AFS

Release Notes

Version 3.6

GC09-4558-00

AFS

Release Notes

Version 3.6

GC09-4558-00

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 93.

First Edition (April 2000)

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.

Order publications through your IBM representative or through the IBM branch office serving your locality.

Contents

About These Release Notes

This section describes the purpose, organization, and conventions used in this document.

Audience and Purpose

This document describes new features, limitations and requirements of the AFS® 3.6 General Availability (GA) release. It assumes that the reader is familiar with administration of AFS 3.5 and of the supported operating systems.

Organization of the Document

This document has the following sections:

Related Documents

The following documents are also 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 Administration Reference

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

The *IBM AFS Administration Reference* lists AFS files and commands in alphabetical order. The reference page for each command specifies its syntax, including the acceptable aliases and abbreviations. It then describes the command's function, arguments, and output if any. Examples and a list of related commands are provided, as are warnings where appropriate.

This manual complements the *IBM AFS Administration Guide*: it does not include procedural information, but describes commands in more detail than the *IBM AFS Administration Guide*.

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® system administrator.

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

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.

For further information on the syntax and input rules for AFS commands, see the appendix to the *IBM AFS Administration Guide* or the **afs_intro** reference page in the *IBM AFS Administration Reference*.

AFS 3.6 Release Notes

This file documents new features, upgrade procedures, and remaining limitations associated with the initial General Availability (GA) release of AFS® 3.6 (build level **afs3.6 2.0**).

Note: This document includes all product information available at the time the document was produced. For additional information that became available later, see the **README.txt** file included on the AFS CD-ROM.

Summary of New Features

AFS 3.6 includes the following new features.

- Support for the 64–bit version of Solaris 7.
- Support for the 64–bit version of HP-UX 11.0.
- The HP-UX 11.0 File Server uses the POSIX-compliant threading package provided with HP-UX. (Other supported operating systems started using native threads in AFS 3.5.) See "Product Notes for HP-UX Systems" on page 10.
- There is a single edition of AFS 3.6, instead of separate United States and international editions as in previous releases. The United States government now permits export outside North America of the encryption software that the Update Server uses to protect user-level data. With AFS 3.6, cells outside North America can run a system control machine to distribute the contents of the **/usr/afs/etc** directory among server machines.

  The AFS 3.6 distribution includes a single CD-ROM for each system type, which contains all AFS software. There is no CD-ROM labeled **Encryption Files** or **Domestic Edition** in the AFS 3.6 distribution.

  Because they were produced before the change in export restrictions, the *IBM AFS Administration Guide* and *IBM AFS Administration Reference* still distinguish between United States and international editions of AFS. However, AFS customers in any country can ignore the distinction and use the United States instructions if they choose.
- Support for volumes up to 8 GB in size. In previous versions of AFS, the limit was 2 GB.

  Note that smaller volumes are still more practical than large ones in general. The larger a volume, the longer it takes to move or clone it, which introduces greater potential for an outage to halt the operation before it completes.

- Support for backing up AFS data to the Tivoli Storage Manager (TSM), formerly called the ADSTAR Distributed Storage Manager (ADSM). TSM implements the Open Group's Backup Service application programming interface (API), also called XBSA. Support for additional XBSA-compliant programs in future releases of AFS is possible. See "Support for Backup to TSM" on page 21.
- A new command and new options to existing commands. See "Changes to AFS Commands, Files, and Functionality" on page 19.

Supported System Types

AFS supports the following system types.

alpha_dux40	DEC AXP system with one or more processors running Digital UNIX 4.0d, 4.0e, or 4.0f
hp_ux110	Hewlett-Packard system with one or more processors running the 32-bit or 64-bit version of HP-UX 11.0
i386_linux22	IBM-compatible PC with one or more processors running Linux kernel version 2.2.5-15 (the version in Red Hat Linux 6.0), 2.2.10, 2.2.12, 2.2.12-20 (the version in Red Hat Linux 6.1), 2.2.13, or 2.2.14
rs_aix42	IBM RS/6000 with one or more 32-bit or 64-bit processors running AIX 4.2, 4.2.1, 4.3, 4.3.1, 4.3.2, or 4.3.3
sgi_65	Silicon Graphics system with one or more processors running IRIX 6.5 or 6.5.4. Support is provided for the following CPU board types, as reported by the IRIX **uname -m** command: IP19, IP20, IP21, IP22, IP25, IP26, IP27, IP28, IP30, IP32
sun4x_56	Sun SPARCstation with one or more processors running Solaris 2.6
sun4x_57	Sun SPARCstation with one or more processors running the 32-bit or 64-bit version of Solaris 7

Hardware and Software Requirements

For a list of requirements for both server and client machines, see the chapter titled *Installation Overview* in the *IBM AFS Quick Beginnings* document.

Accessing the AFS Binary Distribution and Documentation

The AFS Binary Distribution includes a separate CD-ROM for each supported operating system, containing all AFS binaries and files for both server and client machines, plus the documentation set in multiple formats. At the top level of the CD-ROM is a directory called **Documentation** plus a directory containing the system-specific AFS binaries, named using the values listed in "Supported System Types". The CD-ROM for some operating systems has more than one system-specific directory; for example, the Solaris CD-ROM has **sun4x_56** and **sun4x_57**.

The instructions in "Upgrading Server and Client Machines to AFS 3.6" on page 27 specify when to mount the CD-ROM and which files or directories to copy to the local disk or into an AFS volume.

The documents are also available online at **http://www.transarc.com/Library/documentation/afs_doc.html**. The documentation set includes the following documents:

- *IBM AFS Release Notes* (this document)
- *IBM AFS Administration Guide* (called *AFS System Administrator's Guide* in previous releases)
- *IBM AFS Administration Reference* (called *AFS Command Reference Manual* in previous releases)
- *IBM AFS Quick Beginnings* (called *AFS Installation Guide* in previous releases)
- *IBM AFS User Guide* (called *AFS User's Guide* in previous releases)

Documents are provided in the following formats:

- HTML, suitable for online viewing in a Web browser or other HTML viewer
- PDF, suitable for online viewing or for printing using the Acrobat Reader program from Adobe

If you do not already have the Acrobat Reader program, you can download it for free at **http://www.adobe.com/products/acrobat/readstep.html**.

Adobe provides only an English-language version of Acrobat Reader for UNIX platforms. The program can display PDF files written in any language. It is the program interface (menus, messages, and so on) that is available in English only.

To make Reader's interface display properly in non-English language locales, use one of two methods to set the program's language environment to English:

- Set the LANG environment variable in the Reader initialization script. The advantage of this method is that it ensures proper behavior even when Reader is launched by other applications, such as a browser or an application's Help menu. Editing the script usually requires local superuser **root** privilege, however.

  If your Reader distribution includes the script, it is installed by convention as *AcroRead_Dir***/bin/acroread**, where *AcroRead_Dir* is the installation directory for Reader files.

  Add the following line to the script, directly after the `#!/bin/sh` statement:

  ```
  LANG=C; export LANG
  ```

- Set the LANG environment variable to the value **C** in the command shell before starting the Reader program. The following is the appropriate command for some shells.

  ```
  % setenv LANG C
  ```

  Note that this setting affects all programs started in the command shell, with possibly undesirable results if they also use the LANG variable. The preceding method affects Reader only.

Product Notes

The following sections summarize limitations and requirements that pertain to all system types and to individual system types, and describe revisions to the AFS documents:

Product Notes for All System Types

- **Limit on number of file server machine interfaces**

  AFS supports up to 15 addresses on a multihomed file server machine. If more interfaces are configured with the operating system, AFS uses only the first 15.

- **Limit on number of client machine interfaces**

  AFS supports up to 32 addresses on a multihomed client machine. Do not configure more interfaces.

- **Limit on number of AFS server partitions**

  AFS supports up to 256 server (**/vicep**) partitions on a file server machine. This corresponds to directory names **/vicepa** through **/vicepz** and **/vicepaa** through **/vicepiv**.

- **Limit on number of file server machines**

  The VLDB can store up to 255 server entries, each representing one file server machine (single- or multihomed). This effectively determines the maximum number of file server machines in the cell. To make room in the VLDB for new server entries, use the **vos changeaddr** command's **-remove** argument to remove the entries for decommissioned file server machines.

- **Limit on file size**

AFS supports a maximum file size of 2 GB.

- **Limit on volume and partition size**

  AFS supports a maximum volume size of 8 GB. In AFS version 3.5 and earlier, the limit is 2 GB. There is no limit on partition size other than the one imposed by the operating system.

- **Limit on cache size**

  AFS supports a maximum disk cache size of 1 GB. In AFS version 3.1 and earlier, the limit is 700 MB.

- **Limit on number of File Server threads**

  The File Server (**fileserver** process) can use up to 128 threads, unless the operating system imposes a lower limit. Testing for the AFS 3.6 GA release indicates that HP-UX sometimes imposes a lower limit, depending on the resources available on a machine. See "Product Notes for HP-UX Systems" on page 10.

  The File Server always reserves seven threads for special uses, so the maximum effective value for the **fileserver** command's **-p** argument is seven less than the actual limit. On most systems, the effective maximum is therefore **121**.

- **Limit on number of volume site definitions**

  The VLDB entry for a volume can accommodate a maximum of 13 site definitions. 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 partition as the read/write site counts as a separate site).

- **No support for VxFS as server or cache partition**

  AFS does not support use of the VxFS file system as either a client cache partition or server (**/vicep**) partition. It is acceptable to use both VxFS and AFS on the same machine, but the cache partition and all AFS server partitions must use a supported file system type such as UFS. See the following sections of this document for similar restrictions affecting particular operating systems.

- **Run same version of a server process on all server machines**

  For predictable performance, run the same version of an AFS server process on all server machines in a cell. For example, if you upgrade the Volume Location Server process on a database server machine to AFS 3.6, you must upgrade it on all of them. The upgrade instructions in "Upgrading Server and Client Machines to AFS 3.6" on page 27 have you upgrade the binaries for all server processes on all machines to the same version, and in general that is the best policy. Unless otherwise noted, it is acceptable to run different build levels of a major version on different machines (for example, AFS 3.5 build 3.0 on one machine and AFS 3.5 build 3.11 on another).

- **Single edition of AFS for all countries**

There is a single edition of AFS 3.6 for both North American and international customers. For details, see “Summary of New Features” on page 1.

- **TSM is the supported XBSA server**

  The AFS 3.6 Backup System can communicate with one XBSA server, the Tivoli Storage Manager (TSM). There are several requirements and limitations associated with its use, as detailed in “Product Notes for Use of TSM” on page 23.

- **Use Netscape 4.0 or higher**

  If using a Netscape browser to read the HTML version of an AFS document, use version 4.0 or higher. Some fonts used in the documents possibly do not display properly in earlier versions.

- **Set the Acrobat Reader environment to English**

  The user interface to the Adobe Acrobat Reader program for displaying PDF files works correctly only when the program’s language environment is set to English. Users in non-English language locales probably need to adjust the language setting. See “Accessing the AFS Binary Distribution and Documentation” on page 2.

- **No support for IPv6**

  AFS does not support version 6 of the Internet Protocol (IPv6). You must continue to specify the IPv4 protocol names `udp` and `tcp` in the entries for AFS-modified services in the **inetd** configuration file, rather than the IPv6 names `upd6` and `tcp6`. If you use the IPv6 version, the AFS-modified **inetd** daemon cannot locate the service and does not open the service’s port.

  The **inetd** configuration file included with some operating system revisions possibly specifies IPv6 protocols by default. You must modify or replace the file in order to use the AFS–modified version of remote services.

- **Limit on directory size when element names are long**

  If the name of every file system element (file, link, or subdirectory) in a directory is 16 characters or more, then when there are about 31,700 elements it becomes impossible to create any more elements with long names. It is still possible to create elements with names shorter than 16 characters. This limitation is due to the way AFS implements directories. For a more detailed explanation, contact your AFS product support representative.

- **Setting setuid or setgid bit on file fails silently**

  Only members of the **system:administrators** group can turn on the setuid or setgid mode bit on an AFS file or directory. However, AFS generates an error message only when a regular user attempts to set the bit on a directory. Attempts on a file fail silently.

- **The add instruction in the uss bulk input file does not work as documented**

The documentation specifies the following syntax for creating an authentication-only account (entries in the Authentication and Protection Databases only) by using an **add** instruction in the **uss** bulk template file:

```
add username[:]
```

However, you must in fact follow the *username* value with two colons for the **uss bulk** command to create the account:

```
add username::
```

- **Running the backup savedb command blocks other Backup System operations**

  The Backup Server locks the Backup Database as it performs the **backup savedb** command, which can take a long time. Because other backup operations cannot access the database during this time, they appear to hang. Avoid running other backup operations after issuing the **backup savedb** command.

  Actually, this limitation applies to any operation that locks the Backup Database for a significant amount of time, but most other operations do not. In any case, running the **backup savedb** command is appropriate only in the rare case when the Backup Database is corrupted, so this limitation usually does not have a significant impact.

- **NFS/AFS Translator sometimes performs poorly under heavy load**

  The NFS/AFS Translator does not always perform well under heavy load. Sometimes the translator machine hangs, and sometimes NFS client machines display the following error message.

  ```
  NFS Stale File Handle
  ```

- **Sample files for package program not included**

  The AFS distribution does not include the sample files referred to in the chapter of the *IBM AFS Administration Guide* about the **package** program (the files formerly installed by convention in the **etc**, **lib**, and **src** subdirectories of the **/afs/***cellname***/wsadmin** directory). *IBM AFS Quick Beginnings* therefore does not include instructions for installing the sample files. If you wish to use the **package** program and the discussion in the *IBM AFS Administration Guide* is not sufficient to guide you, contact your AFS product support representative for assistance.

Product Notes for AIX Systems

- **The klog command's -setpag flag is supported on AIX 4.2.1 and 4.3.3 only**

  To use the **klog** command's **-setpag** flag, you must install the indicated AIX APAR (Authorized Program Analysis Report), available from IBM, on a machine running the indicated AIX version:

  – APAR IY07834 on AIX 4.2.1 machines

  – APAR IY07835 on AIX 4.3.3 machines

  To determine if the APAR is installed, issue the following command:

```
% instfix -i -k APAR_identifier
```

IBM provides an APAR for the indicated (latest) AIX versions only. Therefore, the **-setpag** flag does not work correctly on machines running the base level of AIX 4.2 or 4.3, or AIX 4.3.1 or 4.3.2.

- **Change to AFS installation procedure for AIX 4.3.3**

  If version 4.3.3.0 or higher of the AIX **bos.rte.security** fileset is installed (usually true on a machine using the AIX 4.3.3 kernel), you must modify the procedure documented in *IBM AFS Quick Beginnings* for enabling integrated AFS login. Instead of editing the **/etc/security/login.cfg** file, you edit the **/usr/lib/security/methods.cfg** file.

  To determine which version of the **bos.rte.security** fileset is installed, issue the following command:

  ```
  # lslpp -L bos.rte.security
  ```

  The change affects Step 3 in the section titled *Enabling AFS Login on AIX Systems* in each of two chapters in *IBM AFS Quick Beginnings*: *Installing the First AFS Machine* and *Installing Additional Client Machines*. For the complete text of the modified step, see "Documentation Notes" on page 15.

- **No support for the NFS/AFS Translator with base level of AIX 4.2**

  AFS does not support the use of machines running the base level of AIX 4.2 as NFS/AFS Translator machines. The AFS distribution does not include the required kernel extensions file, formerly installed by convention as **/usr/vice/etc/dkload/afs.ext.trans**. Do not set the NFS variable to the value **$NFS_NFS** in the AFS initialization script (by convention, **/etc/rc.afs**).

