DCE 1.2.2 DFS Administration Guide and Reference

OSF® DCE Product Documentation

The Open Group

OSF® DCE Product Documentation:

DCE 1.2.2 DFS Administration Guide and Reference, (Volume 1)
ISBN 1–85912–123–3
Document Number F209A

DCE 1.2.2 DFS Administration Guide and Reference, (Volume 2)
ISBN 1–85912–128–4
Document Number F209B

Published in the U.K. by The Open Group, 1997.

Any comments relating to the material contained in this document may be submitted to:

The Open Group
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:
OGPubs@opengroup.org

OTHER NOTICES

Contents

Part 2. DCE 1.2.2 DFS Administration Reference

List of Figures

List of Tables

Preface

The Open Group

The Open Group is the leading vendor-neutral, international consortium for buyers and suppliers of technology. Its mission is to cause the development of a viable global information infrastructure that is ubiquitous, trusted, reliable, and as easy-to-use as the telephone. The essential functionality embedded in this infrastructure is what we term the IT DialTone. The Open Group creates an environment where all elements involved in technology development can cooperate to deliver less costly and more flexible IT solutions.

Formed in 1996 by the merger of the X/Open Company Ltd. (founded in 1984) and the Open Software Foundation (founded in 1988), The Open Group is supported by most of the world's largest user organizations, information systems vendors, and software suppliers. By combining the strengths of open systems specifications and a proven branding scheme with collaborative technology development and advanced research, The Open Group is well positioned to meet its new mission, as well as to assist user organizations, vendors, and suppliers in the development and implementation of products supporting the adoption and proliferation of systems which conform to standard specifications.

With more than 200 member companies, The Open Group helps the IT industry to advance technologically while managing the change caused by innovation. It does this by:

- consolidating, prioritizing, and communicating customer requirements to vendors
- conducting research and development with industry, academia, and government agencies to deliver innovation and economy through projects associated with its Research Institute
- managing cost-effective development efforts that accelerate consistent multi-vendor deployment of technology in response to customer requirements
- adopting, integrating, and publishing industry standard specifications that provide an essential set of blueprints for building open information systems and integrating new technology as it becomes available
- licensing and promoting the Open Brand, represented by the "X" mark, that designates vendor products which conform to Open Group Product Standards
- promoting the benefits of IT DialTone to customers, vendors, and the public.

The Open Group operates in all phases of the open systems technology lifecycle including innovation, market adoption, product development, and proliferation. Presently, it focuses on seven strategic areas: open systems application platform development, architecture, distributed systems management, interoperability, distributed computing environment, security, and the information superhighway. The Open Group is also responsible for the management of the UNIX trademark on behalf of the industry.

The Development of Product Standards

This process includes the identification of requirements for open systems and, now, the IT DialTone, development of CAE and Preliminary Specifications through an industry consensus review and adoption procedure (in parallel with formal standards work), and the development of tests and conformance criteria.

This leads to the preparation of a Product Standard which is the name used for the documentation that records the conformance requirements (and other information) to which a vendor may register a product. There are currently two forms of Product

Standard, namely the Profile Definition and the Component Definition, although these will eventually be merged into one.

The “X” mark is used by vendors to demonstrate that their products conform to the relevant Product Standard. By use of the Open Brand they guarantee, through the X/Open Trade Mark License Agreement (TMLA), to maintain their products in conformance with the Product Standard so that the product works, will continue to work, and that any problems will be fixed by the vendor.

Open Group Publications

The Open Group publishes a wide range of technical documentation, the main part of which is focused on specification development and product documentation, but which also includes Guides, Snapshots, Technical Studies, Branding and Testing documentation, industry surveys, and business titles.

There are several types of specification:

CAE Specifications

CAE (Common Applications Environment) Specifications are the stable specifications that form the basis for our Product Standards, which are used to develop X/Open branded systems. These specifications are intended to be used widely within the industry for product development and procurement purposes.

Anyone developing products that implement a CAE Specification can enjoy the benefits of a single, widely supported industry standard. Where appropriate, they can demonstrate product compliance through the Open Brand. CAE Specifications are published as soon as they are developed, so enabling vendors to proceed with development of conformant products without delay.

Preliminary Specifications

Preliminary Specifications usually address an emerging area of technology and consequently are not yet supported by multiple sources of stable conformant implementations. They are published for the purpose of validation through implementation of products. A Preliminary Specification is not a draft specification; rather, it is as

stable as can be achieved, through applying The Open Group's rigorous development and review procedures.

Preliminary Specifications are analogous to the trial-use standards issued by formal standards organizations, and developers are encouraged to develop products on the basis of them. However, experience through implementation work may result in significant (possibly upwardly incompatible) changes before its progression to becoming a CAE Specification. While the intent is to progress Preliminary Specifications to corresponding CAE Specifications, the ability to do so depends on consensus among Open Group members.

Consortium and Technology Specifications

The Open Group publishes specifications on behalf of industry consortia. For example, it publishes the NMF SPIRIT procurement specifications on behalf of the Network Management Forum. It also publishes Technology Specifications relating to OSF/1, DCE, OSF/Motif, and CDE.

Technology Specifications (formerly AES Specifications) are often candidates for consensus review, and may be adopted as CAE Specifications, in which case the relevant Technology Specification is superseded by a CAE Specification.

In addition, The Open Group publishes:

Product Documentation

This includes product documentation—programmer's guides, user manuals, and so on—relating to the Prestructured Technology Projects (PSTs), such as DCE and CDE. It also includes the Single UNIX Documentation, designed for use as common product documentation for the whole industry.

Guides

These provide information that is useful in the evaluation, procurement, development, or management of open systems, particularly those that relate to the CAE Specifications. The Open Group Guides are advisory, not normative, and should not be referenced for purposes of specifying or claiming conformance to a Product Standard.

Technical Studies

Technical Studies present results of analyses performed on subjects of interest in areas relevant to The Open Group's Technical Program. They

are intended to communicate the findings to the outside world so as to stimulate discussion and activity in other bodies and the industry in general.

Versions and Issues of Specifications

As with all live documents, CAE Specifications require revision to align with new developments and associated international standards. To distinguish between revised specifications which are fully backwards compatible and those which are not:

- A new Version indicates there is no change to the definitive information contained in the previous publication of that title, but additions/extensions are included. As such, it replaces the previous publication.
- A new Issue indicates there is substantive change to the definitive information contained in the previous publication of that title, and there may also be additions/extensions. As such, both previous and new documents are maintained as current publications.

Corrigenda

Readers should note that Corrigenda may apply to any publication. Corrigenda information is published on the World-Wide Web at *http://www.opengroup.org/public/pubs*.

Ordering Information

Full catalogue and ordering information on all Open Group publications is available on the World-Wide Web at *http://www.opengroup.org/public/pubs*.

This Book

The *DCE 1.2.2 DFS Administration Guide and Reference* serves two purposes:

- It provides concepts and procedures that enable you to manage the Distributed File Service (DFS) in your Distributed Computing Environment (DCE) cell.
- It provides detailed reference information to help you learn more about the complete syntax and use of each DFS command and configuration file.

Audience

This guide and reference is written for system and network administrators who have previously administered a UNIX environment.

Applicability

This revision applies to the OSF® DCE Revision 1.2.2 offering. See your software license for details.

Purpose

The purpose of this guide and reference is to help system and network administrators plan, configure, and manage DFS in a DCE cell. After you have initially installed and configured DCE and DFS in your cell, refer to this document for information about expanding and maintaining your DFS configuration. Also refer to this document for complete descriptions of all DFS commands. The *DCE 1.2.2 Release Notes* contain instructions for installing and building DCE source code, and they contain release-specific information about DFS.

Document Usage

The *DCE 1.2.2 DFS Administration Guide and Reference* is divided into the following parts:

- Volume 1
  Document Number F209A, ISBN 1–85912–123–3
  - — Part 1. DCE 1.2.2 DFS Administration Guide
    - Chapter 1. An Overview of DFS
    - Chapter 2. DFS Configuration Issues
    - Chapter 3. Using ACLs and Groups
    - Chapter 4. Using Administrative Lists and Keytab Files
    - Chapter 5. Monitoring and Controlling Server Processes
    - Chapter 6. Making Filesets and Aggregates Available
    - Chapter 7. Managing Filesets
    - Chapter 8. Configuring the Cache Manager
    - Chapter 9. Configuring the Backup System
    - Chapter 10. Backing Up and Restoring Data
    - Chapter 11. Monitoring and Tracing Tools
- Volume 2
  Document Number F209B, ISBN 1–85912–128–4
  - — Part 2. DCE 1.2.2 DFS Administration Reference
    - Chapter 12. Configuration Files
    - Chapter 13. Administrative Commands
  - — Appendix A. The DFS/NFS Secure Gateway

Related Documents

For additional information about the Distributed Computing Environment, refer to the following documents:

- *DCE 1.2.2 Introduction to OSF DCE*
  Document Number F201, ISBN 1–85912–182–9
- *DCE 1.2.2 Command Reference*
  Document Number F212, ISBN 1–85912–138–1
- *DCE 1.2.2 Application Development Reference*
  Document Number F205A, ISBN 1–85912–103–9 (Volume 1)
  Document Number F205B, ISBN 1–85912–108–X (Volume 2)
  Document Number F205C, ISBN 1–85912–159–4 (Volume 3)
- *DCE 1.2.2 Administration Guide—Introduction*
  Document Number F207, ISBN 1–85912–113–6
- *DCE 1.2.2 Administration Guide—Core Components*
  Document Number F208, ISBN 1–85912–118–7
- *DCE 1.2.2 Application Development—Introduction and Style Guide*
  Document Number F202, ISBN 1–85912– 187–X
- *DCE 1.2.2 Application Development Guide—Core Components*
  Document Number F203A, ISBN 1–85912–192–6 (Volume 1)
  Document Number F203B, ISBN 1–85912–154–3 (Volume 2)
- *DCE 1.2.2 Application Development Guide—Directory Services*
  Document Number F204, ISBN 1–85912–197–7
- *DCE 1.2.2 GDS Administration Guide and Reference*
  Document Number F211, ISBN 1–85912–133–0
- *DCE 1.2.2 File-Access Administration Guide and Reference*
  Document Number F216, ISBN 1–85912–158–6
- *DCE 1.2.2 File-Access User's Guide*
  Document Number F217, ISBN 1–85912–163–3
- *DCE 1.2.2 Testing Guide*
  Document Number F215, ISBN 1–85912–153–5
- *DCE 1.2.2 File-Access FVT User's Guide*
  Document Number F210, ISBN 1–85912–189–6

- *DCE 1.2.2 Release Notes*
  Document Number F218, ISBN 1–85912–168–3

Typographic and Keying Conventions

This guide uses the following typographic conventions:

Bold — **Bold** words or characters represent system elements that you must use literally, such as commands, options, and pathnames.

