the Cache Manager fills a file request from an application program, it first contacts the VL Server to learn which file server machine currently houses the volume that contains the file. The Cache Manager then requests the file from the File Server process running on that file server machine.

The VL Server records a trace of its activity in the */usr/afs/logs/VLLog* file. Use the bos getlog command to display the contents of the file. By default, it records on a minimal number of messages. For instructions on increasing the amount of logging, see "VLLog(5)" on page 439.

By default, the VL Server runs nine lightweight processes (LWPs). To change the number, use the -p argument.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Options

-p <*lwp processes*>

Sets the number of server lightweight processes (LWPs) to run. Provide an integer between `4` and `16`. The default is `9`.

-nojumbo

Prohibits the server from sending or receiving jumbograms. A jumbogram is a large-size packet composed of 2 to 4 normal Rx data packets that share the same header. The VL Server uses jumbograms by default, but some routers are not capable of properly breaking the jumbogram into smaller packets and reassembling them.

-enable_peer_stats

Activates the collection of Rx statistics and allocates memory for their storage. For each connection with a specific UDP port on another machine, a separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received. To display or otherwise access the records, use the Rx Monitoring API.

-enable_process_stats

Activates the collection of Rx statistics and allocates memory for their storage. A separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received, aggregated over all connections to other machines. To display or otherwise access the records, use the Rx Monitoring API.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following bos create command creates a vlserver process on the machine `fs2.abc.com` that uses six lightweight processes. Type the command on a single line:

```
% bos create -server fs2.abc.com -instance vlserver -type simple \
             -cmd "/usr/afs/bin/vlserver -p 6"
```

Privilege Required

The issuer must be logged in as the superuser `root` on a file server machine to issue the command at a command shell prompt. It is conventional instead to create and start the process by issuing the bos create command.

See Also

"BosConfig(5)" on page 409, "VLLog(5)" on page 439, "vldb.DB0(5)" on page 519, "bos_create(8)" on page 677, "bos_getlog(8)" on page 691

Copyright

voldump

Name

`voldump` — Dump an AFS volume without using the Volume Server

Synopsis

voldump -part <*partition*> -volumeid <*volume id*> [-file <*dump file*>] [-verbose] [-help]

voldump -p <*partition*> -vo <*volume id*> [-f <*dump file*>] [-ve] [-h]

Description

voldump dumps an AFS volume in the format used by vos dump without using the Volume Server. It must be run on the file server machine and usually must be run as the superuser `root` to have permissions to read the file server data. It's primary use is to recover data from a file server machine where the Volume Server cannot be started for some reason.

The dump output will go to standard output, or to a file if -file is specified. vos restore can be used to load the resulting dump into a new AFS volume. voldump always does a full dump.

Cautions

Normally, use vos dump instead of this command. voldump is a tool of last resort to try to extract data from the raw data structures stored on the file server machine and is not as regularly tested or used as the normal vos dump implementation.

If the AFS volume being dumped changes while voldump is running, the results may be inconsistent. If the File Server and Volume Server are running, stop them with bos shutdown or a similar method before running this command.

Options

-part <*partition*>

> Names the partition on which the volume to be dumped is located. voldump does not take the normal full range of ways of specifying a partition. Instead, *partition* must be either a single letter between `a` and `z`, corresponding to */vicepa* through */vicepz*, or the full path to the file server partition. `aa` is not recognized; use */vicepaa* instead.

-volumeid <*volume id*>

> Specifies the ID of the volume to dump. The volume must be specified by numeric ID, not by name.

-file <*dump file*>

Specifies the output file for the dump. If this option is not given, the volume will be dumped to standard output.

-verbose

Asks for a verbose trace of the dump process. This trace information will be sent to standard error.

Examples

The following command dumps the volume 1936964939 on the */vicepb* partition to the file */tmp/volume.dump*:

```
% voldump -part /vicepb -volumeid 1936964939 -file /tmp/volume.dump
```

Privilege Required

The issuer must have read access to the file server data stored in the specified partition. Usually, this means that the issuer must be the local superuser `root` on the file server machine.

See Also

"bos_shutdown(8)" on page 743, "volserver(8)" on page 925, "vos_dump(1)" on page 319, "vos_restore(1)" on page 369

Copyright

volinfo

Name

`volinfo` — Produces detailed statistics about AFS volume headers

Synopsis

volinfo [-online] [-vnode] [-date] [-inode] [-itime] [-part <*AFS partition name (default current partition)*>+] [-volumeid <*volume id*>+] [-header] [-sizeOnly] [-fixheader] [-saveinodes] [-orphaned] [-help]

Description

The volinfo command displays detailed statistics about one or more volume headers and the partition that houses them. The command must be issued on a file server machine and by default produces output for every volume on every AFS server partition on the machine. To display output for the volumes on one partition only, include the -part argument. To display output for one volume only, include the -volumeid argument.

Options

-online

Is nonoperational.

-vnode

Displays a table for each volume which lists the large (directory) and small (file) vnodes in it, in addition to the default output.

-date

When combined with the -vnode flag, adds the `ServerModTime` field to each vnode entry in the large vnode and small vnode tables, reporting its most recent modification time.

-inode

When combined with the -vnode flag, adds the `inode` field to each vnode entry in the large vnode and small vnode tables, reporting the associated inode number.

-itime

When combined with the -vnode flag, displays a change, modification, and access timestamp for each of the large vnode and small vnode tables.