  Machines running AIX 4.2.1 and higher are supported as NFS/AFS Translator machines. They use the **afs.ext.iauth** kernel extensions file instead.

- **NFS/AFS Translator cannot coexist with NFS/DFS Gateway**

  A machine running AIX 4.2.1 or higher cannot act as both an NFS/AFS Translator and a NFS/DFS Gateway Server at the same time, because both translation protocols must have exclusive access to the AIX **iauth** interface. An attempt by either file system to access the **iauth** interface when the other file system is already using it fails with an error message.

- **No support for NFS Version 3 software on NFS clients**

  Do not run NFS Version 3 software on NFS client machines that use an NFS/AFS Translator machine running AIX. The NFS3 client software uses the **readdir+** NFS command on directories, which can cause excessive volume lookups on the translator machine. This can lead to timeouts, especially when used in the **/afs** directory or other directories with many volume mount points. Use NFS Version 2 instead.

- **No support for Large File Enabled Journalled File System as AFS server partition**

AFS does not support use of AIX's Large File Enabled Journalled File System as an AFS server (**/vicep**) partition. If you configure a partition that uses that file system as an AFS server partition, the File Server ignores it and writes the following message to the **/usr/afs/logs/FileLog** file:

```
/vicepxx is a big files filesystem, ignoring it
```

AFS supports use of the Large File Enabled Journalled File System as the cache partition on a client machine.

- **PASSWORD_EXPIRES variable not set on AIX**

  The AIX secondary authentication system does not support setting the PASSWORD_EXPIRES environment variable during login.

- **The chuser, chfn and chsh commands are inoperative**

  The **chuser**, **chfn**, and **chsh** commands are inoperative on AFS machines running AIX. AFS authentication uses the AIX secondary authentication system, and sets the `registry` variable in the **/etc/security/user** file to `DCE` for the default user. That is, the setting is

  ```
  registry = DCE
  ```

  as described in the sections of *IBM AFS Quick Beginnings* that discuss enabling AFS login on AIX systems. However, when the `registry` variable has any value other than `registry = files`, AIX does not allow edits to **/etc/passwd** and related files, and so disallows the **chuser**, **chfn** and **chsh** commands. Attempts to edit entries by running these commands on the command line result in error messages like the following.

  – From the **chuser** command:

    ```
    You can only change the HOME directory on the name server.
    ```

  – From the **chfn** command:

    ```
    You can only change the User INFORMATION on the name server.
    ```

  – From the **chsh** command:

    ```
    You can only change the Initial PROGRAM on the name server.
    ```

  From within SMIT, using the **chuser** function results in an error message like the following:

  ```
  3004-716: You can only change the HOME directory on the name server
  ```

  It is not possible for AFS Development to alter this behavior, because AIX imposes the restriction. Sites that wish to run these commands must develop a solution appropriate for their needs.

Product Notes for Digital UNIX Systems

- **No support for the NFS/AFS Translator**

  AFS does not support use of Digital UNIX machines as NFS/AFS Translator machines.

- **No support for AdvFS as server or cache partition**

  AFS does not support use of Digital UNIX's Advanced File System (AdvFS) as either a client cache partition or a server (**/vicep**) partition. It is acceptable to use both AdvFS and AFS on the same machine, but the cache partition and all AFS server partitions must be UFS partitions.

- **No support for real-time kernel preemption or related lock modes**

  AFS does not function correctly on a Digital UNIX machine when real-time preemption of system calls is enabled in the kernel. Do not enable this feature in any manner, including the following:

  – By including the following statement in the **/usr/sys/conf/AFS** file:

    ```
    options RT_PREEMPT_OPT
    ```

  – By including either of the following instructions in the **/etc/sysconfigtab** file:

    ```
    rt_preempt_opt=1
    rt-preempt-opt=1
    ```

  Also, AFS does not function correctly when the value of the kernel **lockmode** option is other than **0** (zero, the default) or **2**. Lock mode values **1**, **3**, and **4** are unsupported because they imply that real-time preemption is enabled (indeed, enabling real-time preemption sets the lock mode to **1** automatically).

- **Building AFS from source requires /usr/sys/AFS directory**

  Building AFS from source for Digital UNIX requires that certain header files (such as **cpus.h**) reside in the local **/usr/sys/AFS** directory. This directory exists only if you have previously incorporated AFS modifications into the kernel of the machine on which you are performing the compilation. Otherwise, the required header files reside only in the local directory called **/usr/sys/***machine_name*.

  If the **/usr/sys/AFS** directory does not exist, issue the following command to create it as a link:

  ```
  # ln -s  /usr/sys/machine_name  /usr/sys/AFS
  ```

  When the compilation is complete, remove the link.

Product Notes for HP-UX Systems

- **No support for the NFS/AFS Translator**

  AFS does not support use of HP-UX 11.0 machines as NFS/AFS Translator machines.

- **Upgrade kernel extensions when upgrading the File Server**

  The AFS 3.6 version of the File Server uses the native HP-UX threading package. When upgrading to the new File Server on a machine that previously ran File Server version 3.5 or earlier, you must also upgrade the AFS kernel extensions to the AFS 3.6 version.

For instructions on upgrading server and client machines, see "Upgrading Server and Client Machines to AFS 3.6" on page 27.

- **Possible lower limit on number of File Server threads**

  On some machines, HP-UX reduces the number of threads available to the File Server to fewer than the AFS default of 128. To determine the maximum number of threads available to the File Server (or any single process) on an HP-UX machine, issue the following command:

  ```
  % getconf _SC_THREAD_THREADS_MAX
  ```

  As on other system types, the HP-UX File Server reserves seven threads for special uses, so the maximum effective value for the **fileserver** command's **-p** argument is seven less than the number reported by the **getconf** command.

- **PAM can succeed inappropriately when pam_dial_auth module is optional**

  For AFS authentication to work correctly for a service, all entries for the service in the HP-UX PAM configuration file (**/etc/pam.conf** by convention) must have the value `optional` in the third field, as specified in *IBM AFS Quick Beginnings*. However, when you make the **login** entry that invokes the **pam_dial_auth** module optional in this way, it can mean that PAM succeeds (the user can login) even when the user does not meet all of the **pam_dial_auth** module's requirements. This is not usually considered desirable.

  If you do not use dial-up authentication, comment out or remove the entry for the **login** service that invokes the **pam_dial_auth** module. If you do use dial-up authentication, you must develop a configuration that meets your needs; consult the HP-UX documentation for PAM and the **pam_dial_auth** module.

- **HP patch PHCO_18572 enables PAM to change to home directory**

  You must install Hewlett-Packard patch PHCO_18572 to enable HP-UX's standard PAM to change to a user's home directory during login. The patch is accessible for download via the UNIX File Transfer Protocol (**ftp**) at the following address:

  ftp://hpatlse.atl.hp.com/hp-ux_patches/s700_800/11.X/PHCO_18572

  The patch is also available from HP Electronic Support Centers at the following URLs.

  – In the Americas and Asia Pacific: **http://us-support.external.hp.com**

  – In Europe: **http://europe-support.external.hp.com**

Product Notes for IRIX Systems

- **kdump program does not work with dynamically loaded kernels**

The AFS kernel dump program, **kdump**, cannot retrieve kernel information from an IRIX system on which the dynamic kernel loader, **ml**, was used to load AFS extensions. The **kdump** program can read only static kernels into which AFS is built.

- **No AFS-modified remote commands**

  The AFS distribution for IRIX machines does not include AFS-modified versions of any of the remote (**r***) commands except **inetd.afs**. Silicon Graphics has already modified the IRIX versions of the remote commands to be compatible with AFS.

- **Do not run the fsr program**

  Do not run the IRIX File System Reorganizer (**fsr** program) on a client cache partition (**/usr/vice/cache** directory or equivalent) or AFS server partition (**/vicep** directory). The program can corrupt or remove AFS data.

- **The timed daemon runs by default**

  The IRIX 6.5 distribution includes and starts the **timed** time-synchronization daemon by default. If you want to use the **runntp** program and the Network Time Protocol Daemon (NTPD) on AFS server machines, as documented in *IBM AFS Quick Beginnings*, issue the following commands. They disable the **timed** daemon and remove it from the machine's startup sequence.

  ```
  # /etc/chkconfig -f timed off

  # /sbin/killall timed
  ```

- **Default login program does not grant AFS tokens**

  The IRIX 6.5 distribution includes the **clogin** program as the default login utility. This graphical utility does not grant AFS tokens. If you want your users to obtain tokens during login, you must disable the **clogin** program and substitute either the standard command-line **login** program or the **xdm** graphical login utility, both of which grant AFS tokens if AFS modifications have been incorporated into the kernel. Issue the following command to disable the **clogin** program.

  ```
  # /etc/chkconfig -f visuallogin off
  ```

Product Notes for Linux Systems

- **Supported kernel versions**

  The General Availability release of AFS 3.6 supports Red Hat Software's Linux 6.0 (which incorporates kernel version 2.2.5-15) and Linux 6.1 (which incorporates kernel version 2.2.12–20). The distribution also includes AFS kernel extensions for kernel versions 2.2.10, 2.2.12, 2.2.13, and 2.2.14. The AFS initialization script included in the AFS 3.6 distribution automatically selects the appropriate kernel extensions for the kernel version in use on the local machine.

  Red Hat Linux 6.0 and 6.1 include a compiled kernel, but for the other supported kernel versions you must obtain kernel source and compile the

kernel yourself. In this case, you must use version 2.7.2.3 or higher of the **gcc** program, which is part of the Linux distribution. Do not use other compilers.

The Linux kernel-building tools by default create a symmetric multiprocessor (SMP) kernel, which can run on both uniprocessor and multiprocessor machines. However, a uniprocessor machine generally performs best with a uniprocessor kernel.

You can obtain Linux kernel source via the UNIX File Transfer Protocol (**ftp**) at **ftp.kernel.org** or one of its mirror sites. There is also kernel upgrade information at **http://www.kernel.org**.

- **AFS requires libc6**

  For correct AFS performance, the operating system must use the C library called **libc6** (or **glibc2**), rather than **libc5** (**glibc1**).

- **Modified insmod program required with some kernels**

  If using an SMP kernel or a uniprocessor kernel configured to use more than 1 GB of memory, you must use a modified version of the **insmod** program. You do not need the modified program if using a standard uniprocessor kernel.

  You can download the modified **insmod** program at the following URLs:

  – **http://www.transarc.com/Support/afs/index.html**. See the **Downloads** section of the page. To comply with the GNU Public License (GPL), the download site also makes available the complete modified **insmod.c** source file and a source-code patch against the original **insmod.c** file.

  – **http://www.pi.se/blox/modutils/index.html**. Select the file listed at the top of the index. This is a site for Linux **modutils** source code.

- **No support for the NFS/AFS Translator**

  AFS does not support the use of Linux machines as NFS/AFS Translator machines.

- **No AuthLog database**

  The Authentication Server running on a Linux machine creates and writes messages to the **/usr/afs/logs/AuthLog** file, just as on other system types. However, it does not create or use the two files which constitute the auxiliary AuthLog database on other system types (**AuthLog.dir** and **AuthLog.pag**). The **kdb** command is therefore inoperative on Linux machines. The auxiliary database is useful mostly for debugging and is not required for normal operations.

- **Curses utility required for monitoring programs**

  For the **afsmonitor**, **scout** and **fms** programs to work properly, the dynamic library **/usr/lib/libncurses.so** must be installed on the machine. It is available in most Linux distributions.

Product Notes for Solaris Systems

- **Different location for 64–bit Solaris 7 kernel extensions**

  As noted in "Upgrading Server and Client Machines to AFS 3.6" on page 27, the 64–bit version of Solaris 7 uses a different location for kernel extension library files than previous versions of Solaris: **/kernel/fs/sparcv9/afs**. The 32–bit version of Solaris 7 uses the same location as Solaris 2.6, **/kernel/fs/afs**.

- **SunSoft Patch 106541 for Solaris 7 replaces the /sbin/mountall script**

  As part of replacing the standard **fsck** program on an AFS file server machine that runs Solaris, you make two changes in the **/sbin/mountall** script. If you use Solaris 7 and apply SunSoft Patch 10654, it replaces the **/sbin/mountall** script. This has two implications:

  1. If you apply the patch on an existing file server machine, the changes you already made in the **/sbin/mountall** script are overwritten. You must make the changes again in the new (replacement) script.
  2. In the replacement script, the appearance of one of the sections of code that you must alter is different than in the original script and as specified in *IBM AFS Quick Beginnings*.

  For more details, see "Documentation Notes" on page 15.

- **PAM can succeed inappropriately when pam_dial_auth module is optional**

  For AFS authentication to work correctly for a service, all entries for the service in the Solaris PAM configuration file (**/etc/pam.conf** by convention) must have the value `optional` in the third field, as specified in *IBM AFS Quick Beginnings*. However, when you make the **login** entry that invokes the **pam_dial_auth** module optional in this way, it can mean that PAM succeeds (the user can login) even when the user does not meet all of the **pam_dial_auth** module's required conditions. This is not usually considered desirable.

  If you do not use dial-up authentication, comment out or remove the entry for the **login** service that invokes the **pam_dial_auth** module. If you do use dial-up authentication, you must develop a configuration that meets your needs; consult the Solaris documentation for PAM and the **pam_dial_auth** module.

  The AFS Development group has filed a Request for Enhancement (RFE #4122186) with SunSoft for a design change that eliminates this problem with the **pam_dial_auth** module. There is no projected solution date. For further information, contact your AFS product support representative.

- **Solaris 2.6 patches are required for CDE**

  There were several defects in the initial release of the Solaris 2.6 implementation of the Common Desktop Environment (CDE). They prevented integrated AFS login from working consistently under CDE.

SunSoft now provides patches that correct the problems. You must install them in order to obtain support for use of CDE from your AFS product support representative.

- If using version 1.2 of the Solaris CDE, install SunSoft patches 105703-03 and 106027-01 (or later revisions).
- If using version 1.3 of the Solaris CDE, which is included on the SDE CD-ROM, install SunSoft patch 106661–04 (or a later revision).

Use the following command to determine which version of CDE you are running:

```
% pkginfo -l SUNWdtdte
```

Documentation Notes

- **Instructions for international edition of AFS are obsolete**

  As noted in "Summary of New Features" on page 1, the *IBM AFS Administration Guide* and *IBM AFS Administration Reference* distinguish between United States and international editions of AFS, because the documents were produced before a relaxation of United States government export restrictions. AFS 3.6 includes just one edition. AFS customers in any country can ignore the documented distinction between editions and use the United States instructions if they choose.

- **Clarification on obtaining technical support**

  The AFS documents refer you to the *AFS Product Support group* for technical assistance with AFS problems and questions. This is intended to be a generic term. To learn how to obtain technical support, consult your AFS license agreement or other materials from your AFS vendor.

- **Change to *IBM AFS Quick Beginnings* instructions for enabling AFS login on AIX machines**

  If version 4.3.3.0 or higher of the AIX **bos.rte.security** fileset is installed (usually true on a machine using the AIX 4.3.3 kernel), edit the **/usr/lib/security/methods.cfg** file instead of the **/etc/security/login.cfg** file as documented in *IBM AFS Quick Beginnings*.

  The change affects Step 3 in the section titled *Enabling AFS Login on AIX Systems* in each of two chapters in *IBM AFS Quick Beginnings*: *Installing the First AFS Machine* and *Installing Additional Client Machines*. The corrected text follows.