Italic — *Italic* words or characters represent variable values that you must supply. *Italic* type is also used to introduce a new DCE term.

`Constant width` — Examples and information that the system displays appear in `constant width` typeface.

[ ] — Brackets enclose optional items in format and syntax descriptions.

{ } — Braces enclose a list from which you must choose an item in format and syntax descriptions.

| — A vertical bar separates items in a list of choices.

< > — Angle brackets enclose the name of a key on the keyboard.

... — Horizontal ellipsis points indicate that you can repeat the preceding item one or more times.

This guide uses the following keying conventions:

<Ctrl-*x***>** or ^*x* — The notation **<Ctrl-***x***>** or ^*x* followed by the name of a key indicates a control character sequence. For example, **<Ctrl-C>** means that you hold down the control key while pressing **<C>**.

<Return> — The notation **<Return>** refers to the key on your terminal or workstation that is labeled with the word Return or Enter, or with a left arrow.

Problem Reporting

If you have any problems with the software or vendor-supplied documentation, contact your software vendor's customer service department. Comments relating to this Open Group document, however, should be sent to the addresses provided on the copyright page.

Pathnames of Directories and Files in DCE Documentation

For a list of the pathnames for directories and files referred to in this guide, see the *DCE 1.2.2 Administration Guide—Introduction* and *DCE 1.2.2 Testing Guide*.

Trademarks

Part 2

DCE 1.2.2 DFS Administration Reference

Chapter 12

Configuration Files

dfs_intro

Purpose

dfs_intro – Introduction to DFS files

Description

DFS includes a number of system-specific files. These files can be grouped into the following general categories:

Configuration files
: Define configuration parameters for specific server and kernel processes such as a Tape Coordinator or Cache Manager.

Administrative lists
: List the principals (users, groups, and servers) allowed to access specific server processes, including the Backup Server, the Basic OverSeer Server, the Fileset Server, the Fileset Location Server, and the Update Server.

Cache-related files
: Contain cached data or information about cached data.

Log files
: Contain output from specific processes or commands.

Specific information about the files, such as pathnames and format, is included with the reference pages that describe them. Most of the files are referred to in Part 1 of this manual. Refer to Part 1 for more information about these files and the DFS components and commands with which they are associated.

Related Information

Following is a list of all relevant DFS files for which reference pages are included. See the DCE DFS portion of this reference for information about any of the commands referenced in these pages.

- Configuration files:

  BosConfig(4dfs)

CacheInfo(4dfs)

dfstab(4dfs)

NoAuth(4dfs)

TapeConfig(4dfs)

- Administrative files:

  admin.bak(4dfs)

  admin.bos(4dfs)

  admin.fl(4dfs)

  admin.ft(4dfs)

  admin.up(4dfs)

- Cache-related files:

  CacheItems(4dfs)

  FilesetItems(4dfs)

  Vn(4dfs)

- Log files:

  BakLog(4dfs)

  BosLog(4dfs)

  DfsgwLog(4dfs)

FlLog(4dfs)

FMSLog(4dfs)

FtLog(4dfs)

RepLog(4dfs)

TE(4dfs)

TL(4dfs)

UpLog(4dfs)

BakLog

Purpose

BakLog – Contains messages generated by the Backup Server

Description

The **BakLog** file contains execution and error messages generated by the Backup Server (**bakserver** process). The Backup Server runs on every Backup Database machine in a cell, providing the interface by which authorized users can modify the Backup Database.

If the **BakLog** file does not already exist when the Backup Server starts, the server process creates the file in the directory named *dcelocal*/**var/dfs/adm**. The process then appends any subsequent messages to the file once it exists. If the file exists when the Backup Server starts, the process moves the current version of the file to the **BakLog.old** file in the same directory (overwriting the current **BakLog.old** file if it exists) before creating a new version to which to append messages.

The process can write different types of output to the file, depending on the actions it performs and any problems it encounters. The file can be viewed with the **bos getlog** command. Because it is an ASCII file, it can also be viewed with the **more** command (or a similar command appropriate to the local operating system), which requires **read** permission on the file.

Events are recorded in the log file only at their completion, so the process does not use the file to reconstruct failed operations. However, the contents of the log file can help in evaluating server process failures and other problems.

Related Information

Commands: **bakserver(8dfs)**, **bos getlog(8dfs)**.

BosConfig

Purpose

BosConfig – Defines server processes to be monitored by the Basic OverSeer (BOS) Server

Description

The **BosConfig** file defines the server processes to be monitored by the BOS Server (**bosserver** process) on a server machine. It contains a process entry for each process to be monitored by the BOS Server; each entry defines how its process is to run. The **BosConfig** file also maintains both the weekly and daily restart times for the BOS Server and processes that have entries in the file.

The BOS Server runs on each server machine, continually monitoring and, if necessary, restarting the other server processes on the machine. The BOS Server checks the **BosConfig** file whenever it starts or restarts; the information is then transferred into memory and the file is not read again until the BOS Server restarts. Thus, server processes can be started or stopped, independently of their process entries, based on their status in the BOS Server's memory. The order in which process entries appear in the **BosConfig** file is irrelevant.

The **BosConfig** file must reside in the directory named *dcelocal*/**var**/**dfs** on the local disk of a server machine running the BOS Server. The BOS Server creates a **BosConfig** file with only default restart times and no process entries if the file does not exist when the BOS Server starts. Because it is a local file, the information it contains can be different for different machines.

Each process entry in a **BosConfig** file includes the following information about the process:

Name
: This is the name used by the BOS Server to refer to the process. Although any name can be chosen, the following names are recommended for consistency:

  ftserver For the Fileset Server process

  flserver For the Fileset Location Server process

  upclient For the client portion of the Update Server

upserver For the server portion of the Update Server

repserver For the Replication Server process

bakserver For the Backup Server process

Type A process can be one of two types:

simple A continuous process that runs independently of any other processes on a server machine. All standard DFS processes are **simple** processes.

cron A process that runs independently of any other processes; however, unlike a **simple** process, a **cron** process runs periodically, not continuously.

Status flag Status flags are for internal use only; they do not appear in any output. A process can have one of two status flags:

Run Means the process is to run whenever possible; the BOS Server starts it automatically at reboot and restarts it automatically if it fails. (The **Run** status flag appears in the file as a **1**.)

NotRun Means the BOS Server does not start or restart the process. (The **NotRun** status flag appears in the file as a **0**.)

Command parameters The BOS Server uses these parameters to run the process. For a **simple** process, a single command parameter specifying the complete pathname of the binary file for a DFS command or any other command to be executed is used. For a **cron** process, two command parameters are used: the complete pathname of the binary file for a DFS command or any other command to be executed, and the time the BOS Server is to execute the command.

Although it is an ASCII file, do not edit the **BosConfig** file directly; always use the appropriate **bos** commands. Editing the file directly can introduce changes the BOS Server does not recognize until it is restarted and again reads the file.

The following **bos** commands modify process entries or restart times in the **BosConfig** file:

bos create Adds a process entry to the file, setting the process' status to **Run** in both the file and memory, and starts the process

bos delete Removes a process entry for a stopped process from the file

bos stop Stops a running process by changing its status to **NotRun** in both the file and memory

bos start Starts a stopped process by changing its status to **Run** in both the file and memory

bos setrestart
Sets the weekly and daily restart times included in the file

The following **bos** commands access process entries in the **BosConfig** file:

bos status Lists the statuses of server processes on a machine, from which you can determine information about their process entries

bos restart Stops and immediately restarts processes that have process entries in the file

bos getrestart
Displays both the weekly and daily restart times from the file

Additional **bos** commands can be used to start or stop a process by changing its status in the BOS Server's memory without affecting its process entry in the **BosConfig** file.

Cautions

Do not edit the **BosConfig** file directly. Always use the appropriate **bos** commands to manipulate process entries in the **BosConfig** file. Editing the file directly can introduce changes that the BOS Server is not aware of until it is restarted and again reads the file.