-part <*partition name*>+

Specifies the partition that houses each volume for which to produce output. Use the format */vicepxx*, where *xx* is one or two lowercase letters. This argument can be omitted if the current working directory is the mount location for an AFS server partition; it is not

the mount location for an AFS server partition, the command produces output for every volume on all local AFS server partitions.

-volumeid <*volume id*>+

Specifies the ID number of one volume for which to produce output. The -part argument must be provided along with this one unless the current working directory is the mount location for the AFS server partition that houses the volume.

-header

Displays statistics about the volume header of each volume, in addition to the default output.

-sizeOnly

Displays a single line of output for each volume, reporting the size of various structures associated with it. The default output is suppressed and any flags that modify it (such as -vnode) are ignored.

-fixheader

Repairs damaged inodes in each volume if possible. If there are any, it reports the action it is taking to repair them. Otherwise, it produces no output in addition to the default output.

-saveinodes

Creates a file in the current working directory for each inode in each volume. Each file is called `TmpInode.`*vnode_number* and contains the inode's contents. The default output is suppressed and any flags that modify it (such as -vnode) are ignored.

-orphaned

Displays a large vnode and small vnode table for each volume, which lists only orphaned vnodes (vnodes that have no parent). If there are none, the tables are empty (only the headers appear).

-help

Prints the online help for this command. All other valid options are ignored.

Output

By default, the command produces several line of statistics for each volume. Adding other options produces or substitutes additional information as described in "OPTIONS" on page . The output is intended for debugging purposes and is meaningful to someone familiar with the internal structure of volume headers.

Privilege Required

The issuer must be logged in as the local superuser `root`.

See Also

"vldb.DB0(5)" on page 519, "volserver(8)" on page 925

Copyright

volserver

Name

`volserver` — Initializes the Volume Server component of the fs process

Synopsis

volserver [-log] [-p <*number of processes*>] [-udpsize <*size of socket buffer in bytes*>] [-enable_peer_stats] [-enable_process_stats] [-help]

Description

The volserver command initializes the Volume Server component of the `fs` process. In the conventional configuration, its binary file is located in the */usr/afs/bin* directory on a file server machine.

The volserver command is not normally issued at the command shell prompt but rather placed into a file server machine's */usr/afs/local/BosConfig* file with the bos create command. If it is ever issued at the command shell prompt, the issuer must be logged onto a database server machine as the local superuser `root`.

The Volume Server records a trace of its activity in the */usr/afs/logs/VolserLog* file. Use the bos getlog command to display the contents of the file.

The Volume Server processes the vos commands that administrators use to create, delete, move, and replicate volumes, as well as prepare them for archiving to tape or other media.

By default, the VL Server runs nine lightweight processes (LWPs). To change the number, use the -p argument.

This command does not use the syntax conventions of the AFS command suites. Provide the command name and all option names in full.

Options

-log

Records in the /usr/afs/logs/VolserLog file the names of all users who successfully initiate a vos command. The Volume Server also records any file removals that result from issuing the vos release command with the -f flag.

-p <*number of processes*>

Sets the number of server lightweight processes (LWPs) to run. Provide an integer between `4` and `16`. The default is `9`.

-udpsize <*size of socket buffer*>

Sets the size of the UDP buffer in bytes, which is 64 KB by default. Provide a positive integer, preferably larger than the default.

-enable_peer_stats

Activates the collection of Rx statistics and allocates memory for their storage. For each connection with a specific UDP port on another machine, a separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received. To display or otherwise access the records, use the Rx Monitoring API.

-enable_process_stats

Activates the collection of Rx statistics and allocates memory for their storage. A separate record is kept for each type of RPC (FetchFile, GetStatus, and so on) sent or received, aggregated over all connections to other machines. To display or otherwise access the records, use the Rx Monitoring API.

-help

Prints the online help for this command. All other valid options are ignored.

Examples

The following bos create command creates a `volserver` process on the machine `fs2.abc.com`:

```
% bos create -server fs2.abc.com -instance volserver -type simple \
             -cmd /usr/afs/bin/volserver
```

Privilege Required

The issuer must be logged in as the superuser `root` on a file server machine to issue the command at a command shell prompt. It is conventional instead to create and start the process by issuing the bos create command.

See Also

"BosConfig(5)" on page 409, "VolserLog(5)" on page 441, "bos_create(8)" on page 677, "bos_getlog(8)" on page 691, "vos(1)" on page 289

Copyright

xfs_size_check

Name

`xfs_size_check` — Verifies proper IRIX inode configuration

Synopsis

xfs_size_check

Description

The xfs_size_check command, when run on a file server machine that runs IRIX version 6.2 or higher and uses XFS-formatted partitions as server partitions (conventionally mounted at */vicep* directories), verifies that each partition uses 512-byte inodes. AFS stores information in the inodes on server partitions, and the 256-byte inode size that XFS uses by default is not large enough.

Cautions

This command is available on in the AFS distribution for IRIX system types that can use XFS-formatted partitions as server partitions.

Output

If all server partitions are properly configured, the command produces no output. Otherwise, it prints the following header:

```
Need to remake the following partitions:
```

and then the following message for each partition on which to run the IRIX mkfs command with the indicated options:

```
&lt;device&gt;: mkfs -t xfs -i size=512 -l size=4000b &lt;device&gt;
```

where <device> is in a format like `/dev/dsk/dks0d0s0` for a single disk partition or `/dev/xlv/xlv0` for a logical volume.

Privilege Required

The issuer must be logged in as the local superuser `root`.

Copyright