  Create or edit the `DCE` and `AFS` stanzas in one of two files on the local disk:

  - The **/usr/lib/security/methods.cfg** file, if version 4.3.3.0 or higher of the AIX **bos.rte.security** fileset is installed on the machine (usually true on a machine using the AIX 4.3.3 kernel)
  - The **/etc/security/login.cfg** file, if an earlier version of the fileset is installed

  Edit the stanzas as follows:

– In the DCE stanza, set the program attribute as indicated.

  If you use the AFS Authentication Server (**kaserver** process):

```
DCE:
     program = /usr/vice/etc/afs_dynamic_auth
```

  If you use a Kerberos implementation of AFS authentication:

```
DCE:
     program = /usr/vice/etc/afs_dynamic_kerbauth
```

– In the AFS stanza, set the program attribute as indicated.

  If you use the AFS Authentication Server (**kaserver** process):

```
AFS:
     program = /usr/vice/etc/afs_dynamic_auth
```

  If you use a Kerberos implementation of AFS authentication:

```
AFS:
     program = /usr/vice/etc/afs_dynamic_kerbauth
```

- **Change to** *IBM AFS Quick Beginnings* **instructions for replacing Solaris fsck program**

  In two sections of *IBM AFS Quick Beginnings*, there are instructions for editing the **/sbin/mountall** script on Solaris machines as part of replacing the standard **fsck** program. The two sections are *Configuring the AFS-modified fsck Program on Solaris Systems* in the chapter about the first AFS machine and *Getting Started on Solaris Systems* in the chapter about additional server machines.

  If you use Solaris 7 and apply SunSoft Patch 10654, it replaces the **/sbin/mountall** script. In the replacement script, the appearance of one of the sections of code that you must alter is different than in the original script and as specified in *IBM AFS Quick Beginnings*, which is as follows:

```
# For fsck purposes, we make a distinction between ufs and
# other file systems
#
if [ "$fstype" = "ufs" ]; then
     ufs_fscklist="$ufs_fscklist $fsckdev"
     saveentry $fstype "$OPTIONS" $special $mountp
     continue
fi
```

  In the replacement script, the code is instead similar to the following:

```
# For fsck purposes, we make a distinction between ufs and
# other file systems.  Here we check that the file system is
# not mounted using fsck -m before adding it to the list to
# be checked
#
if [ "$fstype" = "ufs" ]; then
     /usr/sbin/fsck -m -F $fstype $fsckdev >/dev/null 2>&1
     if [ $? != 33 ]; then
             ufs_fscklist="$ufs_fscklist $fsckdev"
```

```
                saveentry $fstype "$OPTIONS" $special $mountp
                continue
        else
                echo "$fsckdev already mounted"
                continue
        fi
fi
```

You still need to change the first `if` statement (the one directly after the comment) to check for both the UFS and AFS file system types, as specified in *IBM AFS Quick Beginnings*:

```
if [ "$fstype" = "ufs" -o "$fstype" = "afs" ]; then
```

- **Correction to** *IBM AFS Quick Beginnings* **instructions for accessing AFS documents**

  The section of *IBM AFS Quick Beginnings* titled *Storing AFS Documents in AFS* (in the chapter about the first AFS machine) incorrectly describes the organization of the top-level **Documentation** directory on the AFS CD-ROM. It states that there is a subdirectory for each document format. Instead, there is a subdirectory for each language in which the documents are available, named using the following codes:

  de_DE for German

  en_US for United States English

  es_ES for Spanish

  ko_KR for Korean

  pt_BR for Brazilian Portuguese

  zh_CN for Simplified Chinese

  zh_TW for Traditional Chinese

  In each language directory is a subdirectory for each available document format. In each format directory is a subdirectory for each document. For example, the path on the CD-ROM to the English-language HTML version of the *IBM AFS Quick Beginnings* is **Documentation/en_US/HTML/QkBegin**.

  Note: Not all documents are available in every language, as determined by the IBM translation center responsible for each language. All documents are available in English.

  Assuming that you want to install the documentation for one language only, substitute the following text for Step 5 in the instructions in *Storing AFS Documents in AFS*:

Copy the AFS documents in one or more formats from the CD-ROM into subdirectories of the **/afs/***cellname***/afsdoc** directory. Repeat the commands for each format.

```
# mkdir format_name

# cd format_name

# cp -rp /cdrom/Documentation/language_code/format  .
```

If you choose to store the HTML version of the documents in AFS, note that in addition to a subdirectory for each document there are several files with a **.gif** extension, which enable readers to move easily between sections of a document. The file called **index.htm** is an introductory HTML page that contains a hyperlink to each of the documents. For online viewing to work properly, these files must remain in the top-level HTML directory (the one named, for example, **/afs/***cellname***/afsdoc/HTML**).

- **Revised reference page for NetRestrict files**

  The *IBM AFS Administration Guide* and *IBM AFS Administration Reference* incorrectly state that the value **255** acts as a wildcard in IP addresses that appear in the **NetRestrict** file (client or server version). Wildcarding does not work and is not supported. For corrected documentation, see "NetRestrict (client version)" on page 71 and "NetRestrict (server version)" on page 72.

- **Revised reference pages for backup commands and configuration file**

  The *IBM AFS Administration Guide* and *IBM AFS Administration Reference* do not document the interoperation of the AFS Backup System and the Tivoli Storage Manager (TSM), because support for TSM was added after the documents were produced.

  For a complete description of the new TSM-related features and configuration procedures, see "Support for Backup to TSM" on page 21 and the indicated reference pages:

  "backup deletedump" on page 73

  "backup dumpinfo" on page 79

  "backup status" on page 85

  "CFG_*tcid*" on page 52

- **Revised reference page for vos delentry command**

  The *IBM AFS Administration Guide* and *IBM AFS Administration Reference* incorrectly state that the **vos delentry** command accepts the name or volume ID number of any type of volume (read/write, read-only, or backup). In fact, it accepts only a read/write volume's name or ID. Because a single VLDB entry represents all versions of a volume (read/write, readonly, and backup), the command removes the entire entry even though only the read/write volume is specified. For complete documentation, see "vos delentry" on page 88.

Changes to AFS Commands, Files, and Functionality

This section briefly describes commands, command options, and functionality that are new or changed in AFS 3.6. Unless otherwise noted, the *IBM AFS Administration Guide* and *IBM AFS Administration Reference* include complete documentation of these items.

A New Command

AFS 3.6 includes the new **fs flushmount** command. The command's intended use is to discard information about mount points that has become corrupted in the cache. The next time an application accesses the mount point, the Cache Manager must fetch the most current version of it from a File Server. Data cached from files or directories in the volume is not affected. The only other way to discard the information 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 the volume root directory represented by the mount point.

New File or Command Functionality

AFS 3.6 adds the following new options and functionality to existing commands and files.

- **Changes that support XBSA servers**

  Several **backup** commands and configuration files include new features that support backup to XBSA servers such as TSM. See "New Command and File Features that Support TSM" on page 22.

- **New instructions in the CFG_*tcid* file**

  There are new instructions in the **CFG**_*tcid* file that apply to all types of backup media: **CENTRALLOG**, **GROUPID**, **LASTLOG**, **MAXPASS**, and **STATUS**. (There are also new instructions that apply only to XBSA servers, as documented in "New Command and File Features that Support TSM" on page 22.)

  The new instructions are not documented in the *IBM AFS Administration Guide* or *IBM AFS Administration Reference*. See "CFG_*tcid*" on page 52. (Note that this is a new way of referring to this file, called **CFG**_*device_name* in the *IBM AFS Administration Guide* and *IBM AFS Administration Reference*. For a Tape Coordinator that communicates with an XBSA server, the variable part of the filename is a port offset number rather than a device name, so the more generic *tcid* is a better description of possible values in this part of the filename.)

- **New -temporary flag to backup addvolset command**

  The **backup addvolset** command has a new **-temporary** flag. A temporary volume set is not recorded in the Backup Database and exists only during the lifetime of the interactive session in which it is created.

- **New options to the backup deletedump command**

  There are new options to the **backup deletedump** command: the **-groupid** argument specifies the group ID number associated with the dump records to delete, and the **-noexecute** flag displays a list of the records to be deleted rather than actually deleting them. (There are also new options that apply only to records for data dumped to an XBSA server, as documented in "New Command and File Features that Support TSM" on page 22.)

  The new options are not documented in the *IBM AFS Administration Guide* or *IBM AFS Administration Reference*. See "backup deletedump" on page 73.

- **New output from the backup dumpinfo command**

  When both the **-id** and **-verbose** options to the **backup dumpinfo** command are provided, the output is divided into several sections. In the first section, headed by the label `Dump`, the new `Group id` field replaces the `id` field that previously appeared about halfway down the list of fields (the first field in the section is still labeled `id`). The `Group id` field reports the dump's group ID number, which is recorded in the Backup Database if the **GROUPID** instruction appears in the Tape Coordinator's **/usr/afs/backup/CFG**_*tcid* file when the dump is created.

  (The command's output also includes a new message that reports whether the dump data is stored on an XBSA server, as detailed in "New Command and File Features that Support TSM" on page 22.)

  The new output is not documented in the *IBM AFS Administration Guide* or *IBM AFS Administration Reference*. See "backup dumpinfo" on page 79.

- **BOS Server sends additional field to notifier programs**

  The AFS 3.6 BOS Server sends additional information to notifier programs when an AFS server process exits. The **bnode_proc** structure now includes the **lastExit** field, which reports the exit code associated with the process's most recent exit. Previously, the only information about exit codes available to the notifier program was in the **bnode** structure's **errorCode** field, which records the exit code generated when the process last exited due to an error. The BOS Server does not clear the **errorCode** field, so the value set at the last exit due to error is reported even for exits that are not due to error.

  If your notifier program currently checks the **errorCode** field but you really want a notification only when the most recent exit is due to an error, change the program to check the **lastExit** field in the **bnode_proc** structure instead. An error code appears in the **lastExit** field only if the most recent exit really was due to an error (in which case the same code also appears in the **errorCode** field).

  The **bos create** command's reference page in the *IBM AFS Administration Reference* describes all of the fields that the BOS Server can include in the **bnode_proc** and **bnode** structures. As noted there, the BOS Server does not necessarily include every field in the structures it sends to a notifier

program, because some of them are for internal use. For best results, the notifier program must correctly handle the absence of a field that it expects to find.

- **Only administrators can use kas examine command's -showkey flag**

  As in AFS 3.5, the AFS 3.6 Authentication Server does not require that you disable authorization checking on its database server machine before it returns the octal digits that constitute the encrypted password or key stored in an Authentication Database entry, which was the requirement with earlier versions of AFS. Instead, it always returns the octal digits, as long as the connection between the **kas** command interpreter and Authentication Server is encrypted. AFS 3.5 introduced the **-showkey** flag to make the **kas examine** command display the octal digits.

  This change in requirements creates a potential security exposure, however, in that earlier versions of the **kas examine** command always display the octal digits (instead of a checksum) when directed to an AFS 3.5 or 3.6 Authentication Server. To eliminate this exposure, in AFS 3.6 the Authentication Server returns octal digits only for a principal that has the `ADMIN` flag in its Authentication Database entry.

  The main effect of the new requirement is that only administrators can include the **-showkey** flag on the AFS 3.6 **kas examine** command. It does not effectively change the privilege required to display the octal digits when using versions of the **kas examine** command before AFS 3.5 Patch 2 (build level **afs3.5 3.17**), because it was assumed with earlier versions that only administrators were able to disable authorization checking. It also does not affect the automated installation and configuration tool provided for AFS for Windows, which still can be used only by administrators.

- **The vos delentry command accepts only read/write volume names**

  The AFS 3.6 version of the **vos delentry** command accepts only read/write volume names or volume ID numbers as values for its **-id** or **-prefix** arguments. The new restriction is not documented in the *IBM AFS Administration Guide* or *IBM AFS Administration Reference*. See "vos delentry" on page 88.

Support for Backup to TSM

AFS 3.6 introduces support for backing up AFS data to media managed by the Tivoli Storage Manager (TSM), a third-party backup program which implements the Open Group's Backup Service application programming interface (API), also called *XBSA*. TSM was formerly called the ADSTAR Distributed Storage Manager, or ADSM. It is assumed that the administrator is familiar with TSM; explaining TSM or XBSA concepts or terminology is beyond the scope of this document.

See the following subsections:

New Command and File Features that Support TSM

The AFS 3.6 version of the following commands and configuration files include new options or instructions to support backup to TSM.

- **New XBSA-related instructions in the CFG_*tcid* file**

  Several new or modified instructions in the **CFG**_*tcid* file support backup of AFS data to XBSA servers such as TSM: **MGMTCLASS**, **NODE**, **PASSFILE**, **PASSWORD**, **SERVER**, and **TYPE**. (There are also new instructions that apply to automated tape devices and backup data files as well as XBSA servers, as detailed in "New File or Command Functionality" on page 19.)

  The new instructions are not documented in the *IBM AFS Administration Guide* or *IBM AFS Administration Reference*. See "CFG_*tcid*" on page 52. (Note that this is a new way of referring to this file, called **CFG**_*device_name* in the *IBM AFS Administration Guide* and *IBM AFS Administration Reference*. For a Tape Coordinator that communicates with XBSA server such as TSM, the variable part of the filename is a port offset number rather than a device name, so the more generic *tcid* is a better description of possible values in this part of the filename.)

- **New options to the backup deletedump command**

  The **backup deletedump** command has new options that enable you to delete dump records stored on an XBSA server such as TSM, as well as the corresponding Backup Database records:

  – The **-dbonly** flag deletes Backup Database records without attempting to delete the corresponding records stored on the XBSA server.

  – The **-force** flag deletes Backup Database records even if it is not possible to delete the corresponding records stored on the XBSA server.

  – The **-port** argument identifies the Tape Coordinator that communicates with the XBSA server on which to delete records.

  There are also two new options that apply to automated tape devices and backup data files as well as XBSA servers, as detailed in "New File or Command Functionality" on page 19.

  The new options are not documented in the *IBM AFS Administration Guide* or *IBM AFS Administration Reference*. See "backup deletedump" on page 73.

- **New output from the backup dumpinfo command**

  When the **-id** option is provided to the **backup dumpinfo** command, the following line appears in the output if a TSM server was the backup medium for the dump:

  ```
  Backup Service: TSM: Server: hostname
  ```

where *hostname* is the name of the TSM server machine. (There is also new output for dumps to all types of backup media, as detailed in "New File or Command Functionality" on page 19.)

The new output is not documented in the *IBM AFS Administration Guide* or *IBM AFS Administration Reference*. See "backup dumpinfo" on page 79.

- **New output from the backup status command**

  If the Tape Coordinator is communicating with a TSM server, the following message appears last in the output from the **backup status** command:

  ```
  TSM Tape coordinator
  ```

  The new output is not documented in the *IBM AFS Administration Guide* or *IBM AFS Administration Reference*. See "backup status" on page 85.

Product Notes for Use of TSM

- **Supported Tape Coordinator machine types**

  To communicate with a TSM server, the Tape Coordinator must run on a machine that uses one of the following operating systems: AIX 4.3 or higher, Solaris 2.6, Solaris 7.

- **Supported version of TSM API**

  The AFS 3.6 Tape Coordinator uses version 3.7.1 (Version 3, Release 7) of the TSM client API. Use of other versions of the API is not supported or tested. For instructions on obtaining the API, see "Configuring the Backup System and TSM" on page 24.