Examples

The following **bos create** command creates a process entry in the **BosConfig** file and starts the process. The command adds the process entry to the **BosConfig** file on the server machine named **fs1.abc.com**. It specifies that a **cron** process identified by **backup** is to use the **fts clonesys** command daily at 5:30 a.m. to create backup versions of all read/write filesets on **fs1.abc.com**. The **-localauth** option is used with the **fts clonesys** command to use the identity of the local machine as the identity of the issuer of the command.

```
$ bos create /.../abc.com/hosts/fs1 backup cron "dcelocal/bin/fts clonesys \
-server /.../abc.com/hosts/fs1 -localauth" 5:30
```

The following **bos setrestart** command sets the general restart time when the BOS Server restarts itself and all of the processes with entries in the **BosConfig** file. It specifies that all processes, including the **bosserver** process, on **fs1.abc.com** are to be restarted every Sunday morning at 4:00 a.m.

```
$ bos setrestart /.../abc.com/hosts/fs1 -general "sun 4:00"
```

Related Information

Commands: **bos create(8dfs)**, **bos delete(8dfs)**, **bos setrestart(8dfs)**, **bos start(8dfs)**, **bos stop(8dfs)**, **bosserver(8dfs)**.

BosLog

Purpose

BosLog – Contains messages generated by the Basic OverSeer (BOS) Server

Description

The **BosLog** file contains execution and error messages generated by the Basic OverSeer (BOS) Server (**bosserver** process). The BOS Server runs on every server machine in a cell, monitoring the other server processes on the machine and restarting them as necessary.

If the **BosLog** file does not already exist when the BOS Server starts, the server process creates the file in the directory named *dcelocal*/**var/dfs/adm**. The process then appends any subsequent messages to the file once it exists. If the file exists when the BOS Server starts, the process moves the current version of the file to the **BosLog.old** file in the same directory (overwriting the current **BosLog.old** file if it exists) before creating a new version to which to append messages.

The process can write different types of output to the file, depending on the actions it performs and any problems it encounters. The file can be viewed with the **bos getlog** command. Because it is an ASCII file, it can also be viewed with the **more** command (or a similar command appropriate to the local operating system), which requires **read** permission on the file.

Events are recorded in the log file only at their completion, so the process does not use the file to reconstruct failed operations. However, the contents of the log file can help you evaluate server process failures and other problems.

Related Information

Commands: **bos getlog(8dfs)**, **bosserver(8dfs)**.

CacheInfo

Purpose

CacheInfo – Defines the initial configuration of the Cache Manager

Description

The **CacheInfo** file specifies the initial configuration of the Cache Manager on a client machine. The Cache Manager checks the file at initialization to determine certain cache configuration information. It uses the file regardless of the type of caching, disk or memory, in use on the machine.

The **CacheInfo** file is manually created during DFS client installation. (See Part 1 of this manual for details on creating the file.) It must reside in the directory named */dcelocal* **/etc**.

The file is a one-line ASCII file consisting of the following three fields separated by colons:

- The first field names a directory on the local disk where the Cache Manager mounts the DCE global namespace. The default entry is the global namespace designation (**/...**). If **/...** is not specified, symbolic links to the global namespace fail.

  The value of this field can be overridden with the **dfsd** command and the **-mountdir** option.

- The second field names a directory on the local disk to serve as the cache directory for a disk cache. This is the directory in which the Cache Manager stores the **V** *n*, **CacheItems**, and **FilesetItems** files that it creates. The default entry is */dcelocal/* **var/adm/dfs/cache**. You can change this to a directory on another partition if more space is available elsewhere. Although the indicated directory is not used with a memory cache, an entry must appear in this field even if memory caching is employed on the machine.

  The value of this field can be overridden with the **dfsd** command and the **-cachedir** option.

- The third field specifies the cache size in 1024-byte (1-kilobyte) blocks. The amount of disk space or machine memory used for caching depends on several

factors. The size of the partition housing the cache directory or the amount of memory available on the machine places an absolute limit on the cache size. However, do not use more than 90% of the cache directory's partition for a disk cache, and do not use more than 20 to 25% of available memory for a memory cache.

The value of this field can be overridden with the **dfsd** command and the **-blocks** option. It can also be overridden with the **cm setcachesize** command. The **cm getcachesize** command can be used to view the current size of the cache and the amount in use.

Because it is an ASCII file, the **CacheInfo** file can be directly modified with a text editor. To modify the file, log in as **root** on the machine.

Cautions

The size of the partition housing the cache directory or the amount of memory available on the machine places an absolute limit on the cache size. However, do not use more than 90% of the cache directory's partition for a disk cache, and do not use more than 20 to 25% of available memory for a memory cache.

Be precise when editing the **CacheInfo** file; use colons to separate the fields in the file, and do not include any spaces in the file. Improper formatting of this file can cause the kernel to panic.

Examples

An example of a typical **CacheInfo** file follows. It lists the DCE global namespace mounted at the global namespace designation (**/...**), */.../dcelocal/var/adm/dfs/cache* used for the cache directory, and a defined cache size of 50,000 1-kilobyte blocks.

```
/...:/.../dcelocal/var/adm/dfs/cache:50000
```

Related Information

Commands: **cm getcachesize(8dfs)**, **cm setcachesize(8dfs)**, **dfsd(8dfs)**.

Files: **CacheItems(4dfs)**, **FilesetItems(4dfs)**, **Vn(4dfs)**.

CacheItems

Purpose

CacheItems – Records information about each V file in a disk cache

Description

The **CacheItems** file is a binary file created and maintained by the Cache Manager for its own use and for use by developers for debugging. It records information about each V file on a client machine using a disk cache. The information includes the file ID number and data version number of each V file.

The **CacheItems** file always resides in the cache directory with the cache's V files. The default directory for the files is *dcelocal* **/var/adm/dfs/cache**. This directory is specified in the second field of the **CacheInfo** file; it can be overridden to name a different directory.

Cautions

Never directly modify or delete the **CacheItems** file; this can cause the kernel to panic. Always use the commands provided with DFS to alter the cache. If the file is accidentally modified or deleted, rebooting the machine should restore normal performance.

Related Information

Files: **CacheInfo(4dfs)**, **Vn(4dfs)**.

DfsgwLog

Purpose

DfsgwLog – Contains messages generated by the Gateway Server process of the DFS/NFS Secure Gateway

Description

The **DfsgwLog** file contains messages generated by the Gateway Server (**dfsgwd**) process. The Gateway Server process runs on machines configured as DFS clients to allow users to authenticate to DCE from NFS clients.

If the **DfsgwLog** file does not already exist when the Gateway Server process starts, the process creates the file in the directory named *dcelocal*/**var/dfs/adm**. Once the file exists, the process appends messages to it. If the file exists when the Gateway Server process starts, the process moves the current version of the file to the **DfsgwLog.old** file in the same directory (overwriting the current **DfsgwLog.old** file if it exists) before creating a new version to which to append messages.

The process can write different types of output to the file, depending on the actions it performs and any problems it encounters. The file can be viewed with the **bos getlog** command. Because it is an ASCII file, it can also be viewed with the **more** command (or a similar command appropriate to the local operating system), which requires **read** permission on the file.

Events are recorded in the log file only at their completion, so the process does not use the file to reconstruct failed operations. However, the contents of the log file can help in evaluating server process failures and other problems.

Related Information

Commands: **bos getlog(8dfs)**, **dfsgwd(8dfs)**.

FMSLog

Purpose

FMSLog – Lists the output of the **fms** command

Description

The **FMSLog** file lists the output generated by the **fms** (file mark size) command. The **fms** command determines the tape capacity and end-of-file (EOF) mark size for a tape drive. The command both displays its output on the screen and writes it to the **FMSLog** file, which it creates in the directory from which it is issued.

The command creates the **FMSLog** file if it does not already exist in the current working directory, in which case the issuer of the command must have write, execute, and insert permissions on the directory from which the command is issued. If the file already exists in the current working directory, the command reinitializes the file (clears its contents) before writing to it, in which case the issuer needs only write permission on the file.

The information written to the **FMSLog** file is useful for specifying a tape drive's configuration parameters in the **TapeConfig** file on a Tape Coordinator machine. The **FMSLog** file is an ASCII file, so it can be viewed with the **more** command (or a similar command appropriate to the local operating system), which requires read permission on the file.

The tape size reported in the file should be reduced by 10 to 15% before being used in the **TapeConfig** file. The EOF mark size in the file should be increased by 10 to 15% before being used in the **TapeConfig** file.

The **FMSLog** file is not created if a problem with the tape drive prevents execution of the **fms** command.

Examples

An example of the **FMSLog** file follows. The file lists the tape capacity and the size of the EOF mark for the tape drive used in the **fms** command. The tape drive used in the example uses tapes 2,136,604,672 bytes in size, and creates EOF marks of size 1,910,220 bytes in size.

```
fms test started
wrote 130408 blocks
Tape capacity is 2136604672 bytes
File marks are 1910220 bytes
```

Related Information

Commands: **fms(8dfs)**.

Files: **TapeConfig(4dfs)**.

FilesetItems

Purpose

FilesetItems – Records location mappings for filesets accessed by a Cache Manager using a disk cache

Description

The **FilesetItems** file is a binary file created and maintained by the Cache Manager for its own use and for use by developers for debugging. It stores the fileset-to-mount point mapping for each fileset accessed by a Cache Manager using a disk cache. The mappings enable the Cache Manager to respond correctly to operating system and related commands such as **pwd**.

The **FilesetItems** file always resides in the cache directory. The default directory is *dcelocal*/**var/adm/dfs/cache**. This directory is specified in the second field of the **CacheInfo** file; it can be overridden to name a different directory.