- **CFG_*tcid* file is required for TSM servers**

  To communicate with a TSM server, a Tape Coordinator must have a **CFG**_*tcid* file that includes the following fields: **SERVER**, **TYPE**, and **PASSWORD** or **PASSFILE**. For instructions on creating the file, see "Configuring the Backup System and TSM" on page 24.

- **No entry in tapeconfig file for TSM servers**

  Do not create an entry in the **/usr/afs/backup/tapeconfig** file for a Tape Coordinator that communicates with an XBSA server such as TSM. Creating the **CFG**_*tcid* file is sufficient.

- **Acceptable value for the TYPE instruction**

  In AFS 3.6, there is one acceptable value for the **TYPE** instruction in the **CFG**_*tcid* file: **tsm**.

- **TSM node name must be defined**

  If the **NODE** instruction is not included in the **/usr/afs/backup/CFG**_*tcid* file, the TSM **dsm.sys** file must define a value for the NODENAME variable.

- **Unsupported backup commands and options**

The following commands are not supported for XBSA servers such as TSM. In other words, the commands fail with an error message when their **-port** argument specifies a Tape Coordinator that communicates with an XBSA server:

- **backup labeltape**
- **backup readlabel**
- **backup restoredb**
- **backup savedb**
- **backup scantape**

In addition, the **-append** flag to the **backup dump** command is ignored when the **-port** argument specifies a Tape Coordinator that communicates with an XBSA server (the notion of appended dumps does not apply to XBSA servers).

Configuring the Backup System and TSM

Perform the following steps to configure TSM and the AFS Backup System for interoperation.

Note: You possibly need to perform additional TSM configuration procedures unrelated to AFS. See the TSM documentation.

1. Become the local superuser **root**, if you are not already.

   ```
   % su root
   Password: root_password
   ```

2. Install version 3.7.1 of the TSM client API on the local disk of the Tape Coordinator machine. If you do not already have the API, you can use the following instructions to download it using the UNIX File Transfer Protocol (**ftp**).

   a. Verify that there is enough free space on the local disk to accommodate the API package:

      - On AIX systems, 4 MB on the disk that houses the **/usr/tivoli** directory
      - On Solaris systems, 13 MB on the disk that houses the **/opt/tivoli** directory

   b. Connect to the **ftp** server called **ftp.software.ibm.com**, logging in as **anonymous** and providing your electronic mail address as the password.

   c. Switch to binary mode.

      ```
      ftp> bin
      ```

   d. Change directory as indicated:

      ```
      ftp> cd storage/tivoli-storage-management-maintenance/client/v3r7
      ```

   e. Change to the appropriate directory and retrieve the API file.

      - On an AIX 4.3 system:

```
ftp> cd AIX/v371
ftp> get tivoli.tsm.client.api.aix43.32bit
```

- On a Solaris 2.6 or 7 system:

```
ftp> cd Solaris/v371
ftp> get IP21804.tar.Z
```

f. Use the appropriate tool to install the TSM API package locally:

- On AIX machines, use **smit**, which installs the files in the **/usr/tivoli/tsm/client/api/bin** directory
- On Solaris machines, use the following command, which installs the files in the **/opt/tivoli/tsm/client/api/bin** directory:

```
# uncompress IP21804.tar.Z | tar xvf -
```

3. Set the following TSM environment variables as indicated. If you do not set them, you must use the default values specified in the TSM documentation.

DSMI_DIR
: Specifies the pathname of the directory that contains the TSM client system options file, **dsm.sys**. The directory must have a subdirectory (which can be a symbolic link) called **en_US** that contains the **dsmclientV3.cat** catalog file.

  Do not put a final slash (**/**) on the directory name. Examples of appropriate values are **/opt/tivoli/tsm/client/api/bin** on Solaris machines and **/usr/tivoli/tsm/client/api/bin** on AIX machines.

DSMI_CONFIG
: Specifies the pathname of the directory that contains the TSM client user options file, **dsm.opt**. The value can be the same as for the **DSMI_DIR** variable. Do not put a final slash (**/**) on the directory name.

DSMI_LOG
: Specifies the full pathname (including the filename) of the log file for error messages from the API. An appropriate value is **/usr/afs/backup/butc.TSMAPI.log**.

4. Verify that the **dsm.sys** file includes the following instructions. For a description of the fields, see the TSM documentation.

```
ServerName           machine_name
   CommMethod        tcpip
   TCPPort           TSM_port
   TCPServerAddress  full_machine_name
   PasswordAccess    prompt
   Compression       yes
```

The following is an example of appropriate values:

```
ServerName tsm3
   CommMethod tcpip
   TCPPort 1500
   TCPServerAddress  tsm3.abc.com
   PasswordAccess  prompt
   Compression  yes
```

5. Verify that the **dsm.opt** file includes the following instructions. For a description of the fields, see the TSM documentation.

```
ServerName        machine_name
   tapeprompt     no
   compressalways yes
```

6. Create a Backup Database entry for each Tape Coordinator that is to communicate with the TSM server. Multiple Tape Coordinators can interact with the same TSM server if the server has sufficient capacity.

```
# backup addhost <tape machine name> <TC port offset>
```

   where

   tape machine name
   : Specifies the fully qualified hostname of the Tape Coordinator machine.

   TC port offset
   : Specifies the Tape Coordinator's port offset number. Acceptable values are integers in the range from **0** (zero) through **58510**.

7. Create a device configuration file for the Tape Coordinator called **/usr/afs/backup/CFG**_*tcid*, where *tcid* is the Tape Coordinator's port offset number as defined in Step 6. The file must include the following instructions:

   - **SERVER**, which takes as its argument the fully qualified hostname of the TSM server machine. It matches the value in the *full_machine_name* field of the **dsm.sys** file, as defined in Step 4 on page 25.
   - **TYPE**, which takes as its argument the string **tsm** (the only acceptable value in AFS 3.6).
   - One of **PASSWORD** or **PASSFILE**, to define the password which the Tape Coordinator uses when communicating with the TSM server. **PASSWORD** takes as its argument the actual password character string. **PASSFILE** takes as its argument the complete pathname of the file that contains the string on its first line.

   For more detailed descriptions of the instructions, and of other instructions you can include in the configuration file, see "CFG_*tcid*" on page 52.

Upgrading Server and Client Machines to AFS 3.6

This section explains how to upgrade server and client machines from AFS 3.5 or AFS 3.6 Beta to AFS 3.6. Before performing an upgrade, please read all of the introductory material in this section.

If you are installing AFS for the first time, skip this chapter and refer to the *IBM AFS Quick Beginnings* document for AFS 3.6.

AFS provides backward compatibility to the previous release only: AFS 3.6 is certified to be compatible with AFS 3.5 but not necessarily with earlier versions.

Note: This document does not provide instructions for upgrading from AFS 3.4a or earlier directly to AFS 3.6. A file system conversion is required on some system types. See the *AFS Release Notes* for AFS 3.5 and contact your AFS product support representative for assistance.

Prerequisites for Upgrading

You must meet the following requirements to upgrade successfully to AFS 3.6:

- You can access the AFS 3.6 binaries by network, or have the CD-ROM labeled **AFS Version 3.6** for each system type you need to upgrade. See "Obtaining the Binary Distribution".
- You have access to the *IBM AFS Quick Beginnings* document for AFS 3.6, either in hardcopy (for English, IBM document number SC09-4560-00, part number CT6Q7NA) or at **http://www.transarc.com/Library/documentation/afs_doc.html**. See also "Accessing the AFS Binary Distribution and Documentation" on page 2.
- The partition that houses the **/usr/afs/bin** directory on each server machine has at least 18 MB of disk space for storing the AFS server binaries.
- The partition that houses the **/usr** directory on each client machine has at least 4 MB of disk space for storing the AFS client binaries and kernel library files (stored by convention in the **/usr/vice/etc** directory).
- You can log into all server and client machines as the local superuser **root**.
- You are listed in the cell's **/usr/afs/etc/UserList** file and can authenticate as a member of the **system:administrators** group.

Obtaining the Binary Distribution

Use one of the following methods to obtain the AFS distribution of each system type for which you are licensed.

- Working with your AFS Sales Representative, obtain the AFS 3.6 CD-ROM for each system type.
- Access the distribution by network in IBM's Electronic Software Distribution system.

Storing Binaries in AFS

It is conventional to store many of the programs and files from the AFS binary distribution in a separate volume for each system type, mounted in your AFS filespace at **/afs/***cellname*/*sysname*/**usr/afsws**. These instructions rename the volume currently mounted at this location and create a new volume for AFS 3.6 binaries.

Repeat the instructions for each system type.

1. Authenticate as an administrator listed in the **/usr/afs/etc/UserList** file.
2. Issue the **vos create** command to create a new volume for AFS 3.6 binaries called *sysname***.3.6**. Set an unlimited quota on the volume to avoid running out of space as you copy files from the distribution.

   ```
   % vos create <machine name> <partition name> sysname.3.6  -maxquota  0
   ```

3. Issue the **fs mkmount** command to mount the volume at a temporary location.

   ```
   % fs mkmount  /afs/.cellname/temp  sysname.3.6
   ```

4. Prepare to access the files using the method you have selected:
   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

     ```
     % cd /cdrom/sysname
     ```

   - If accessing the distribution electronically, download the necessary file or files. If necessary, use commands such as **gunzip** and **tar xvf** to uncompress and unpack the distribution. Place the contents in a temporary location (*temp_afs36_dir*) and change directory to that location.

     ```
     %  cd  temp_afs36_dir
     ```

5. Copy files from the distribution into the *sysname***.3.6** volume.

   ```
   % cp -rp  bin  /afs/.cellname/temp

   % cp -rp  etc  /afs/.cellname/temp

   % cp -rp  include  /afs/.cellname/temp

   % cp -rp  lib  /afs/.cellname/temp
   ```

6. **(Optional)** By convention, the contents of the distribution's **root.client** directory are not stored in AFS. However, if you are upgrading client functionality on many machines, it can be simpler to copy the client files from your local AFS space than from the CD-ROM or from IBM's Electronic Software Distribution system. If you wish to store the contents of the **root.client** directory in AFS temporarily, copy them now.

   ```
   % cp -rp  root.client  /afs/.cellname/temp
   ```

7. Issue the **vos rename** command to change the name of the volume currently mounted at the **/afs/***cellname***/***sysname***/usr/afsws** directory. A possible value for the *extension* reflects the AFS version and build level (for example: **3.5-bld3.32**).

   If you do not plan to retain the old volume, you can substitute the **vos remove** command in this step.

   ```
   % vos rename sysname.usr.afsws  sysname.usr.afsws.extension
   ```

8. Issue the **vos rename** command to change the name of the *sysname***.3.6** volume to *sysname***.usr.afsws**.

   ```
   % vos rename sysname.3.6  sysname.usr.afsws
   ```

9. Issue the **fs rmmount** command to remove the temporary mount point for the *sysname***.3.6** volume.

   ```
   % fs rmmount  /afs/.cellname/temp
   ```

Upgrading the Operating System

AFS 3.6 supports the 64–bit version of HP-UX 11.0 and Solaris 7. To upgrade from the 32–bit version, you possibly need to reinstall the operating system completely before installing AFS 3.6. When performing any operating system upgrade, you must take several actions to preserve AFS functionality, including the following:

- Unmount the AFS server partitions (those mounted on **/vicep** directories) on all file server machines, to prevent the standard vendor version of the **fsck** program from running on them when you reboot the machine during installation of the new operating system. On several operating systems, the standard **fsck** program does not recognize AFS volume data and discards it. Also, disable automatic mounting of the partitions during reboot until you have substituted the AFS **vfsck** program for the vendor **fsck** program.
- Create copies of the AFS-modified versions of binaries or files so that they are not overwritten by the standard versions during the operating system upgrade, particularly if you are not performing an immediate AFS upgrade. Examples include the remote commands (**ftpd**, **inetd**, **rcp**, **rsh**, and so on) and the **vfsck** binary. After you have successfully installed the new version of the operating system, move the AFS-modified files and commands back to the directories from which they are accessed during normal use.

Distributing Binaries to Server Machines

The instructions in this section explain how to use the Update Server to distribute server binaries from a binary distribution machine of each system type.

Repeat the steps on each binary distribution machine in your cell. If you do not use the Update Server, repeat the steps on every server machine in your cell. If you are copying files from the AFS product tree, the server machine must also be configured as an AFS client machine.

1. Become the local superuser **root**, if you are not already.

   ```
   % su root
   Password: root_password
   ```

2. Create a temporary subdirectory of the **/usr/afs/bin** directory to store the AFS 3.6 server binaries.

   ```
   # mkdir /usr/afs/bin.36
   ```

3. Prepare to access server files using the method you have selected from those listed in "Obtaining the Binary Distribution" on page 27:

   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

     ```
     # cd /cdrom/sysname/root.server/usr/afs/bin
     ```

   - If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

     ```
     # cd  temp_afs36_dir/root.server/usr/afs/bin
     ```

4. Copy the server binaries from the distribution into the **/usr/afs/bin.36** directory.

   ```
   # cp -p  *  /usr/afs/bin.36
   ```

5. Rename the current **/usr/afs/bin** directory to **/usr/afs/bin.old** and the **/usr/afs/bin.36** directory to the standard location.

   ```
   # cd /usr/afs

   # mv  bin  bin.old

   # mv  bin.36  bin
   ```

Upgrading Server Machines

Repeat the following instructions on each server machine. Perform them first on the database server machine with the lowest IP address, next on the other database server machines, and finally on other server machines.

The AFS data stored on a server machine is inaccessible to client machines during the upgrade process, so it is best to perform it at the time and in the manner that disturbs your users least.

1. If you have just followed the steps in "Distributing Binaries to Server Machines" on page 29 to install the server binaries on binary distribution machines, wait the required interval (by default, five minutes) for the local **upclientbin** process to retrieve the binaries.

If you do not use binary distribution machines, perform the instructions in "Distributing Binaries to Server Machines" on page 29 on this machine.

2. Become the local superuser **root**, if you are not already, by issuing the **su** command.

```
% su root
Password: root_password
```

3. If the machine also functions as a client machine, prepare to access client files using the method you have selected from those listed in "Obtaining the Binary Distribution" on page 27:

   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

```
# cd /afs/cellname/sysname/usr/afsws/root.client
```

   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

```
# cd /cdrom/sysname/root.client
```

   - If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

```
# cd  temp_afs36_dir/root.client
```

4. If the machine also functions as a client machine, copy the AFS 3.6 version of the **afsd** binary and other files to the **/usr/vice/etc** directory.

   Note: Some files in the **/usr/vice/etc** directory, such as the AFS initialization file (called **afs.rc** on many system types), do not necessarily need to change for a new release. It is a good policy to compare the contents of the distribution directory and the **/usr/vice/etc** directory before performing the copying operation. If there are files in the **/usr/vice/etc** directory that you created for AFS 3.5 or 3.6 Beta and that you want to retain, either move them to a safe location before performing the following instructions, or alter the following instructions to copy over only the appropriate files.

```
# cp  -p  usr/vice/etc/*   /usr/vice/etc

# cp  -rp  usr/vice/etc/C  /usr/vice/etc
```

   If you have not yet incorporated AFS into the machine's authentication system, perform the instructions in the section titled *Enabling AFS Login* for this system type in the *IBM AFS Quick Beginnings* chapter about

configuring client machines. If this machine was running the same operating system revision with AFS 3.5 or AFS 3.6 Beta, you presumably already incorporated AFS into its authentication system.

5. AFS performance is most dependable if the AFS release version of the kernel extensions and server processes is the same. Therefore, it is best to incorporate the AFS 3.6 kernel extensions into the kernel at this point.

   First issue the following command to shut down the server processes, preventing them from restarting accidently before you incorporate the AFS 3.6 extensions into the kernel.

   ```
   # bos shutdown <machine name> -localauth -wait
   ```

   Then perform the instructions in "Incorporating AFS into the Kernel and Enabling the AFS Initialization Script" on page 34, which have you reboot the machine. Assuming that the machine's AFS initialization script is configured to invoke the **bosserver** command as specified in *IBM AFS Quick Beginnings*, the BOS Server starts itself and then the other AFS server processes listed in its local **/usr/afs/local/BosConfig** file.

   There are two circumstances in which you must incorporate the kernel extensions and reboot now rather than later:

   - You are upgrading the File Server on an HP-UX machine
   - The machine also serves as a client, you upgraded the client files in the previous step, and you want the new Cache Manager to become operative right away

   In any other circumstances, you can choose to upgrade the kernel extensions later. Choose one of the following options:

   - Restart all server processes by issuing the **bos restart** command with the **-bosserver** flag.

     ```
     # bos restart <machine name> -localauth -bosserver
     ```

   - Wait to start using the new binaries until the processes restart automatically at the binary restart time specified in the **/usr/afs/local/BosConfig** file.

6. Once you are satisfied that the machine is functioning correctly at AFS 3.6, there is no need to retain previous versions of the server binaries in the **/usr/afs/bin** directory. (You can always use the **bos install** command to reinstall them if it becomes necessary to downgrade). If you use the Update Server, the **upclientbin** process renamed them with a **.old** extension in Step 1 on page 30. To reclaim the disk space occupied in the **/usr/afs/bin** directory by **.bak** and **.old** files, you can use the following command:

   ```
   # bos prune <machine name> -bak -old -localauth
   ```

Step 5 on page 30 of "Distributing Binaries to Server Machines" on page 29 had you move the previous version of the binaries to the **/usr/afs/bin.old** directory. You can also remove that directory on any machine where you created it.

```
# rm -rf  /usr/afs/bin.old
```

Upgrading Client Machines

1. Become the local superuser **root**, if you are not already, by issuing the **su** command.

   ```
   % su root
   Password: root_password
   ```

2. Prepare to access client files using the method you have selected from those listed in "Obtaining the Binary Distribution" on page 27:

   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

     ```
     # cd /afs/cellname/sysname/usr/afsws/root.client
     ```

   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

     ```
     # cd /cdrom/sysname/root.client
     ```

   - If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

     ```
     # cd  temp_afs36_dir/root.client
     ```

3. Copy the AFS 3.6 version of the **afsd** binary and other files to the **/usr/vice/etc** directory.

   Note: Some files in the **/usr/vice/etc** directory, such as the AFS initialization file (called **afs.rc** on many system types), do not necessarily need to change for a new release. It is a good policy to compare the contents of the distribution directory and the **/usr/vice/etc** directory before performing the copying operation. If there are files in the **/usr/vice/etc** directory that you created for AFS 3.5 or 3.6 Beta and that you want to retain, either move them to a safe location before performing the following instructions, or alter the following instructions to copy over only the appropriate files.

   ```
   # cp  -p  usr/vice/etc/*   /usr/vice/etc

   # cp  -rp  usr/vice/etc/C  /usr/vice/etc
   ```

If you have not yet incorporated AFS into the machine's authentication system, perform the instructions in the section titled *Enabling AFS Login* for this system type in the *IBM AFS Quick Beginnings* chapter about configuring client machines. If this machine was running the same operating system revision with AFS 3.5 or AFS 3.6 Beta, you presumably already incorporated AFS into its authentication system.

4. Perform the instructions in "Incorporating AFS into the Kernel and Enabling the AFS Initialization Script" to incorporate AFS extensions into the kernel. The instructions conclude with a reboot of the machine, which starts the new Cache Manager.

Incorporating AFS into the Kernel and Enabling the AFS Initialization Script

As part of upgrading a machine to AFS 3.6, you must incorporate AFS 3.6 extensions into its kernel and verify that the AFS initialization script is included in the machine's startup sequence. Proceed to the instructions for your system type:

- "Loading AFS into the AIX Kernel"
- "Building AFS into the Digital UNIX Kernel" on page 36
- "Building AFS into the HP-UX Kernel" on page 39
- "Incorporating AFS into the IRIX Kernel" on page 41
- "Loading AFS into the Linux Kernel" on page 46
- "Loading AFS into the Solaris Kernel" on page 48

Loading AFS into the AIX Kernel

The AIX kernel extension facility is the dynamic kernel loader provided by IBM Corporation. AIX does not support incorporation of AFS modifications during a kernel build.

For AFS to function correctly, the kernel extension facility must run each time the machine reboots, so the AFS initialization script (included in the AFS distribution) invokes it automatically. In this section you copy the script to the conventional location and edit it to select the appropriate options depending on whether NFS is also to run.

After editing the script, you verify that there is an entry in the AIX **inittab** file that invokes it, then reboot the machine to incorporate the new AFS extensions into the kernel and restart the Cache Manager.

1. Access the AFS distribution by changing directory as indicated. Substitute **rs_aix42** for the *sysname* variable.
   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

     ```
     # cd /afs/cellname/sysname/usr/afsws/root.client
     ```

- If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

  ```
  # cd /cdrom/sysname/root.client
  ```

- If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

  ```
  # cd  temp_afs36_dir/root.client
  ```

2. Copy the AFS kernel library files to the local **/usr/vice/etc/dkload** directory.

   ```
   # cd  usr/vice/etc

   # cp -rp  dkload  /usr/vice/etc
   ```

3. Because you ran AFS 3.5 on this machine, the appropriate AFS initialization file possibly already exists as **/etc/rc.afs**. Compare it to the version in the **root.client/usr/vice/etc** directory of the AFS 3.6 distribution to see if any changes are needed.

   If the initialization file is not already in place, copy it now.

   ```
   # cp -p  rc.afs  /etc/rc.afs
   ```

4. Edit the **/etc/rc.afs** script, setting the `NFS` variable if it is not already.

   - If the machine is not to function as an NFS/AFS Translator, set the NFS variable as follows:

     ```
     NFS=$NFS_NONE
     ```

   - If the machine is to function as an NFS/AFS Translator and is running AIX 4.2.1 or higher, set the NFS variable as follows. Only sites that have a license for the NFS/AFS Translator are allowed to run translator machines. Machines running the base level of AIX 4.2 cannot be translator machines.

     NFS must already be loaded into the kernel. It is loaded automatically on machines running AIX 4.1.1 and later, as long as the file **/etc/exports** exists.

     ```
     NFS=$NFS_IAUTH
     ```

5. Place the following line in the AIX initialization file, **/etc/inittab**, if it is not already. It invokes the AFS initialization script and needs to appear just after the line that starts NFS daemons.

   ```
   rcafs:2:wait:/etc/rc.afs > /dev/console 2>&1 # Start AFS services
   ```

6. **(Optional)** There are now copies of the AFS initialization file in both the **/usr/vice/etc** and **/etc** directories. If you want to avoid potential confusion

by guaranteeing that they are always the same, create a link between them. You can always retrieve the original script from the AFS distribution if necessary.

```
# cd  /usr/vice/etc

# rm  rc.afs

# ln -s  /etc/rc.afs
```

7. Reboot the machine.

```
# shutdown -r now
```

8. If you are upgrading a server machine, login again as the local superuser **root**, then return to Step 6 on page 32 in "Upgrading Server Machines" on page 30.

```
login: root
Password: root_password
```

Building AFS into the Digital UNIX Kernel

On Digital UNIX machines, you must build AFS modifications into a new static kernel; Digital UNIX does not support dynamic loading. If the machine's hardware and software configuration exactly matches another Digital UNIX machine on which AFS 3.6 is already built into the kernel, you can choose to copy the kernel from that machine to this one. In general, however, it is better to build AFS modifications into the kernel on each machine according to the following instructions.

If the machine was running a version of Digital UNIX 4.0 with a previous version of AFS, the configuration changes specified in Step 1 through Step 4 on page 37 are presumably already in place.

1. Create a copy called **AFS** of the basic kernel configuration file included in the Digital UNIX distribution as **/usr/sys/conf/***machine_name*, where *machine_name* is the machine's hostname in all uppercase letters.

```
# cd /usr/sys/conf

# cp machine_name AFS
```

2. Add AFS to the list of options in the configuration file you created in the previous step, so that the result looks like the following:

```
        .                   .
        .                   .
options                 UFS
options                 NFS
options                 AFS
        .                   .
        .                   .
```

3. Add an entry for AFS to two places in the **/usr/sys/conf/files** file.
   - Add a line for AFS to the list of `OPTIONS`, so that the result looks like the following:

```
        .               .       .
        .               .       .
OPTIONS/nfs         optional nfs define_dynamic
OPTIONS/afs         optional afs define_dynamic
OPTIONS/cdfs        optional cdfs define_dynamic
        .               .       .
        .               .       .
```

- Add an entry for AFS to the list of MODULES, so that the result looks like the following:

```
        .           .       .          .
        .           .       .          .
#
MODULE/nfs_server optional nfs_server Binary
nfs/nfs_server.c module nfs_server optimize -g3
nfs/nfs3_server.c module nfs_server optimize -g3
#
MODULE/afs optional afs Binary
afs/libafs.c module afs
#
```

4. Add an entry for AFS to two places in the **/usr/sys/vfs/vfs_conf.c** file.
   - Add AFS to the list of defined file systems, so that the result looks like the following:

```
   .       .
   .       .
#include <afs.h>
#if defined(AFS) && AFS
       extern struct vfsops afs_vfsops;
#endif
   .       .
   .       .
```

   - Put a declaration for AFS in the **vfssw[]** table's MOUNT_ADDON slot, so that the result looks like the following:

```
      .                          .             .
      .                          .             .
   &fdfs_vfsops,         "fdfs",   /* 12 = MOUNT_FDFS */
#if defined(AFS)
   &afs_vfsops,          "afs",
#else
   (struct vfsops *)0,   "",       /* 13 = MOUNT_ADDON */
#endif
#if NFS && INFS_DYNAMIC
    &nfs3_vfsops,        "nfsv3",  /* 14 = MOUNT_NFS3 */
```

5. Access the AFS distribution by changing directory as indicated. Substitute **alpha_dux40** for the *sysname* variable.
   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

```
# cd /afs/cellname/sysname/usr/afsws/root.client
```

- If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

  ```
  # cd /cdrom/sysname/root.client
  ```

- If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

  ```
  # cd  temp_afs36_dir/root.client
  ```

6. Because you ran AFS 3.5 on this machine, the appropriate AFS initialization file possibly already exists as **/sbin/init.d/afs**. Compare it to the version in the **root.client/usr/vice/etc** directory of the AFS 3.6 distribution to see if any changes are needed.

   If the initialization file is not already in place, copy it now. Note the removal of the **.rc** extension as you copy.

   ```
   # cp  -p  usr/vice/etc/afs.rc  /sbin/init.d/afs
   ```

7. Copy the AFS kernel module to the local **/usr/sys/BINARY** directory.

   The AFS 3.6 distribution includes only the **libafs.nonfs.o** version of the library, because Digital UNIX machines are not supported as NFS/AFS Translator machines.

   ```
   # cp  -p  bin/libafs.nonfs.o  /usr/sys/BINARY/afs.mod
   ```

8. Configure and build the kernel. Respond to any prompts by pressing <**Return**>. The resulting kernel is in the file **/sys/AFS/vmunix**.

   ```
   # doconfig -c AFS
   ```

9. Rename the existing kernel file and copy the new, AFS-modified file to the standard location.

   ```
   # mv  /vmunix  /vmunix_orig

   # cp  -p  /sys/AFS/vmunix  /vmunix
   ```

10. Verify the existence of the symbolic links specified in the following commands, which incorporate the AFS initialization script into the Digital UNIX startup and shutdown sequence. If necessary, issue the commands to create the links.

    ```
    # ln -s  ../init.d/afs  /sbin/rc3.d/S67afs

    # ln -s  ../init.d/afs  /sbin/rc0.d/K66afs
    ```

11. **(Optional)** If the machine is configured as a client, there are now copies of the AFS initialization file in both the **/usr/vice/etc** and **/sbin/init.d** directories. If you want to avoid potential confusion by guaranteeing that

they are always the same, create a link between them. You can always retrieve the original script from the AFS distribution if necessary.

```
# cd  /usr/vice/etc

# rm  afs.rc

# ln -s  /sbin/init.d/afs  afs.rc
```

12. Reboot the machine.

```
# shutdown -r now
```

13. If you are upgrading a server machine, login again as the local superuser **root**, then return to Step 6 on page 32 in "Upgrading Server Machines" on page 30.

```
login: root
Password: root_password
```

Building AFS into the HP-UX Kernel

On HP-UX machines, you must build AFS modifications into a new kernel; HP-UX does not support dynamic loading. If the machine's hardware and software configuration exactly matches another HP-UX machine on which AFS 3.6 is already built into the kernel, you can choose to copy the kernel from that machine to this one. In general, however, it is better to build AFS modifications into the kernel on each machine according to the following instructions.

1. Move the existing kernel-related files to a safe location.

```
# cp -p  /stand/vmunix  /stand/vmunix.noafs

# cp -p  /stand/system  /stand/system.noafs
```

2. Access the AFS distribution by changing directory as indicated. Substitute **hp_ux110** for the *sysname* variable.
   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

     ```
     # cd /afs/cellname/sysname/usr/afsws/root.client
     ```

   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

     ```
     # cd /cdrom/sysname/root.client
     ```

   - If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

```
# cd temp_afs36_dir/root.client
```

3. Because you ran AFS 3.5 on this machine, the appropriate AFS initialization file possibly already exists as **/sbin/init.d/afs**. Compare it to the version in the **root.client/usr/vice/etc** directory of the AFS 3.6 distribution to see if any changes are needed.

   If the initialization file is not already in place, copy it now. Note the removal of the **.rc** extension as you copy.

   ```
   # cp -p usr/vice/etc/afs.rc /sbin/init.d/afs
   ```

4. Copy the file **afs.driver** to the local **/usr/conf/master.d** directory, changing its name to **afs** as you do so.

   ```
   # cp -p usr/vice/etc/afs.driver /usr/conf/master.d/afs
   ```

5. Copy the AFS kernel module to the local **/usr/conf/lib** directory.

   HP-UX machines are not supported as NFS/AFS Translator machines, so AFS 3.6 includes only libraries called **libafs.nonfs.a** (for the 32–bit version of HP-UX) and **libafs64.nonfs.a** (for the 64–bit version of HP-UX). Change the library's name to **libafs.a** as you copy it.

   For the 32–bit version of HP-UX:

   ```
   # cp -p bin/libafs.nonfs.a /usr/conf/lib/libafs.a
   ```

   For the 64–bit version of HP-UX:

   ```
   # cp -p bin/libafs64.nonfs.a /usr/conf/lib/libafs.a
   ```

6. Verify the existence of the symbolic links specified in the following commands, which incorporate the AFS initialization script into the HP-UX startup and shutdown sequence. If necessary, issue the commands to create the links.