Cautions

Never directly modify or delete the **FilesetItems** file; this can cause the kernel to panic. Always use the commands provided with DFS to alter the cache. If the file is accidentally modified or deleted, rebooting the machine should restore normal performance.

Related Information

Files: **CacheInfo(4dfs)**.

FlLog

Purpose

FlLog – Contains messages generated by the Fileset Location Server

Description

The **FlLog** file contains execution messages and error messages generated by the Fileset Location Server (**flserver** process). The Fileset Location Server runs on every Fileset Database machine in a cell, providing the interface by which authorized users can modify the Fileset Location Database (FLDB).

If the **FlLog** file does not already exist when the Fileset Location Server starts, the server process creates the file in the directory named *dcelocal*/**var/dfs/adm**. The process then appends any subsequent messages to the file once it exists. If the file exists when the Fileset Location Server starts, the process moves the current version of the file to the **FlLog.old** file in the same directory (overwriting the current **FlLog.old** file if it exists) before creating a new version to which to append messages.

The process can write different types of output to the file, depending on the actions it performs and any problems it encounters. The file can be viewed with the **bos getlog** command. Because it is an ASCII file, it can also be viewed with the **more** command (or a similar command appropriate to the local operating system), which requires read permission on the file.

Events are recorded in the log file only at their completion, so the process does not use the file to reconstruct failed operations. However, the contents of the log file can help in evaluating server process failures and other problems.

Related Information

Commands: **bos getlog(8dfs)**, **flserver(8dfs)**.

FtLog

Purpose

FtLog – Contains messages generated by the Fileset Server

Description

The **FtLog** file contains execution messages and error messages generated by the Fileset Server (**ftserver** process). The Fileset Server runs on every File Server machine in a cell. It provides the interface for any commands that affect filesets on a File Server machine.

If the **FtLog** file does not already exist when the Fileset Server starts, the server process creates the file in the directory named *dcelocal*/**var/dfs/adm**. The process then appends any subsequent messages to the file once it exists. If the file exists when the Fileset Server starts, the process moves the current version of the file to the **FtLog.old** file in the same directory (overwriting the current **FtLog.old** file if it exists) before creating a new version to which to append messages.

The process can write different types of output to the file, depending on the actions it performs and any problems it encounters. The file can be viewed with the **bos getlog** command. Because it is an ASCII file, it can also be viewed with the **more** command (or a similar command appropriate to the local operating system), which requires **read** permission on the file.

Events are recorded in the log file only at their completion, so the process does not use the file to reconstruct failed operations. However, the contents of the log file can help in evaluating server process failures and other problems.

Related Information

Commands: **bos getlog(8dfs)**, **ftserver(8dfs)**.

NoAuth

Purpose

NoAuth – Indicates that DFS authorization checking is disabled

Description

The **NoAuth** file is a zero-length file that dictates whether DFS authorization checking is enabled or disabled on a server machine. The presence of the **NoAuth** file in the *dcelocal* **/var/dfs** directory on a local disk indicates to all DFS server processes on that machine that DFS authorization checking is disabled. All DFS server processes, including the BOS Server, check for the presence of the file when they are requested to perform an operation; they do not check for the necessary administrative privilege for a requested operation when the file is present.

When the **NoAuth** file is present in *dcelocal* **/var/dfs** on a server machine, DFS authorization checking is disabled on that machine. The server processes on the machine perform any action for any user who requests it, including the unprivileged identity **nobody**. This is a serious security risk and should be used only in the following situations:

- During initial DFS installation
- If the Security Service is unavailable
- During server encryption key emergencies
- To view the actual keys stored in a keytab file

When the **NoAuth** file is *not* present in *dcelocal***/var/dfs** on a server machine, DFS authorization checking is enabled on that machine. All DFS server processes on the machine check that the issuer of a command has the proper authorization (is included in the necessary administrative lists) before they perform the requested operation. By default, DFS authorization checking is always enabled on every server machine.

The **bos status** command can be used to determine whether DFS authorization checking is enabled or disabled on a server machine. The command displays the following message if DFS authorization checking is disabled on a machine. (It does not display the message if DFS authorization checking is enabled.)

```
Bosserver reports machine is not checking authorization.
```

The BOS Server on a server machine creates the **NoAuth** file when an authorized user (one listed in the **admin.bos** file on the machine) executes the **bos setauth** command with the **-authchecking** option set to **off**. (The file can also be created with the **-noauth** option of the **bosserver** command used to start the BOS Server.) The BOS Server removes the file when a user executes the **bos setauth** command with the **-authchecking** option set to **on**. Whenever the **bos setauth** command is used to change the state of DFS authorization checking, all server processes immediately recognize the changed state and respond accordingly to any subsequent commands.

Cautions

Always use the **bos setauth** command to create the *dcelocal*/**var/dfs/NoAuth** file. Do not create the file directly except when explicitly told to do so by instructions for dealing with emergencies (such as server encryption key emergencies). Creating the file directly requires logging into the local operating system of a machine as **root** and using the **touch** command (or its equivalent).

Related Information

Commands: **bos setauth(8dfs)**, **bos status(8dfs)**, **bosserver(8dfs)**.

RepLog

Purpose

RepLog – Contains messages generated by the Replication Server

Description

The **RepLog** file contains execution messages and error messages generated by the Replication Server (**repserver** process). The Replication Server runs on every File Server machine in a cell, allowing read-only replicas of filesets to be stored on any File Server machine.

If the **RepLog** file does not already exist when the Replication Server starts, the server process creates the file in the directory named *dcelocal*/**var**/**dfs**/**adm**. The process then appends any subsequent messages to the file once it exists. If the file exists when the Replication Server starts, the process moves the current version of the file to the **RepLog.old** file in the same directory (overwriting the current **RepLog.old** file if it exists) before creating a new version to which to append messages.

The process can write different types of output to the file, depending on the actions it performs and any problems it encounters. The file can be viewed with the **bos getlog** command. Because it is an ASCII file, it can also be viewed with the **more** command (or a similar command appropriate to the local operating system), which requires read permission on the file.

Events are recorded in the log file only at their completion, so the process does not use the file to reconstruct failed operations. However, the contents of the log file can help you evaluate server process failures and other problems.

Related Information

Commands: **bos getlog(8dfs)**, **repserver(8dfs)**.

TE

Purpose **TE_** *device_name* – Lists error messages from the **butc** process

DESCRIPTION

The **TE_** *device_name* file lists error messages generated by the **butc** (Backup Tape Coordinator) process. The **butc** process initializes a Tape Coordinator on a Tape Coordinator machine (a machine having a tape drive and an associated Tape Coordinator). The **butc** program prompts for new tapes (and displays some additional output) and, if the value set with the **butc** command's **-debuglevel** option is 1, displays information about restore operations on the screen.

The **butc** process also writes error messages to an ASCII file named **TE_** *device_name*, where *device_name* is the device name of the tape drive with which the process is associated. The file is located in the directory named *dcelocal***/var/dfs/backup** on the local disk of the Tape Coordinator machine. Messages written to the file by the process describe any problems the process encountered while executing an operation; for instance, it can include the names of any filesets the process was unable to include in a dump operation.

Each time the **butc** process is started for a tape drive and Tape Coordinator pair, it automatically creates the error file. It then appends any messages to the file once it exists. If the file already exists when the **butc** process is started, the process moves the current version of the file to the **TE_***device_name* **.old** file in the same directory (overwriting the current **TE_** *device_name***.old** file if one exists) before creating a new version to which to append messages. In either case, the issuer of the **butc** command must have write and execute permissions on the directory *dcelocal* **/var/dfs/backup**. (The process also writes execution information it generates to the *dcelocal* **/var/dfs/backup/TL_***device_name* file, which it maintains exactly as it does the **TE_** *device_name* file.)

EXAMPLES

The following example displays an error file generated by the **butc** process for a tape drive whose device name is **/dev/rmt1h**. The file, named *dcelocal***/var/dfs/backup/**

TE_rmt1h (the log file associated with this tape drive is named **TL_rmt1h**), shows routine error messages generated during a typical execution of the **butc** process. The messages that follow indicate that three dump sets were not added to the Backup Database; messages also indicate why each dump set was not added to the database (in all three cases, dump sets having the specified dump IDs already existed). The **bak scantape** command was used to attempt to add the dump sets to the database.

```
Thu Aug 22 10:52:49 1991
Dump id 681664660 not added to database
Thu Aug 22 10:52:49 1991
DFS:bakserver : dump with specified id already exists
Thu Aug 22 10:52:49 1991
Dump id 681749283 not added to database
Thu Aug 22 10:52:49 1991
DFS:bakserver : dump with specified id already exists
Thu Aug 22 10:52:49 1991
Dump id 681657088 not added to database
Thu Aug 22 10:52:49 1991
DFS:bakserver : dump with specified id already exists
```

RELATED INFORMATION

Commands: **butc(8dfs)**.

Files: **TL(4dfs)**.

TL

Purpose

TL_ *device_name* – Lists execution information from the **butc** process

Description

The **TL_** *device_name* file is a log file containing execution messages generated by the **butc** (Backup Tape Coordinator) process. The **butc** process initializes a Tape Coordinator on a Tape Coordinator machine (a machine having a tape drive and an associated Tape Coordinator). The **butc** program prompts for new tapes (and displays some additional output) and, if the value set with the **butc** command's **-debuglevel** option is 1, displays information about restore operations on the screen.