   ```
   # ln -s ../init.d/afs /sbin/rc2.d/S460afs

   # ln -s ../init.d/afs /sbin/rc2.d/K800afs
   ```

7. **(Optional)** If the machine is configured as a client, there are now copies of the AFS initialization file in both the **/usr/vice/etc** and **/sbin/init.d** directories. If you want to avoid potential confusion by guaranteeing that they are always the same, create a link between them. You can always retrieve the original script from the AFS distribution if necessary.

   ```
   # cd /usr/vice/etc

   # rm afs.rc

   # ln -s /sbin/init.d/afs afs.rc
   ```

8. Incorporate the AFS driver into the kernel, either using the **SAM** program or a series of individual commands. Both methods reboot the machine, which loads the new kernel and starts the Cache Manager.

   - To use the **SAM** program:

a. Invoke the **SAM** program, specifying the hostname of the local machine as *local_hostname*. The **SAM** graphical user interface pops up.

```
# sam -display local_hostname:0
```

b. Choose the **Kernel Configuration** icon, then the **Drivers** icon. From the list of drivers, select **afs**.

c. Open the pull-down **Actions** menu and choose the **Add Driver to Kernel** option.

d. Open the **Actions** menu again and choose the **Create a New Kernel** option.

e. Confirm your choices by choosing **Yes** and **OK** when prompted by subsequent pop-up windows. The **SAM** program builds the kernel and reboots the system.

f. Login again as the superuser **root**.

```
login: root
Password: root_password
```

- To use individual commands:

a. Edit the file **/stand/system**, adding an entry for **afs** to the `Subsystems` section.

b. Change to the **/stand/build** directory and issue the **mk_kernel** command to build the kernel.

```
# cd /stand/build

# mk_kernel
```

c. Move the new kernel to the standard location (**/stand/vmunix**), reboot the machine to start using it, and login again as the superuser **root**.

```
# mv /stand/build/vmunix_test /stand/vmunix

# cd /

# shutdown -r now

login: root
Password: root_password
```

9. If you are upgrading a server machine, login again as the local superuser **root**, then return to Step 6 on page 32 in "Upgrading Server Machines" on page 30.

```
login: root
Password: root_password
```

Incorporating AFS into the IRIX Kernel

To incorporate AFS into the kernel on IRIX machines, choose one of two methods:

- Dynamic loading using the **ml** program distributed by Silicon Graphics, Incorporated (SGI).
- Building a new static kernel. Proceed to "Building AFS into the IRIX Kernel" on page 43.

Loading AFS into the IRIX Kernel

The **ml** program is the dynamic kernel loader provided by SGI for IRIX systems. If you use it rather than building AFS modifications into a static kernel, then for AFS to function correctly the **ml** program must run each time the machine reboots. Therefore, the AFS initialization script (included on the AFS CD-ROM) invokes it automatically when the **afsml** configuration variable is activated. In this section you activate the variable and run the script.

1. Issue the **uname -m** command to determine the machine's CPU type. The **IP***xx* value in the output must match one of the supported CPU types listed in "Supported System Types" on page 2.

   ```
   # uname -m
   ```

2. Access the AFS distribution by changing directory as indicated. Substitute **sgi_65** for the *sysname* variable.
   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

     ```
     # cd /afs/cellname/sysname/usr/afsws/root.client
     ```

   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

     ```
     # cd /cdrom/sysname/root.client
     ```

   - If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

     ```
     # cd  temp_afs36_dir/root.client
     ```

3. Copy the appropriate AFS kernel library file to the local **/usr/vice/etc/sgiload** directory; the **IP***xx* portion of the library file name must match the value returned by the **uname -m** command. Also choose the file appropriate to whether the machine's kernel supports NFS server functionality (NFS must be supported for the machine to act as an NFS/AFS Translator). Single- and multiprocessor machines use the same library file.

You can choose to copy all of the kernel library files into the **/usr/vice/etc/sgiload** directory, but they require a significant amount of space.

```
# cd  usr/vice/etc/sgiload
```

If the machine is not to act as an NFS/AFS translator:

```
# cp -p  libafs.IPxx.nonfs.o  /usr/vice/etc/sgiload
```

If the machine is to act as an NFS/AFS translator, in which case its kernel must support NFS server functionality:

```
# cp -p   libafs.IPxx.o   /usr/vice/etc/sgiload
```

4. Proceed to "Enabling the AFS Initialization Script on IRIX Systems" on page 44.

Building AFS into the IRIX Kernel

If you prefer to build a kernel, and the machine's hardware and software configuration exactly matches another IRIX machine on which AFS 3.6 is already built into the kernel, you can choose to copy the kernel from that machine to this one. In general, however, it is better to build AFS modifications into the kernel on each machine according to the following instructions.

1. Access the AFS distribution by changing directory as indicated. Substitute **sgi_65** for the *sysname* variable.
   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

     ```
     # cd /afs/cellname/sysname/usr/afsws/root.client
     ```

   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

     ```
     # cd /cdrom/sysname/root.client
     ```

   - If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

     ```
     # cd  temp_afs36_dir/root.client
     ```

2. Issue the **uname -m** command to determine the machine's CPU type. The **IP***xx* value in the output must match one of the supported CPU types listed in the *IBM AFS Release Notes* for the current version of AFS.

   ```
   # uname -m
   ```

3. Copy the appropriate AFS kernel library file to the local file **/var/sysgen/boot/afs.a**; the **IP***xx* portion of the library file name must match the value returned by the **uname -m** command. Also choose the file appropriate to whether the machine's kernel supports NFS server functionality (NFS must be supported for the machine to act as an NFS/AFS Translator). Single- and multiprocessor machines use the same library file.

   ```
   # cd  bin
   ```

   If the machine is not to act as an NFS/AFS translator:

   ```
   # cp -p  libafs.IPxx.nonfs.a   /var/sysgen/boot/afs.a
   ```

   If the machine is to act as an NFS/AFS translator, in which case its kernel must support NFS server functionality:

   ```
   # cp -p   libafs.IPxx.a   /var/sysgen/boot/afs.a
   ```

4. Copy the kernel initialization file **afs.sm** to the local **/var/sysgen/system** directory, and the kernel master file **afs** to the local **/var/sysgen/master.d** directory.

   ```
   # cp -p  afs.sm  /var/sysgen/system

   # cp -p  afs  /var/sysgen/master.d
   ```

5. Copy the existing kernel file, **/unix**, to a safe location and compile the new kernel. It is created as **/unix.install**, and overwrites the existing **/unix** file when the machine reboots.

   ```
   # cp -p  /unix  /unix_orig

   # autoconfig
   ```

6. Proceed to "Enabling the AFS Initialization Script on IRIX Systems".

Enabling the AFS Initialization Script on IRIX Systems

1. Because you ran AFS 3.5 on this machine, the appropriate AFS initialization file possibly already exists as **/etc/init.d/afs**. Compare it to the version in the **root.client/usr/vice/etc** directory of the AFS 3.6 distribution to see if any changes are needed.

   If the initialization file is not already in place, copy it now. If the machine is configured as a client machine, you already copied the script to the local **/usr/vice/etc** directory. Otherwise, change directory as indicated, substituting **sgi_65** for the *sysname* variable.

   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

     ```
     # cd /afs/cellname/sysname/usr/afsws/root.client
     ```

   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on

mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

```
# cd /cdrom/sysname/root.client
```

- If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

  ```
  # cd  temp_afs36_dir/root.client
  ```

Now copy the script. Note the removal of the **.rc** extension as you copy.

```
# cp -p  script_location/afs.rc  /etc/init.d/afs
```

2. If the **afsml** configuration variable is not already set appropriately, issue the **chkconfig** command.

   If you are using the **ml** program:

   ```
   # /etc/chkconfig -f afsml on
   ```

   If you built AFS into a static kernel:

   ```
   # /etc/chkconfig -f afsml off
   ```

   If the machine is to function as an NFS/AFS Translator, the kernel supports NFS server functionality, and the **afsxnfs** variable is not already set appropriately, set it now.

   ```
   # /etc/chkconfig -f afsxnfs on
   ```

3. Verify the existence of the symbolic links specified in the following commands, which incorporate the AFS initialization script into the IRIX startup and shutdown sequence. If necessary, issue the commands to create the links.

   ```
   # ln -s ../init.d/afs /etc/rc2.d/S35afs

   # ln -s ../init.d/afs /etc/rc0.d/K35afs
   ```

4. **(Optional)** If the machine is configured as a client, there are now copies of the AFS initialization file in both the **/usr/vice/etc** and **/etc/init.d** directories. If you want to avoid potential confusion by guaranteeing that they are always the same, create a link between them. You can always retrieve the original script from the AFS distribution if necessary.

   ```
   # cd /usr/vice/etc

   # rm afs.rc

   # ln -s  /etc/init.d/afs  afs.rc
   ```

5. Reboot the machine.

   ```
   # shutdown -i6 -g0 -y
   ```

6. If you are upgrading a server machine, login again as the local superuser **root**, then return to Step 6 on page 32 in "Upgrading Server Machines" on page 30.

```
login: root
Password: root_password
```

Loading AFS into the Linux Kernel

The **insmod** program is the dynamic kernel loader for Linux. Linux does not support incorporation of AFS modifications during a kernel build.

For AFS to function correctly, the **insmod** program must run each time the machine reboots, so the AFS initialization script (included on the AFS CD-ROM) invokes it automatically. The script also includes commands that select the appropriate AFS library file automatically. In this section you run the script.

1. Access the AFS distribution by changing directory as indicated. Substitute **i386_linux22** for the *sysname* variable.
   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

     ```
     # cd /afs/cellname/sysname/usr/afsws/root.client
     ```
   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

     ```
     # cd /cdrom/sysname/root.client
     ```
   - If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

     ```
     # cd  temp_afs36_dir/root.client
     ```
2. Copy the AFS kernel library files to the local **/usr/vice/etc/modload** directory. The filenames for the libraries have the format **libafs-***version***.o**, where *version* indicates the kernel build level. The string **.mp** in the *version* indicates that the file is appropriate for use with symmetric multiprocessor (SMP) kernels.

   ```
   # cd  usr/vice/etc

   # cp -rp  modload  /usr/vice/etc
   ```
3. The AFS 3.6 distribution includes a new AFS initialization file that can select automatically from the kernel extensions included in AFS 3.6. Copy it to the **/etc/rc.d/init.d** directory, removing the **.rc** extension as you do.

```
# cp -p   afs.rc  /etc/rc.d/init.d/afs
```

The **afsd** options file possibly already exists as **/etc/sysconfig/afs** from running a previous version of AFS on this machine. Compare it to the version in the **root.client/usr/vice/etc** directory of the AFS 3.6 distribution to see if any changes are needed.

If the options file is not already in place, copy it now. Note the removal of the **.conf** extension as you copy.

```
# cp  -p  afs.conf  /etc/sysconfig/afs
```

If necessary, edit the options file to invoke the desired arguments on the **afsd** command in the initialization script. For further information, see the section titled *Configuring the Cache Manager* in the *IBM AFS Quick Beginnings* chapter about configuring client machines.

4. Issue the **chkconfig** command to activate the **afs** configuration variable, if it is not already. Based on the instruction in the AFS initialization file that begins with the string `#chkconfig`, the command automatically creates the symbolic links that incorporate the script into the Linux startup and shutdown sequence.

   ```
   # /sbin/chkconfig  --add afs
   ```

5. **(Optional)** If the machine is configured as a client, there are now copies of the AFS initialization file in both the **/usr/vice/etc** and **/etc/init.d** directories, and copies of the **afsd** options file in both the **/usr/vice/etc** and **/etc/sysconfig** directories. If you want to avoid potential confusion by guaranteeing that the two copies of each file are always the same, create a link between them. You can always retrieve the original script or options file from the AFS distribution if necessary.

   ```
   # cd /usr/vice/etc

   # rm afs.rc afs.conf

   # ln -s  /etc/rc.d/init.d/afs  afs.rc

   # ln -s  /etc/sysconfig/afs  afs.conf
   ```

6. Reboot the machine.

   ```
   # shutdown -r now
   ```

7. If you are upgrading a server machine, login again as the local superuser **root**, then return to Step 6 on page 32 in "Upgrading Server Machines" on page 30.

   ```
   login: root
   Password: root_password
   ```

Loading AFS into the Solaris Kernel

The **modload** program is the dynamic kernel loader provided by Sun Microsystems for Solaris systems. Solaris does not support incorporation of AFS modifications during a kernel build.

For AFS to function correctly, the **modload** program must run each time the machine reboots, so the AFS initialization script (included on the AFS CD-ROM) invokes it automatically. In this section you copy the appropriate AFS library file to the location where the **modload** program accesses it and then run the script.

1. Access the AFS distribution by changing directory as indicated. Substitute **sun4x_56** or **sun4x_57** for the *sysname* variable.
   - If you copied the contents of the **root.client** directory into AFS (in Step 6 on page 28 of "Storing Binaries in AFS" on page 28), change directory as indicated.

     ```
     # cd /afs/cellname/sysname/usr/afsws/root.client
     ```

   - If copying files from the CD-ROM, mount the CD-ROM for this machine's system type on the local **/cdrom** directory. For instructions on mounting CD-ROMs (either locally or remotely via NFS), consult the operating system documentation. Then change directory as indicated.

     ```
     # cd /cdrom/sysname/root.client
     ```

   - If accessing the distribution electronically, you possibly already downloaded it in "Storing Binaries in AFS" on page 28. If so, it is still in the *temp_afs36_dir* directory. If not, download it and run any commands necessary to uncompress or unpack the distribution. Place it in a temporary location (*temp_afs36_dir*), and change directory to the indicated subdirectory.

     ```
     # cd  temp_afs36_dir/root.client
     ```

2. If this machine is running Solaris 2.6 or the 32–bit version of Solaris 7, and ran that operating system with AFS 3.5, the appropriate AFS initialization file possibly already exists as **/etc/init.d/afs**. Compare it to the version in the **root.client/usr/vice/etc** directory of the AFS 3.6 distribution to see if any changes are needed.

   If this machine is running the 64–bit version of Solaris 7, the AFS initialization file differs from the AFS 3.5 version. Copy it from the AFS 3.6 distribution.

   Note the removal of the **.rc** extension as you copy.

   ```
   # cd  usr/vice/etc

   # cp  -p  afs.rc  /etc/init.d/afs
   ```

3. Copy the appropriate AFS kernel library file to the appropriate file in a subdirectory of the local **/kernel/fs** directory.

If the machine is running Solaris 2.6 or the 32–bit version of Solaris 7 and is not to act as an NFS/AFS translator:

```
# cp -p modload/libafs.nonfs.o /kernel/fs/afs
```

If the machine is running Solaris 2.6 or the 32–bit version of Solaris 7 and is to act as an NFS/AFS translator, in which case its kernel must support NFS server functionality and the **nfsd** process must be running:

```
# cp -p modload/libafs.o /kernel/fs/afs
```

If the machine is running the 64–bit version of Solaris 7 and is not to act as an NFS/AFS translator:

```
# cp -p modload/libafs64.nonfs.o /kernel/fs/sparcv9/afs
```

If the machine is running the 64–bit version of Solaris 7 and is to act as an NFS/AFS translator, in which case its kernel must support NFS server functionality and the **nfsd** process must be running:

```
# cp -p modload/libafs64.o /kernel/fs/sparcv9/afs
```

4. Verify the existence of the symbolic links specified in the following commands, which incorporate the AFS initialization script into the Solaris startup and shutdown sequence. If necessary, issue the commands to create the links.

```
# ln -s ../init.d/afs /etc/rc3.d/S99afs

# ln -s ../init.d/afs /etc/rc0.d/K66afs
```

5. **(Optional)** If the machine is configured as a client, there are now copies of the AFS initialization file in both the **/usr/vice/etc** and **/etc/init.d** directories. If you want to avoid potential confusion by guaranteeing that they are always the same, create a link between them. You can always retrieve the original script from the AFS distribution if necessary.

```
# cd /usr/vice/etc

# rm afs.rc

# ln -s /etc/init.d/afs afs.rc
```

6. Reboot the machine.

```
# shutdown -i6 -g0 -y
```

7. If you are upgrading a server machine, login again as the local superuser **root**, then return to Step 6 on page 32 in "Upgrading Server Machines" on page 30.

```
login: root
Password: root_password
```

Storing AFS Documents in AFS

This section explains how to create and mount a volume to house AFS documents. The recommended mount point for the volume is **/afs/***cellname***/afsdoc**. If you ran AFS 3.5, the volume possibly already exists. You can choose to overwrite its contents with the AFS 3.6 version of documents, or can create a new volume for the AFS 3.6 documents and mount it at **/afs/***cellname***/afsdoc** instead of the volume of AFS 3.5 documents. Alter the following instructions as necessary.