The **butc** process also writes output to an ASCII file named **TL_** *device_name*, where *device_name* is the device name of the tape drive with which the process is associated. The file is located in the directory named *dcelocal*/**var/dfs/backup** on the local disk of the Tape Coordinator machine. Output written to the file by the process provides information about all operations the process executes, from its startup to its shutdown. The level of detail to which each operation is described depends upon the operation; some operations are described in more detail than others.

Each time the **butc** process is started for a tape drive and Tape Coordinator pair, it automatically creates the log file. It then appends any messages to the file once it exists. If the file already exists when the **butc** process is started, the process moves the current version of the file to the **TL_** *device_name* **.old** file in the same directory (overwriting the current **TL_** *device_name***.old** file if one exists) before creating a new version to which to append messages. In either case, the issuer of the **butc** command must have write and execute permissions on the directory *dcelocal* **/var/dfs/backup**. (The process also writes any error messages it generates to the *dcelocal* **/var/dfs/backup/TE_** *device_name* file, which it maintains exactly as it does the **TL_** *device_name* file.)

Examples

The following example displays a log file generated by the **butc** process for a tape drive with the device name **/dev/rmt1h**. The file is named *dcelocal*/**var/dfs/backup/**

TL_rmt1h (the error file associated with this tape drive is named **TE_rmt1h**); it shows routine status messages generated during a typical execution of the **butc** process. The process is executed with the **-debuglevel** set to 0 (zero) on a Tape Coordinator whose TCID is 1.

```
Thu Aug 22 10:45:02 1991
10:45:02 Starting tape coordinator: TCID 1, debug level: 0,
     cell: /.../abc.com
10:45:15 Reading tape label .. 10:45:28 Done
10:46:02 Labelling tape size 153600 .. 10:46:31 Done
10:46:57 Labelling tape ftfamily1.month.1 size 153600 .. 10:47:25 Done
10:49:23 Database dump aborted
10:50:08 Labelling tape size 153600 .. 10:51:46 Done
10:52:25 Database successfully dumped on Thu Aug 22 10:52:25 1991
10:54:37 Reading tape label .. 10:54:48 Done
10:55:16 Labelling tape size 153600 .. 10:55:45 Done
```

Related Information

Commands: **butc(8dfs)**.

Files: **TE(4dfs)**.

TapeConfig

Purpose

TapeConfig – Defines configuration parameters for tape drives on a Tape Coordinator machine

Description

The **TapeConfig** file includes configuration information about all of the Tape Coordinators running on a Tape Coordinator machine. A **TapeConfig** file must reside in the directory named *dcelocal* **/var/dfs/backup** on each Tape Coordinator machine.

The **TapeConfig** file must contain a single line specifying information about each tape drive and its associated Tape Coordinator. It must contain a line for each tape drive whose Tape Coordinator is to be started with the **butc** command. Otherwise, the **butc** process cannot start the Tape Coordinator for the drive.

The **TapeConfig** file is an ASCII file. Each line specifies the following parameters for a tape drive:

Tape size
: The Tape Coordinator uses this capacity whenever a tape is used in the drive.

  The unit of measurement to be applied to the tape size can be specified as k or K (for kilobytes), m or M (for megabytes), or g or G (for gigabytes); do not leave a space between the number and letter used as a unit identifier. The default unit is kilobytes. You should use a number 10 to 15% lower than the actual tape capacity for the tape size.

End-of-file (EOF) mark size
: The Backup System appends an EOF mark of this size after each fileset dump on a tape. The size of the mark can affect the amount of space available for backup data.

  The EOF mark size can vary from 2 kilobytes to 2 megabytes, depending on the type of tape drive used. Use the same abbreviations used for tape capacity to specify the unit of measurement for the EOF mark size. The default unit is *bytes* (not kilobytes, as for tape capacity). You should increase the file mark size by 10 to 15% to allow for tape variations.

If you do not know the EOF mark size for a tape drive, use the **fms** command to determine the EOF size. This command produces screen output and an **FMSLog** file listing the tape capacity and EOF mark size for the drive.

Device name
: The name of the tape drive. The format of this name varies with each operating system.

Tape Coordinator ID (TCID)
: The identifier of the Tape Coordinator associated with the drive.

  Legal values are the integers 0 through 1023. The Backup System can track a maximum of 1024 tape drives; a single machine can house any number of drives.

  TCIDs can be specified in any order; it is not necessary to assign them sequentially. Because the **bak** commands that require you to specify a TCID always use a default TCID of 0 (zero), assign a TCID of 0 (zero) to the Tape Coordinator for the drive you will use most often.

Because it is an ASCII file, the **TapeConfig** file can be created or modified with a text editor. Creating the file requires write and execute permissions on the *dcelocal*/**var/dfs/backup** directory. Editing the file requires write permission on the file. Be precise when editing the file; a tape drive will be inaccessible if its line in the **TapeConfig** file is specified incorrectly.

Examples

An example of a **TapeConfig** file for a Tape Coordinator machine follows. The file configures three tape drives on a machine. The first drive, whose device name is **/dev/rmth0h**, has a tape size of 1 gigabyte and an EOF mark size of 4 kilobytes; its associated Tape Coordinator has a TCID of 0. The second two drives, **/dev/rmth3h** and **/dev/rmth4h**, each have tape sizes of 2 gigabytes and EOF mark sizes of 1 megabyte; the TCIDs of their respective Tape Coordinators are 3 and 2.

```
1G 4K /dev/rmth0h 0
2g 1M /dev/rmth3h 3
2G 1m /dev/rmth4h 2
```

Related Information

Commands: **butc(8dfs)**, **fms(8dfs)**.

Files: **FMSLog(4dfs)**.

UpLog

Purpose

UpLog – Contains messages generated by the server portion of the Update Server

Description

The **UpLog** file contains execution and error messages generated by the server portion (**upserver** process) of the Update Server. The **upserver** process distributes files from the local disk of a machine in response to requests from the client portion (**upclient** process) of the Update Server running on other machines. The **upserver** process should run on the cell's System Control machine and on the Binary Distribution machine for each CPU/operating system type.

If the **UpLog** file does not already exist when the **upserver** process starts, the server process creates the file in the directory named *dcelocal*/**var/dfs/adm**. The process then appends any subsequent messages to the file once it exists. If the file exists when the **upserver** process starts, the process moves the current version of the file to the **UpLog.old** file in the same directory (overwriting the current **UpLog.old** file if it exists) before creating a new version to which to append messages.

The process can write different types of output to the file, depending on the actions it performs and any problems it encounters. The file can be viewed with the **bos getlog** command. Because it is an ASCII file, it can also be viewed with the **more** command (or a similar command appropriate to the local operating system), which requires read permission on the file.

Events are recorded in the log file only at their completion, so the process does not use the file to reconstruct failed operations. However, the contents of the log file can help you evaluate server process failures and other problems.

Note that the **UpLog** file contains execution and error messages for the **upserver** process only; it does not log messages for the **upclient** process. A log file can be specified for use with the **upclient** process when that process is started on a client machine.

Related Information

Commands: **bos getlog(8dfs)**, **upclient(8dfs)**, **upserver(8dfs)**.

Vn

Purpose

V *n* – Contains a chunk of data cached in a disk cache

Description

A **V** *n* file, or V file, holds a chunk of cached data on a client machine that is using a disk cache. In the name of an actual V file, *n* is an integer; the name of each V file has a unique integer different from other V files on the machine (for example, **V1**, **V2**, and so on). The format of a V file depends on the format of the data it is caching: a V file containing a cached binary file has a binary format; a V file storing a cached ASCII file has an ASCII format.

Each V file always resides in the cache directory, which by default is *dcelocal*/**var/adm/dfs/cache**. This directory is specified in the second field of the **CacheInfo** file; it can be overridden to name a different directory. The **CacheItems** file in the cache directory records information about each V file, such as its file ID and data version numbers.

The number of V files, or cache chunks, depends on the size of the disk cache (specified in the third field of the **CacheInfo** file, defined with the **dfsd** command's **-blocks** option, or set with the **cm setcachesize** command). For a disk cache, the number of chunks is heuristically computed as the number of cache blocks divided by 8. You can override the default number of chunks with the **dfsd** command by using the **-files** option. Specify a positive integer not greater than 32,000.

To use a cache most effectively, issue the **du** command on the cache directory to determine the number of cache blocks used; compare this number to the number of blocks allocated to the cache. If you are not using 90% of the cache, increase the number of V files (chunks).

By default, each V file holds up to 65,536 bytes (64 kilobytes) of a cached file; files larger than 65,536 bytes are divided among multiple V files. A V file can hold only one cached element; if a cached element is smaller than the size of a V file (the chunk size), the remaining space in the V file remains unused.

You can override the default chunk size with the **dfsd** command by using the **-chunksize** option. Specify an integer between 13 and 18 to be used as an exponent

of 2; the unit of measure is bytes. For example, a value of 16 equals the default chunk size (2^{16} equals 65,536). A value less than 13 or greater than 18 sets the chunk size to the default, as does a value of 16.

Cautions

Never directly modify or delete a V file; this can cause the kernel to panic. Always use the commands provided with DFS to alter the cache. If a V file is accidentally modified or deleted, rebooting the machine should restore normal performance.

Related Information

Commands: **cm setcachesize(8dfs)**, **dfsd(8dfs)**.