If you wish, you can create a link to the mount point on each client machine's local disk, called **/usr/afsdoc**. Alternatively, you can create a link to the mount point in each user's home directory. You can also choose to permit users to access only certain documents (most probably, the *IBM AFS User Guide*) by creating different mount points or setting different ACLs on different document directories.

To create a new volume for storing AFS documents:

1. Issue the **vos create** command to create a volume for storing the AFS documentation. Include the **-maxquota** argument to set an unlimited quota on the volume.

   If you wish, you can set the volume's quota to a finite value after you complete the copying operations. At that point, use the **vos examine** command to determine how much space the volume is occupying. Then issue the **fs setquota** command to set a quota value that is slightly larger.

   ```
   % vos create <machine name> <partition name>  afsdoc  -maxquota 0
   ```

2. Issue the **fs mkmount** command to mount the new volume. If your **root.cell** volume is replicated, you must precede the *cellname* with a period to specify the read/write mount point, as shown. Then issue the **vos release** command to release a new replica of the **root.cell** volume, and the **fs checkvolumes** command to force the local Cache Manager to access them.

   ```
   % fs mkmount -dir /afs/.cellname/afsdoc -vol afsdoc

   % vos release root.cell

   % fs checkvolumes
   ```

3. Issue the **fs setacl** command to grant the **rl** permissions to the **system:anyuser** group on the new directory's ACL.

   ```
   % cd /afs/.cellname/afsdoc

   % fs setacl  .  system:anyuser rl
   ```

4. Access the documents via one of the sources listed in "Accessing the AFS Binary Distribution and Documentation" on page 2. Copy the documents in one more formats from a *source_format* directory into subdirectories of

the **/afs/***cellname***/afsdoc** directory. Repeat the commands for each format. Suggested substitutions for the *format_name* variable are **HTML** and **PDF**.

```
# mkdir format_name

# cd  format_name

# cp -rp /cdrom/Documentation/language_code/source_format  .
```

If you copy the HTML version of the documents, note that in addition to a subdirectory for each document there are several files with a **.gif** extension, which enable readers to move easily between sections of a document. The file called **index.htm** is an introductory HTML page that has a hyperlink to the documents. For HTML viewing to work properly, these files must remain in the top-level HTML directory (the one named, for example, **/afs/***cellname***/afsdoc/Html**).

5. **(Optional)** If you believe it is helpful to your users to access AFS documents via a local disk directory, create **/usr/afsdoc** on the local disk as a symbolic link to the directory housing the desired format (probably HTML or PDF).

   ```
   # ln -s /afs/cellname/afsdoc/format_name  /usr/afsdoc
   ```

   An alternative is to create a link in each user's home directory to the documentation directory in AFS.

Reference Pages

Following are reference pages that include new information not included in *IBM AFS Administration Reference*.

CFG_*tcid*

Purpose

Defines Tape Coordinator configuration instructions for automated tape devices, backup data files, or XBSA server programs

Description

A **CFG**_*tcid* file includes instructions that configure a Tape Coordinator for more automated operation and for transferring AFS data to and from a certain type of backup media:

- An automated tape device, such as a stacker or jukebox. The file is optional for a Tape Coordinator that writes to such a device, and unnecessary if the default value for all types of instruction are appropriate for the device.
- A *backup data file* on a local disk device. The configuration file is mandatory and must include the **FILE** instruction at least.
- A third-party backup utility that implements the Open Group's Backup Service API (XBSA), hereafter referred to as an *XBSA server*. The file is mandatory and must include the **SERVER**, **TYPE**, and **PASSFILE** or **PASSWORD** instructions. The General Availability release of AFS 3.6 can communicate with one XBSA server, the Tivoli Storage Manager (TSM).

The configuration file is in ASCII-format and must reside in the **/usr/afs/backup** directory on the Tape Coordinator machine. Each Tape Coordinator has its own configuration file (multiple Tape Coordinators cannot use the same file), and only a single Tape Coordinator in a cell can write to a given tape device or backup data file. Multiple Tape Coordinators can interact with the same XBSA server if the server has sufficient capacity, and in this case the configuration file for each Tape Coordinator refers to the same XBSA server.

The Tape Coordinator for a tape device or backup data file must also have an entry in the Backup Database and in the **/usr/afs/backup/tapeconfig** file on the Tape Coordinator machine. The Tape Coordinator for an XBSA server has only an entry in the Backup Database, not in the **tapeconfig** file.

Naming the Configuration File

For a Tape Coordinator that communicates with an XBSA server, the *tcid* portion of the configuration file's name is the Tape Coordinator's port offset number as defined in the Backup Database. An example filename is **CFG_22**.

For the Tape Coordinator for a tape device or backup data file, there are two possible types of values for the *tcid* portion of the filename. The Tape Coordinator first attempts to open a file with a *tcid* portion that is the Tape Coordinator's port offset number as defined in the Backup Database and **tapeconfig** file. If there is no such file, the Tape Coordinator attempts to

access a file with a *tcid* portion that is based on the tape device's device name the backup data file's filename. To enable the Tape Coordinator to locate the file, construct the *tcid* portion of the filename 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**.

Summary of Instructions

The following list briefly describes the instructions that can appear in a configuration file. Each instruction appears on its own line, in any order. Unless otherwise noted, the instructions apply to all backup media (automated tape device, backup data file, and XBSA server). 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. Does not apply to XBSA servers.

BUFFERSIZE
Sets the size of the memory buffer the Tape Coordinator uses when dumping data to or restoring data from a backup medium.

CENTRALLOG
Names a log file in which to record a status message as each dump or restore operation completes. The Tape Coordinator also writes to its standard log and error files.

FILE Determines whether the Tape Coordinator uses a backup data file as the backup medium.

GROUPID
Sets an identification number recorded in the Backup Database for all dumps performed by the Tape Coordinator.

LASTLOG
Controls whether the Tape Coordinator creates and writes to a separate log file during its final pass through the set of volumes to be included in a dump.

MAXPASS
Specifies how many times the Tape Coordinator attempts to access a volume during a dump operation if the volume is inaccessible on the first attempt (which is included in the count).

MGMTCLASS
Specifies which of an XBSA server's management classes to use, which often indicates the type of backup medium the XBSA server uses. Applies only to XBSA servers.

MOUNT
Identifies the file that contains routines for inserting tapes into a tape device or controlling how the Tape Coordinator handles a backup data file. Does not apply to XBSA servers.

NAME_CHECK
Controls whether the Tape Coordinator verifies that a tape or backup data file has the expected name. Does not apply to XBSA servers.

NODE
Names which node associated with an XBSA server to use. Applies only to XBSA servers.

PASSFILE
Names the file that contains the password or security code for the Tape Coordinator to pass to an XBSA server. Applies only to XBSA servers.

PASSWORD
Specifies the password or security code for the Tape Coordinator to pass to an XBSA server. Applies only to XBSA servers.

SERVER
Names the XBSA server machine with which the Tape Coordinator communicates. Applies only to XBSA servers.

STATUS
Controls how often the Tape Coordinator writes a status message in its window during an operation.

TYPE Defines which XBSA-compliant program (third-party backup utility) is running on the XBSA server. Applies only to XBSA servers.

UNMOUNT
Identifies the file that contains routines for removing tapes from a tape device or controlling how the Tape Coordinator handles a backup data file. Does not apply to XBSA servers.

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**_*tcid* 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 its **/usr/afs/backup/TE**_*tcid* 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 instruction does not apply to XBSA servers.

The BUFFERSIZE Instruction

The **BUFFERSIZE** instruction takes an integer or decimal value, and optionally units, in the following format:

```
BUFFERSIZE size[{k | K | m | M | g | G | t | T}]
```

where *size* specifies the amount of memory the Tape Coordinator allocates to use as a buffer during both dump and restore operations. If *size* is a decimal number, the number of digits after the decimal point must not translate to fractions of bytes. The default unit is bytes, but use **k** or **K** to specify kilobytes, **m** or **M** for megabytes, **g** or **G** for gigabytes, and **t** or **T** for terabytes. There is no space between the *size* value and the units letter.

As the Tape Coordinator receives volume data from the Volume Server during a dump operation, it gathers the specified amount of data in the buffer before transferring the entire amount to the backup medium. Similarly, during a restore operation the Tape Coordinator by default buffers data from the backup medium before transferring the entire amount to the Volume Server for restoration into the file system.

The default buffer size is 16 KB, which is usually large enough to promote tape streaming in a normal network configuration. If the network connection between the Tape Coordinator machine and file server machines is slow, it can help to increase the buffer size.

For XBSA servers, the range of acceptable values is **1K** through **64K**. For tape devices and backup data files, the minimum acceptable value is **16K**, and if the specified value is not a multiple of 16 KB, the Tape Coordinator automatically rounds it up to the next such multiple.

The CENTRALLOG Instruction

The **CENTRALLOG** instruction takes a pathname as its argument, in the following format:

```
CENTRALLOG  filename
```

where *filename* is the full pathname of a local disk file in which to record a status message as each dump or restore operation completes. It is acceptable to have multiple Tape Coordinators write to the same log file. Each Tape Coordinator also writes to its own standard error and log files (the **TE**_*tcid* and **TL**_*tcid* files in the **/usr/afs/backup** directory). This instruction is always optional.

The line for each dump operation has the following format:

```
task_ID   start_time   complete_time   duration  volume_set  \
      success of total volumes dumped (data_dumped KB)
```

The line for each restore operation has the following format:

```
task_ID   start_time   complete_time   duration  success of total volumes restored
```

where

task_ID
: Is the task identification number assigned to the operation by the Tape Coordinator. The first digits in the number are the Tape Coordinator's port offset number.

start_time
: The time at which the operation started, in the format *month*/*day*/*year* *hours*:*minutes*:*seconds*.

complete_time
: Is the time at which the operation completed, in the same format as the *start_time* field.

duration
: Is the amount of time it took to complete the operation, in the format *hours*:*minutes*:*seconds*.

volume_set
: Is the name of the volume set being dumped during this operation (for dump operations only).

success Is the number of volumes successfully dumped or restored.

total Is the total number of volumes the Tape Coordinator attempted to dump or restore.

data_dumped
Is the number of kilobytes of data transferred to the backup medium (for dump operations only).

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** and the **SERVER** instruction does not appear in the configuration file, the Tape Coordinator uses a tape device as the backup medium. If the **SERVER** instruction does appear, the Tape Coordinator communicates with the XBSA server that it names. This is the default behavior if the **FILE** instruction does not appear in the file.

When the value is **YES**, the Tape Coordinator uses a backup data file on the local disk as the backup medium. If the file does not exist when the Tape Coordinator attempts 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 is accidently dumped to a tape device while the **FILE** instruction is set to **YES**. (In the same way, if the **FILE** instruction is set to **NO** and there is no **SERVER** instruction, 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's name in this field. This configuration has a couple of advantages:

- It makes the *tcid* portion of the **CFG**_*tcid*, **TE**_*tcid*, and **TL**_*tcid* 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 GROUPID Instruction

The **GROUPID** instruction takes an integer as its argument, in the following format:

```
GROUPID integer
```

where *integer* is in the range from **1** through **2147483647** (one less than 2 GB). The value is recorded in the Backup Database record for each dump created by this Tape Coordinator. It appears in the `Group id` field in the output from the **backup dumpinfo** command when the command's **-verbose** and **-id** options are provided. It can be specified as the value of the **-groupid** argument to the **backup deletedump** command to delete only records marked with the group ID. This instruction is always optional.

The LASTLOG Instruction

The **LASTLOG** instruction takes a boolean value as its argument, in the following format:

```
LASTLOG  {YES | NO}
```

When the value is **YES**, the Tape Coordinator creates and writes to a separate log file during the final pass through the volumes to be included in a dump operation. The log file name is **/usr/afs/backup/TL**_*tcid*__**lp**, where *tcid* is either the Tape Coordinator's port offset number or a value derived from the device name or backup data filename.

When the value is **NO**, the Tape Coordinator writes to its standard log files (the **TE**_*tcid* and **TL**_*tcid* files in the **/usr/afs/backup** directory) for all passes. This is the behavior if the instruction does not appear in the file.

The MAXPASS Instruction

The **MAXPASS** instruction takes an integer as its argument, in the following format:

```
MAXPASS integer
```

where *integer* specifies how many times the Tape Coordinator attempts to access a volume during a dump operation if the volume is inaccessible on the first attempt (which is included in the count). Acceptable values are in the range from **1** through **10**. The default value is **2** if this instruction does not appear in the file.

The MGMTCLASS Instruction

The **MGMTCLASS** instruction takes a character string as its argument, in the following format:

```
MGMTCLASS class_name
```

where *class_name* is the XBSA server's management class, which often indicates the type of backup medium it is using. For a list of the possible management classes, see the XBSA server documentation. This instruction applies only to XBSA servers and is always optional; there is no default value if it is omitted.

The MOUNT Instruction

The **MOUNT** instruction takes a pathname as its argument, in the following format:

```
MOUNT filename
```

where *filename* is the full pathname of an executable file on the local disk that contains a shell script or program (for clarity, the following discussion refers to scripts only). If the configuration file is for an automated tape device, the script invokes the routine or command provided by the device's manufacturer for mounting a tape (inserting it into the tape reader). If the configuration file is for a backup data file, it can instruct the Tape Coordinator to switch automatically to another backup data file when the current one becomes full; for further discussion, see the preceding description of the **FILE** instruction. This instruction does not apply to XBSA servers.

The administrator must write the script, including the appropriate routines and logic. The AFS distribution does not include any scripts, although an example appears in the following **Examples** section. The command or routines invoked by the script inherit the local identity (UNIX UID) and AFS tokens of the **butc** command's issuer.

When the Tape Coordinator needs to mount a tape or access another backup data file, it checks the configuration file for a **MOUNT** instruction. If there is no 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 script.

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.

The Tape Coordinator passes the following parameters to the script indicated by the **MOUNT** instruction, in the indicated order:

1. The tape device or backup data file's pathname, as recorded in the **/usr/afs/backup/tapeconfig** file.
2. The tape operation, which generally matches the **backup** command operation code used to initiate the operation (the following list notes the exceptional cases) :
   - **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**

3. 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.
4. 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).
5. 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 or opened the backup data file. 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 or open the backup data file. The Tape Coordinator terminates the operation.
- Any other code indicates that the routine was not able to access the correct tape or backup data file. 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 there is no permanent name on the label of the tape or backup data file, the Tape Coordinator checks the AFS tape name on the label when dumping a volume in response to the **backup dump** command. The AFS tape name must be `<NULL>` or match the name that the

backup dump operation constructs 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. This instruction does not apply to XBSA servers.