Files: **CacheInfo(4dfs)**, **CacheItems(4dfs)**.

admin.bak

Purpose

admin.bak – Contains the administrative list for the Backup Server

Description

The **admin.bak** file is an administrative list of all users and groups that can issue commands in the **bak** command suite. Most commands in the **bak** command suite are used to communicate with the Backup Server. The commands are used to modify information in the Backup Database and to dump and restore data, as necessary.

A master copy of the Backup Database resides on one server machine; other server machines (optimally two) house replicated copies of the database. Any machine that houses a copy of the Backup Database is referred to as a Backup Database machine. The Backup Server, or **bakserver** process, must run on all Backup Database machines.

An **admin.bak** file must reside on each Backup Database machine. For the most part, the **admin.bak** file contains the UUIDs of users and groups. However, it must also contain the abbreviated DFS server principals of all Backup Database machines in the local cell to allow the synchronization site for the Backup Database to distribute changes to the secondary sites. The server principals can be present as members of a group included in the list.

Each time the Backup Server is started on any machine, it automatically creates the *dcelocal*/**var/dfs/admin.bak** file if the file does not already exist. You can also create the file by including the **-createlist** option with the **bos addadmin** command. Once the **admin.bak** file exists, principals and groups can be added to it with the **bos addadmin** command, and they can be removed from it with the **bos rmadmin** command. The **bos lsadmin** command can be used to list the principals and groups currently in the file. Because administrative lists are stored as binary files, you must use these commands to modify them; you cannot edit them directly.

The **admin.bak** file should be stored in the directory named *dcelocal*/**var/dfs** on each Backup Database machine. If it is stored in a different directory, the full pathname of the file must be specified when the Backup Server is started. Do not create multiple copies of the **admin.bak** file and store them in different directories on the same machine; unauthorized users may be able to use the extraneous copies to access the Backup Server.

A single version of the **admin.bak** file should be created and maintained on a System Control machine. The **upclient** processes running on the cell's Backup Database machines can then update their local copies of the file via the **upserver** process running on the System Control machine.

Independent versions of the **admin.bak** file should not be maintained on each Backup Database machine in a cell. Because the Backup Database is a Ubik database, any of the secondary sites may be obliged to assume the role of synchronization site for the Backup Database at any time. A system administrator who is listed in the **admin.bak** file on the machine housing the former synchronization site may not be listed in the **admin.bak** file on the machine housing the new synchronization site; the administrator, who could issue commands that affect the Backup Database on the former machine, may not be able to issue commands that affect the database on the new machine.

Related Information

Commands: **bakserver(8dfs)**, **bos addadmin(8dfs)**, **bos lsadmin(8dfs)**, **bos rmadmin(8dfs)**.

admin.bos

Purpose

admin.bos – Contains the administrative list for the Basic OverSeer (BOS) Server

Description

The **admin.bos** file is an administrative list of all users and groups that can use the Basic OverSeer Server (BOS Server) to manage server processes on a server machine. The **admin.bos** file usually includes the UUIDs of users and groups only; it is not necessary to add a server machine to the **admin.bos** file.

The BOS server, or **bosserver** process, runs on every DFS server machine in a domain. An **admin.bos** file must reside on each machine running the **bosserver** process.

A user must be represented in the **admin.bos** file on a machine (either directly or indirectly, through a group) to issue commands that affect the server processes on that machine (for example, to create, start, or stop processes). Because system administrators listed in the **admin.bos** file can issue **bos** commands, they can cause DFS server processes to run with DFS authorization checking disabled. Because inclusion in the **admin.bos** file gives an administrator such additional privileges, the administrators listed in the **admin.bos** file are usually a subset of the users in the administrative lists for a server machine or domain.

Each time the BOS Server is started on any machine, it automatically creates the *dcelocal*/**var/dfs/admin.bos** file if the file does not already exist. Once the file exists, principals and groups can be added to it with the **bos addadmin** command, and they can be removed from it with the **bos rmadmin** command. The **bos lsadmin** command can be used to list the principals and groups currently in the file. Because administrative lists are stored as binary files, you must use these commands to modify them; you cannot edit them directly.

The **admin.bos** file should be stored in the directory named *dcelocal*/**var/dfs** on each server machine. If it is stored in a different directory, the full pathname of the file must be specified when the BOS Server is started. Do not create multiple copies of the **admin.bos** file and store them in different directories on the same machine; unauthorized users may be able to use the extraneous copies to access the BOS Server.

It is recommended that a single version of the **admin.bos** file be created and maintained on a domain System Control machine. The **upclient** processes running on the domain's server machines can then reference the file via the **upserver** process running on the System Control machine.

Independent versions of the **admin.bos** file should not be maintained on each server machine in a domain. Doing so may result in a system administrator being permitted to manage processes on one machine but not on another.

(Note that a Private File Server machine might have a separate **admin.bos** file. The administrative users included in such a file would represent a superset of the administrative users listed in the domain's **admin.bos** file, the additional members being the users who are to administer the Private File Server machine.)

Related Information

Commands: **bos addadmin(8dfs)**, **bos lsadmin(8dfs)**, **bos rmadmin(8dfs)**, **bosserver(8dfs)**.

admin.fl

Purpose

admin.fl – Contains the administrative list for the Fileset Location (FL) Server

Description

The **admin.fl** file is an administrative list of all users and groups that can use the Fileset Location (FL) Server to modify the Fileset Location Database (FLDB). A master copy of the FLDB resides on one server machine; other server machines (usually two) house replicated copies of the database. Any machine that houses a copy of the FLDB is referred to as a Fileset Database machine. The FL Server, or **flserver** process, must run on all Fileset Database machines.

An **admin.fl** file must reside on each Fileset Database machine. For the most part, the **admin.fl** file contains the UUIDs of users and groups. However, it must also contain the abbreviated DFS server principals of all Fileset Database machines in the local cell to allow the synchronization site for the FLDB to distribute changes to the secondary sites. The server principals can be present as members of a group included in the list.

Each time the Fileset Location Server is started on any machine, it automatically creates the *dcelocal*/**var/dfs/admin.fl** file if the file does not already exist. You can also create the file by including the **-createlist** option with the **bos addadmin** command. Once the **admin.fl** file exists, principals and groups can be added to it with the **bos addadmin** command, and they can be removed from it with the **bos rmadmin** command. The **bos lsadmin** command can be used to list the principals and groups currently in the file. Because administrative lists are stored as binary files, you must use these commands to modify them; you cannot edit them directly.

The **admin.fl** file should be stored in the directory named *dcelocal*/**var/dfs** on each Fileset Database machine. If it is stored in a different directory, the full pathname of the file must be specified when the FL Server is started. Do not create multiple copies of the **admin.fl** file and store them in different directories on the same machine; unauthorized users may be able to use the extraneous copies to access the FLDB.

A single version of the **admin.fl** file should be created and maintained on a System Control machine. The **upclient** processes running on the cell's Fileset Database machines can then update their local copies of the file via the **upserver** process running on the System Control machine.

Independent versions of the **admin.fl** file should not be maintained on each Fileset Database machine in a cell. Because the FLDB is a Ubik database, any of the secondary sites may be obliged to assume the role of synchronization site for the FLDB at any time. A system administrator listed in the **admin.fl** file on the machine housing the former synchronization site may not be listed in the **admin.fl** file on the machine housing the new synchronization site. The administrator, who could issue commands that affect the FLDB on the former machine, may not be able to issue commands that affect the database on the new machine, or vice versa.

Related Information

Commands: **bos addadmin(8dfs)**, **bos lsadmin(8dfs)**, **bos rmadmin(8dfs)**, **flserver(8dfs)**.

admin.ft

Purpose

admin.ft – Contains the administrative list for the Fileset Server

Description

The **admin.ft** file is an administrative list of all principals and groups that can use the Fileset Server to manipulate filesets on a File Server machine. The **admin.ft** file includes the UUIDs of users and groups who can issue commands that affect a machine's filesets; it includes the UUIDs of servers the machine can accept filesets from.

A File Server machine is defined as any machine that exports data for use in the global namespace. The Fileset Server, or **ftserver** process, runs on every File Server machine in a domain. The **ftserver** process provides the interface for any commands that affect filesets on a File Server machine. An **admin.ft** file must reside on each machine running the **ftserver** process.

A user must be represented in the **admin.ft** file on a machine (either directly or indirectly, through a group) to issue commands that affect the filesets on a machine (for example, to create, move, delete, back up, or restore a fileset). The user must also be listed in the file in order to move filesets onto the machine from a different machine. In addition, the principal name for a server machine must be included in the **admin.ft** file on another machine if filesets are to be moved from it to the other machine.

Each time the Fileset Server is started on any machine, it automatically creates the *dcelocal*/**var/dfs/admin.ft** file if the file does not already exist. You can also create the file by including the **-createlist** option with the **bos addadmin** command.

Once the **admin.ft** file exists, principals and groups can be added to it with the **bos addadmin** command, and they can be removed from it with the **bos rmadmin** command. The **bos lsadmin** command can be used to list the principals and groups currently in the file. Because administrative lists are stored as binary files, you must use these commands to modify them; you cannot edit them directly.

The **admin.ft** file should be stored in the directory named *dcelocal*/**var/dfs** on each File Server machine. If it is stored in a different directory, the full pathname of the file must

be specified when the Fileset Server is started. Do not create multiple copies of the **admin.ft** file and store them in different directories on the same machine; unauthorized users may be able to use the extraneous copies to access the Fileset Server or to allow the File Server machine to accept filesets from unprivileged machines.

It is recommended that a single version of the **admin.ft** file be created and maintained on a domain's System Control machine. The **upclient** processes running on the domain's File Server machines can then reference the file via the **upserver** process running on the System Control machine.

Independent versions of the **admin.ft** file should not be maintained on each File Server machine in a domain. Doing so may result in a system administrator being permitted to manipulate filesets on one machine but not on another, or it may result in the administrator being able to move filesets among only some of the machines in the domain.

(Note that a Private File Server machine might have a separate **admin.ft** file. The administrative users included in such a file would represent a superset of the administrative users listed in the domain's **admin.ft** file, the additional members being the users who are to administer the Private File Server machine.)

Related Information

Commands: **bos addadmin(8dfs)**, **bos lsadmin(8dfs)**, **bos rmadmin(8dfs)**, **ftserver(8dfs)**.

admin.up

Purpose

admin.up – Contains the administrative list for the Update Server

Description

The **admin.up** file is an administrative list of all server principals that can receive copies of files using the Update Server. The **admin.up** file usually contains the UUIDs of server machines only; it is not necessary to add users or groups to the **admin.up** file.

The Update Server distributes files such as common configuration files, binary files, and administrative lists from System Control and Binary Distribution machines to the other server machines in a domain. Server machines that rely on System Control and Binary Distribution machines for these kinds of files run the **upclient** process, the client portion of the Update Server. System Control and Binary Distribution machines run the **upserver** process, the server portion of the Update Server.

Each instance of the **upclient** process frequently checks with the **upserver** process on the System Control and Binary Distribution machines to ensure that its local copies of the proper files are current. If newer versions of the files exist, the **upclient** process retrieves them from the **upserver** process and installs them in place of the outdated versions of the files. The **admin.up** file resides on machines running the **upserver** process; it specifies the machines whose **upclient** processes are permitted to obtain copies of files from the **upserver** process.

Each time the **upserver** process is started on any machine, it automatically creates the *dcelocal*/**var/dfs/admin.up** file if the file does not already exist. You can also create the file by including the **-createlist** option with the **bos addadmin** command.

Once the **admin.up** file exists, principals can be added to it with the **bos addadmin** command, and they can be removed from it with the **bos rmadmin** command. The **bos lsadmin** command can be used to list the principals currently in the file. Because administrative lists are stored as binary files, you must use these commands to modify them; you cannot edit them directly.

The **admin.up** file should be stored in the directory named *dcelocal*/**var/dfs** on each machine running the **upserver** portion of the Update Server. If it is stored in a different

directory, the full pathname of the file must be specified when the **upserver** process is started. Do not create multiple copies of the **admin.up** file and store them in different directories; unauthorized users may be able to use the extraneous copies to have the **upserver** process allow unprivileged machines to obtain copies of files.

Related Information

Commands: **bos addadmin(8dfs)**, **bos lsadmin(8dfs)**, **bos rmadmin(8dfs)**, **upclient(8dfs)**, **upserver(8dfs)**.

conf_tape_device

Purpose

Defines configuration parameters for automated backup devices

Description

The **conf_** *tape_device* file, also called the user-defined configuration file, sets parameters to configure the Tape Coordinator to use automated backup devices, such as stackers and jukeboxes. The file can also be used to configure the Tape Coordinator to control direct dumps to and restores from a file. The user-defined configuration file must reside in the *dcelocal* **/var/dfs/backup** directory and must have a name of the form **conf_** *tape_device*, where *tape_device* specifies the relevant device.

The user-defined configuration file is an ASCII file that contains configuration parameters. Each parameter is specified on a separate line. The valid parameters are as follows:

MOUNT Specifies a file that contains an executable routine. The routine can mount an automated backup device, such as a stacker or jukebox.

UNMOUNT
Specifies a file that contains an executable routine to perform tape unmount operations for an automated backup device.

ASK Forces all Backup System prompts (except the initial prompt to mount the first tape) to accept the default answers for all error cases rather than query the operator. This parameter is useful for fully automating the backup process. Valid arguments are **YES** and **NO**. The **YES** argument enables operator prompts; omitting **ASK** has the same result. The **NO** argument disables operator prompts and assumes the default responses to all error case prompts.

AUTOQUERY
Disables the initial Backup System prompt to mount the first tape. This parameter is also useful for fully automating the backup process. Valid arguments are **YES** and **NO**. The **YES** argument enables the initial prompt to mount the first tape for a dump set; omitting **AUTOQUERY** has the same result. The **NO** argument disables the prompt.

NAME_CHECK

Prevents the Backup System from checking tape names. This is a convenience setting you can use to recycle a group of tapes without first relabeling them. Valid arguments are **YES** and **NO**. The **YES** argument enables tape name checking; the Tape Coordinator verifies that each tape in the set has the name of the same dump set. Omitting **NAME_CHECK** has the same result. The **NO** argument disables tape name checking; the Tape Coordinator accepts any expired tape.

FILE Directs dump or restore operations to tape or to a specified file. Valid arguments are **YES** and **NO** . The **YES** argument directs the operations to a specified file. The **NO** argument directs the operations to a specified tape; omitting **FILE** has the result.

Do not specify the **YES** parameter when using a tape device or the **NO** parameter when referring to a file. Neither combination works.

If the Tape Coordinator needs another file to continue an operation it prompts the operator to mount another tape. You can use this pause in the operation to specify a new file by changing the pathname in the *dcelocal* **/var/dfs/backup/TapeConfig** file. After you respond to the prompt the Tape Coordinator will use the new pathname.

Because the user-defined configuration file is an ASCII file, it can be created or modified with a text editor. Creating the file requires **write** and **execute** permissions for the **/opt/dcelocal/var/dfs/backup** directory. Editing the file requires **write** permission for the file.

Examples

The following is an example of a user-defined configuration file for a stacker-type tape device. In this file, the **AUTOQUERY** parameter is used to disable the initial prompt to the operator to mount a tape. The **ASK** parameter enables prompts to the operator if errors occur. The **MOUNT** parameter refers to the **/opt/backup/stacker0.1** file, which contains an executable routine (written by the user) to control the stacker. The **NAME_CHECK** parameter prevents the Backup System from checking the names of tapes during a dump operation.

```
AUTOQUERY NO
ASK YES
```

```
MOUNT /opt/backup/stacker0.1
NAME_CHECK NO
```

In the following example, a user-defined configuration file configures the Tape Coordinator to control a jukebox. In this example, the **ASK** parameter is set to **NO** to disable error prompts. This example calls a user-defined executable routine for mounting and unmounting tapes. The **NAME_CHECK** parameter is set to **NO** so that the Tape Coordinator will accept any expired tape.

```
MOUNT /opt/backup/jukebox0.1
UNMOUNT /opt/backup/jukebox0.1
ASK NO
NAME_CHECK NO
```

Related Information

Commands: **butc(8dfs)**

Files: **TapeConfig(4dfs)**

dfstab

Purpose

LFS partitions that can be exported

Description

The **dfstab** file includes information about each DCE LFS aggregate and each non-LFS partition that can be exported from the local disk to the DCE namespace. The file is read by the **dfsexport** command, which exports specified aggregates and partitions to the DCE namespace. (It is also read by the **newaggr** command, which initializes DCE LFS aggregates.) The **dfstab** file must reside in the directory named *dcelocal*/**var/dfs**. The **dfsexport** command looks in that directory for the file; if the file is not there, no aggregates or partitions can be exported.

The **dfstab** file is an ASCII file that can be created and edited with a text editor. You must have write and execute permissions on the *dcelocal*/**var/dfs** directory to create the file. You must have write permission on the file to edit it.

The file contains a one-line entry for each aggregate or partition available for exporting. Each entry in the file must appear on its own line. The fields in the following list must appear for each entry; they must appear in the order listed, and each field must be separated by at least one space or tab. Because DCE LFS aggregates contain an arbitrary number of filesets, *do not include a fileset ID number when creating an entry for a DCE LFS aggregate*.

Device name
: The block device name of the aggregate or partition to be exported; for example, **/dev/lv02**.

Aggregate name
: The name to be associated with the aggregate or partition to be exported. An aggregate name can contain any characters, but it cannot be longer than 31 characters. It must be different from any other aggregate name in the **dfstab** file. Aggregate names cannot be abbreviated, so you should choose a short, descriptive name; for example, **lfs1**. The aggregate name of a non-LFS partition must match the name of the partition's local mount point (for example, **/usr**).

File system type
: The identifier for the type of file system housing the aggregate or partition. For DCE LFS aggregates, this must be **lfs**; for non-LFS partitions, it must be **ufs**. Enter the identifier in all lowercase letters.

Aggregate ID
: A positive integer different from any other aggregate ID in the **dfstab** file. In the entry for a non-LFS partition, this field must contain the aggregate ID number specified with the **-aggrid** option of the **fts crfldbentry** command.

Fileset ID
: The unique fileset ID number to be associated with the fileset on a non-LFS partition; for example, **0,,18756**. In the entry for a non-LFS partition, this field must contain the fileset ID number generated with the **fts crfldbentry** command. *Do not include a fileset ID number with an entry for a DCE LFS aggregate.*

When the **dfsexport** command is executed, it reads the **dfstab** file to verify that each aggregate or partition to be exported is listed in the file. An aggregate or partition must have an entry in the **dfstab** file if it is to be exported. To ensure that it does not export an aggregate or partition that is currently exported, the **dfsexport** command refers to a list of all currently exported aggregates and partitions that exists in the kernel of the local machine.