The NODE Instruction

The **NODE** instruction takes a character string as its argument, in the following format:

```
NODE node_name
```

where *node_name* names the node associated with the XBSA server named by the **SERVER** instruction. To determine if the XBSA server uses nodes, see its documentation. This instruction applies only to XBSA servers, and there is no default if it is omitted. However, TSM requires that a NODENAME instruction appear in its **dsm.sys** configuration file in that case.

The PASSFILE Instruction

The **PASSFILE** instruction takes a pathname as its argument, in the following format:

```
PASSFILE filename
```

where *filename* is the full pathname of a file on the local disk that records the password for the Tape Coordinator to use when communicating with the XBSA server. The password string must appear on the first line in the file, and have a newline character only at the end. The mode bits on the file must enable the Tape Coordinator to read it.

This instruction applies only to XBSA servers, and either it or the **PASSWORD** instruction must be provided along with the **SERVER** instruction. (If both this instruction and the **PASSWORD** instruction are included, the Tape Coordinator uses only the one that appears first in the file.)

The PASSWORD Instruction

The **PASSWORD** instruction takes a character string as its argument, in the following format:

```
PASSWORD string
```

where *string* is the password for the Tape Coordinator to use when communicating with the XBSA server. It must appear on the first line in the file, and have a newline character only at the end.

This instruction applies only to XBSA servers, and either it or the **PASSFILE** instruction must be provided along with the **SERVER** instruction. (If both this instruction and the **PASSFILE** instruction are included, the Tape Coordinator uses only the one that appears first in the file.)

The SERVER Instruction

The **SERVER** instruction takes a character string as its argument, in the following format:

```
SERVER machine_name
```

where *machine_name* is the fully qualified hostname of the machine where an XBSA server is running. This instruction is required for XBSA servers, and applies only to them.

The STATUS Instruction

The **STATUS** instruction takes an integer as its argument, in the following format:

```
STATUS integer
```

where *integer* expresses how often the Tape Coordinator writes a status message to its window during an operation, in terms of the number of buffers of data that have been dumped or restored. Acceptable values range from **1** through **8192**. The size of the buffers is determined by the **BUFFERSIZE** instruction if it is included.

As an example, the value **512** means that the Tape Coordinator writes a status message after each 512 buffers of data. It also writes a status message as it completes the dump of each volume.

The message has the following format:

```
time_stamp: Task task_ID: total KB: volume: volume_total B
```

where

time_stamp
: Records the time at which the message is printed, in the format *hours*:*minutes*:*seconds*.

task_ID
Is the task identification number assigned to the operation by the Tape Coordinator. The first digits in the number are the Tape Coordinator's port offset number.

total Is the total number of kilobytes transferred to the backup medium during the current dump operation.

volume Names the volume being dumped as the message is written.

volume_total
Is the total number of bytes dumped so far from the volume named in the *volume* field.

This instruction is intended for use with XBSA servers. For tape devices and backup data files, the value in the *volume_total* field is not necessarily as expected. It does not include certain kinds of Backup System metadata (markers at the beginning and end of each volume, for example), so summing together the final *volume_total* value for each volume does not necessarily equal the running total in the *total* field. Also, the Tape Coordinator does not write a message at all if it is dumping metadata rather than actual volume data as it reaches the end of the last buffer in each set of *integer* buffers.

The TYPE Instruction

The **TYPE** instruction takes a character string as its argument, in the following format:

```
TYPE program_name
```

where *program_name* names the XBSA server program that is running on the machine named by the **SERVER** instruction. This instruction is mandatory when the **SERVER** instruction appears in the file. The acceptable values depend on which XBSA servers are supported in the current AFS release. In the General Availability release of AFS 3.6, the only acceptable value is **tsm**.

The UNMOUNT Instruction

The **UNMOUNT** instruction takes a pathname as its argument, in the following format:

```
UNMOUNT filename
```

where *filename* is the full pathname of an executable file on the local disk that contains a shell script or program (for clarity, the following discussion refers to scripts only). If the configuration file is for an automated tape device, the script invokes the routine or command provided by the device's manufacturer for unmounting a tape (removing it from the tape reader). If the configuration

file is for a backup data file, it can instruct the Tape Coordinator to perform additional actions after closing the backup data file. This instruction does not apply to XBSA servers.

The administrator must write the script, including the appropriate routines and logic. The AFS distribution does not include any scripts, although an example appears in the following **Examples** section. The command or routines invoked by the script inherit the local identity (UNIX UID) and AFS tokens of the **butc** command's issuer.

After closing a tape device or backup data file, 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_*tcid* 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 the preceding **Description** section.

```
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} > 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 the operation being performed is not one of those explicitly mentioned in the file (such as a restore operation).

Example CFG_*tcid* File for Dumping to a Backup 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 chooses to name the configuration file **/usr/afs/backup/CFG_20**, using the port offset number rather than deriving the *tcid* portion of the name from the backup data file's name. She includes the following lines in the file. To review the meaning of each instruction, see the preceding **Description** section.

```
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} > 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.

Example CFG_*tcid* File for an XBSA Server

The following is an example of a configuration file called **/usr/afs/backup/CFG_22**, for a Tape Coordinator with port offset 22 that communicates with an Tivoli Storage Management (TSM) server. The combination of **BUFFERSIZE** and **STATUS** instructions results in a status message after each 16 MB of data are dumped. To review the meaning of the other instructions, see the preceding **Description** section.

```
SERVER tsmserver1.abc.com
TYPE tsm
PASSWORD  TESTPASS
NODE testnode
MGMTCLASS standard
MAXPASS 1
```

```
GROUPID 1000
CENTRALLOG /usr/afs/backup/centrallog
BUFFERSIZE 16K
STATUS 1024
```

Related Information

tapeconfig

backup deletedump

backup diskrestore

backup dump

backup dumpinfo

backup restoredb

backup savedb

backup volrestore

backup volsetrestore

NetRestrict (client version)

Purpose
Defines client interfaces not to register with the File Server

Description
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 the reference page for the client version of the **NetInfo** file.

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.

To display the addresses the Cache Manager is currently registering with File Servers, use the **fs getclientaddrs** command.

Related Information
NetInfo (client version)

fs getclientaddrs

NetRestrict (server version)

Purpose

Defines interfaces that File Server does not register in VLDB and Ubik does not use for database server machines

Description

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.

To display the File Server interface addresses registered in the VLDB, use the **vos listaddrs** command.

Related Information

NetInfo (server version)

sysid

vldb.DB0 and **vldb.DBSYS1**

fileserver

vos listaddrs

backup deletedump

Purpose

Deletes one or more dump records from the Backup Database

Synopsis

```
backup deletedump [-dumpid <dump id>+]  [-from <date time>+]  [-to <date time>+]
                  [-port <TC port offset>]  [-groupid <group ID>]
                  [-dbonly]  [-force]  [-noexecute]
                  [-localauth]  [-cell <cell name>]  [-help]

backup dele [-du <dump id>+]  [-fr <date time>+]  [-t <date time>+]
            [-p <TC port offset>]  [-g <group ID>]  [-db]  [-fo]  [-n]
            [-l]  [-c <cell name>]  [-h]
```

Description

The **backup deletedump** command deletes one or more dump records from the Backup Database. Using this command is appropriate when dump records are incorrect (possibly because a dump operation was interrupted or failed), or when they represent dumps that are expired or otherwise no longer needed.

To specify the records to delete, use one of the following arguments or combinations of arguments:

- The **-dumpid** argument deletes the record for each specified dump ID number.
- The **-groupid** argument deletes each record with the specified group ID number. A group ID number is associated with a record if the **GROUPID** instruction appears in the Tape Coordinator's **/usr/afs/backup/CFG**_*tcid* file when the dump is created. To display a dump set's group ID, include the **-verbose** and **-id** options to the **backup dumpinfo** command; the group ID appears in the output's `Group id` field.
- The **-from** and **-to** arguments delete the records for all regular dumps created during the time period bracketed by the specified values. The **-from** argument can be omitted, in which case the command deletes records created before the time specified by the **-to** argument.
- The combination of the **-groupid**, **-to** and optionally **-from** arguments deletes the records for all regular dumps created during the specified time period that are also marked with the specified group ID number.

The command can also delete dump records maintained by an XBSA server at the same time as the corresponding Backup Database records. (An *XBSA server* is a third-party backup utility that implements the Open Group's Backup Service API [XBSA].) Include the **-port** argument to identify the Tape Coordinator that communicates with the XBSA server. To delete the Backup Database records without attempting to delete the records at the XBSA server,

include the **-dbonly** flag. To delete the Backup Database records even if an attempt to delete the records at the XBSA server fails, include the **-force** flag.

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 dumps appended to the initial dump.

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 a dump's record makes it impossible to restore data from it or from any dump that refers to the deleted dump as its parent, directly or indirectly. That is, restore operations must begin with a 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. If necessary, use the **-dbadd** flag to the **backup scantape** command to regenerate a dump record so that the dump can act as a parent again.

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

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, the **-to** (and optionally **-from**) argument, or the **-groupid** argument.

-from 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.

Note: A plus sign follows this argument in the command's syntax statement because it accepts a multiword value which does not need to be enclosed in double quotes or other delimiters, not because it accepts multiple dates. Provide only one date (and optionally, time) definition.

-to Specifies the end of a range of dates; the record of any regular 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, the **-dumpid** argument, or the **-groupid** argument, or combine this argument and the **-groupid** 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).

Note: A plus sign follows this argument in the command's syntax statement because it accepts a multiword value which does not need to be enclosed in double quotes or other delimiters, not because it accepts multiple dates. Provide only one date (and optionally, time) definition.

-port Specifies the port offset number of the Tape Coordinator that communicates with the XBSA server that maintains the records to

delete. It must be the Tape Coordinator that transferred AFS data to the XBSA server when the dump was created. The corresponding records in the Backup Database are also deleted.

This argument is meaningful only when deleting records maintained by an XBSA server. Do not combine it with the **-dbonly** flag. If this argument is omitted when other options pertinent to an XBSA server are included, the Tape Coordinator with port offset 0 (zero) is used.

-groupid
Specifies the group ID number that is associated with the records to delete. The Tape Coordinator ignores group IDs if this argument is omitted.

Provide either this argument, the **-dumpid** argument, or the **-to** argument, or combine this argument and the **-to** argument with any options other than the **-dumpid** argument.

-dbonly
Deletes records from the Backup Database without attempting to delete the corresponding records maintained by an XBSA server. Do not combine this flag with the **-port** argument or the **-force** flag.

-force Deletes the specified records from the Backup Database even when the attempt to delete the corresponding records maintained by an XBSA server fails. Do not combine this flag with the **-dbonly** flag. To identify the Tape Coordinator when this argument is used, either provide the **-port** argument or omit it to specify the Tape Coordinator with port offset 0 (zero).

-noexecute
Displays a list of the dump records to be deleted, without actually deleting them. Combine it with the options to be included on the actual 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 the introductory **backup** reference page.

-cell Names the cell in which to run the command. Do not combine this argument with the **-localauth** flag. For more details, see the introductory **backup** reference page.

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

Output

If the **-noexecute** flag is not included, the output generated at the conclusion of processing lists the dump IDs of all deleted dump records, in the following format:

```
The following dumps were deleted:
     dump ID 1
     dump ID 2
     etc.
```

If the **-noexecute** flag is included, the output instead lists the dump IDs of all dump records to be deleted, in the following format:

```
The following dumps would have been deleted:
     dump ID 1
     dump ID 2
     etc.
```

The notation `Appended Dump` after a dump ID indicates that the dump is to be deleted because it is appended to an initial dump that also appears in the list, even if the appended dump's dump ID or group ID number was not specified on the command line. For more about deleting appended dumps, see the preceding **Cautions** section of this reference page.

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 1999 and 23:59:59 hours on 31 December 1999:

```
% backup deletedump -from 01/01/1999 -to 12/31/1999
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.

Related Information

CFG_*tcid*

backup

backup dumpinfo

backup scantape

backup dumpinfo

Purpose
Displays a dump record from the Backup Database

Synopsis
```
backup dumpinfo [-ndumps <no. 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
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 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 if desired, 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 the introductory **backup** reference page.

-cell Names the cell in which to run the command. Do not combine this argument with the **-localauth** flag. For more details, see the introductory **backup** reference page.

-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

volume_set_name.*dump_level_name* (*initial_dump_ID*)

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: XBSA_program: Server: hostname
```

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:

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
Group id  = 10
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.

Related Information

backup

backup deletedump

backup status

Purpose
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
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 the introductory **backup** reference page.

-cell Names the cell in which to run the command. Do not combine this argument with the **-localauth** flag. For more details, see the introductory **backup** reference page.

-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 task_ID:  operation:  status
```

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:

```
XBSA_program 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.

Related Information

backup

butc

vos delentry

Purpose

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. Specify one or more read/write volumes; specifying a read-only or backup volume results in an error. 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

A single VLDB entry represents all versions of a volume (read/write, readonly, and backup). The command removes the entire entry even though only the read/write volume is specified.

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 Specifies the complete name or volume ID number of each read/write volume for which to remove the VLDB entry. The entire entry is removed. Provide this argument or some combination of the **-prefix**, **-server**, and **-partition** arguments.

-prefix

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

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 the introductory reference page for the **vos** command suite.

Combine this argument with the **-prefix** argument, the **-partition** argument, or both.

-partition

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 the introductory reference page for the **vos** command suite.

Combine this argument with the **-prefix** argument, the **-server** argument, or both.

-cell Names the cell in which to run the command. Do not combine this argument with the **-localauth** flag. For more details, see the introductory **vos** reference page.

-noauth
Assigns the unprivileged identity **anonymous** to the issuer. Do not combine this flag with the **-localauth** flag. For more details, see the introductory **vos** reference page.

-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 the introductory **vos** reference page.

-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 number 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**.

Related Information

vos

vos remove

vos syncserv

vos syncvldb

vos zap

Notices

First Edition (April 2000)

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some

states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the document. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
ATTN: Software Licensing
11 Stanwix Street
Pittsburgh, PA 15222-1312
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy

of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices show are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machine Corporation in the United States, other countries, or both:

AFS
AIX
AS/400
CICS
CICS OS/2
CICS/400
CICS/6000
CICS/ESA
CICS/MVS
CICS/VSE
CICSPlex
DB2
DCE Encina Lightweight Client
DFS
Encina
IBM
IMS
MQSeries
MVS/ESA
OS/2
OS/390
OS/400
PowerPC
RISC System/6000
RS/6000
S/390
Transarc
TXSeries
VSE/ESA
VTAM
VisualAge
WebSphere

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark in the United States, other countries or both and is licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Readers' Comments — We'd Like to Hear from You

AFS
Release Notes
Version 3.6

Publication No. GC09-4558-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
GC09-4558-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
ATTN: File Systems Documentation Group
11 Stanwix Street
Pittsburgh, PA
15222-1312

Fold and Tape **Please do not staple** Fold and Tape

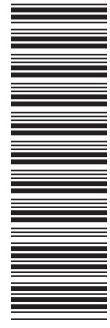
GC09-4558-00

Cut or Fold
Along Line

Part Number: 0000000
Program Number:

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

(1P) P/N: 0000000



GC09-4558-00

Spine information:

AFS Release Notes Version 3.6