Cautions

Do not change the aggregate ID number assigned to an aggregate or partition in this file once Fileset Location Database (FLDB) entries have been created for filesets on the aggregate or partition. Changing the aggregate ID number used for an aggregate or partition in this file invalidates existing FLDB entries for filesets on the aggregate or partition.

Examples

The following **dfstab** file specifies that one non-LFS partition (**/dev/lv02**) and two DCE LFS aggregates (**/dev/lv03** and **/dev/lv04**) can be exported:

```
/dev/lv02      /usr  ufs   1  0,,18756
/dev/lv03      lfs1  lfs   3
/dev/lv04      lfs2  lfs  11
```

Related Information

Commands: **dfsexport(8dfs)**, **fts crfldbentry(8dfs)**.

Chapter 13

Administrative Commands

dfs_intro

Purpose

dfs_intro – Introduction to the DFS commands

Description

Most DFS commands are divided into the following categories, or command suites:

bak	Operates the DFS Backup System
bos	Operates the Basic OverSeer (BOS) Server
cm	Configures the Cache Manager
dfstrace	Provides DFS kernel and server process logging information
fts	Manipulates filesets

In addition, DFS provides a number of miscellaneous commands (for example, **salvage** and **scout**) not associated with a specific command suite. DFS also provides an additional command suite, **dfsgw**, that is used with the DFS/NFS Secure Gateway.

System administrators use the majority of DFS commands. However, DCE users can use the following commands:

- The **cm** commands **cm_statservers** and **cm_whereis** to determine machine, file, and directory information
- The **fts** command **fts_lsquota** to check quota information

DFS Command Types

DFS commands follow these general naming rules. Commands that begin with

- **add** or **rm** (remove) affect lists or groups of DFS objects. For example, **bos addadmin** adds an administrative user to an administrative list.
- **cr** (create) or **del** (delete) affect DFS objects. For example, **fts crserverentry** creates a DFS object, a server entry.
- **ls** (list) are used to display objects and groups of objects.

- **set** are used to assign values to parameters; for example, **fts setrepinfo** assigns replication parameters. Analogously, commands beginning with **get** are used to display parameters; for example, **cm getcachesize** displays parameters used by the Cache Manager.

Rules For Using DFS Commands

When supplying an argument to a command, the option associated with the argument can be omitted if

- All arguments supplied with the command are entered in the order in which they appear in the command's synopsis.
- Arguments are supplied for all options that precede the option to be omitted.
- All options that precede the option to be omitted accept only a single argument.
- No options, either those that accept an argument or those that do not, are supplied before the option to be omitted.

In the case where two options are presented in { | } (braces separated by a vertical bar), the option associated with the first argument can be omitted if that argument is provided; however, the option associated with the second argument is required if that argument is provided.

If it must be specified, an option can be abbreviated to the shortest possible form that distinguishes it from other options of the command. For example, the **-server** option found in many DFS commands can typically be omitted or abbreviated to be simply **-s**.

It is also valid to abbreviate a command name to the shortest form that still distinguishes it from the other command names in the suite. For example, it is acceptable to shorten the **bos install** command to **bos i** because no other command names in the **bos** command suite begin with the letter "i." However, there are three **bos** commands that begin with the letter "g": **bos getdates**, **bos getlog**, and **bos getrestart**. To remain unambiguous, they can be abbreviated to **bos getd**, **bos getl**, and **bos getr**.

The following examples illustrate three acceptable ways to enter the same **bos getlog** command.

Complete command:

$ bos getlog -server*/.../abc.com/hosts/fs1* **-file** *BosLog*

Abbreviated command name and abbreviated options:

```
$ bos getl -s/.../abc.com/hosts/fs1 -f BosLog
```

Abbreviated command name and omitted options:

```
$ bos getl/.../abc.com/hosts/fs1 BosLog
```

Note: The **dfs_login** and **dfs_logout** commands provided with the DFS/NFS Secure Gateway do not provide the shortcuts and help available with other DFS commands. See the reference pages for these two commands for information about using them.

Aliases

An alias is an alternative way of entering an existing command. Each alias is either shorter than the original command, or it is unique within the command's suite. (Because only the number of characters sufficient to uniquely identify a command need to be entered to execute the command, unique aliases require less typing.)

The **bak** suite is the only command suite with aliases. Refer to the **bak(8dfs)** reference page for a list of the **bak** commands that have aliases.

Receiving Help

There are several different ways to receive help about DFS commands. The following list summarizes the syntax for the different help options:

Reference pages for a command suite
: To view the introductory page for a command suite, enter **man** followed by the command suite:

```
$ man bak
```

Reference page for an individual command
: To view the reference page for a command in a suite, enter **man** followed by the command suite and the command name. Use an _ (underscore) to connect the command suite to the command name. *Do not use the underscore when issuing the command in DFS.*

```
$ man bak_ command
```

List of commands in a command suite
: To view a list of all commands in a command suite, enter the command suite name followed by **help**:

```
$ bak help
```

The command syntax for a single command
: To view the syntax of a specific command, enter the suite name, **help**, and the command name, in that order:

```
$ bak help command
```

In addition, all DFS commands include a **-help** option you can use to display the syntax of the command.

The **apropos** command displays the first line of the online help entry for any command that has a specified string in its name or short description; this is useful if you cannot remember the exact name of a command. If the string is more than a single word, surround it with " " (double quotes) or other delimiters; enter all strings in lowercase letters. For example, the following command produces a list of all **bos** commands with the word **create** in their names or short descriptions:

```
$ bos apropos -topic create
```

Privileges Required

The majority of DFS commands, because they are administrative in nature, require that the issuer be included in an **admin** file (for example, **admin.bos**). Some commands require that the issuer have specific permissions to access files (for example, the delete permission on a directory) or be logged in as **root** on the machine on which the command is issued. The exact privilege needed to execute each command is detailed with the command.

Cautions

Specific cautionary information is included with individual commands.

Related Information

For more information about the commands in a specific suite and a list of the commands in the suite, see the introductory page for that suite.

bak(8dfs)

bos(8dfs)

cm(8dfs)

dfstrace(8dfs)

fts(8dfs)

bak

Purpose

bak – Introduction to the **bak** command suite

Options

The following options are used with many **bak** commands; they are also listed with the commands that use them:

-server *machine*
Specifies the File Server machine to use with the command. You can use any of the following to specify the File Server machine:

- The machine's DCE pathname (for example, */.../abc.com/hosts/fs1*)
- The machine's host name (for example, **fs1.abc.com** or **fs1**)
- The machine's IP address (for example, **11.22.33.44**)

-tapehost *machine*
Specifies the machine for which a Tape Coordinator is being added. You can use the machine's DCE pathname, its host name, or its IP address.

-tcid *tc_number*
Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator being used to execute the command. Legal values for this argument are the integers 0 (zero) to 1023. Because the default for the TCID is **0** (zero), the drive used most often should be assigned a TCID of **0** (zero).

-help
Prints the online help for the command. All other valid options specified with this option are ignored. For complete details about receiving help, see the **dfs_intro(8dfs)** reference page.

Description

Commands in the **bak** command suite are issued by system administrators to work with the DFS Backup System. The commands copy user and system files to backup tapes and restore information from the tapes, if necessary. All **bak** commands are restricted to administrative users only.

The Backup System relies primarily on the following two types of machines and the information and services they provide:

- *Backup Database machines*, which are server machines that house the DFS Backup Database. A cell must have at least one Backup Database machine to use the Backup System; it is recommended that a cell have at least three Backup Database machines. The Backup Database stores two types of records: dump set records, which list the fileset families and tapes in the dump sets; and administrative records, which list fileset families and their entries, as well as dump levels and tape hosts.
- *Tape Coordinator machines*, which are client or server machines with attached tape drives. A Tape Coordinator machine runs an instance of the **butc** process, which is the Backup Tape Coordinator process, for each drive. A Tape Coordinator process controls the behavior of its associated drive and accepts service requests from the Backup System.

  A Tape Coordinator ID (TCID) acts as an identifier for the Tape Coordinator. The TCID for each Tape Coordinator is assigned in the **TapeConfig** file on the machine that houses the tape drive and in the Backup Database. Each TCID is unique to the cell in which the Tape Coordinator is used. With **bak** commands, the TCID specifies the Tape Coordinator to use with the command.

Interactive Mode

The **bak** command suite can be used in regular command mode or in interactive mode. To enter interactive mode, enter **bak** at a command shell prompt. While you are using this mode, the following information applies:

- The word **bak** does not need to be entered with each command; the **bak>** prompt replaces the command shell prompt.
- Regular expression characters do not need to be escaped; in regular command mode, all regular expression characters must be placed in " " (double quotes) or escaped with a \ (backslash).
- New connections do not have to be established to the **bakserver** and **flserver** processes, as necessary, each time a command is issued, so execution time is faster than in noninteractive mode.
- Multiple operations can be tracked with the **bak jobs** command; in regular command mode, pending operations cannot be tracked.
- Currently executing and pending operations can be canceled with the **bak kill** command; in regular command mode, the **bak kill** command cannot be used.

Descriptions of the **bak jobs**, **bak kill**, and **bak quit** interactive commands follow; interactive commands can be issued *only* in interactive mode (at the **bak>** interactive prompt).

The bak jobs Command

The **bak jobs** command lists the job ID number the Backup System has assigned to each dump and restore operation for a Tape Coordinator; the listed operations can be currently executing or pending. The job ID number is not the same as the unique dump ID number assigned to each dump set by the Backup System. (It is also not the same as the task ID number that is sometimes displayed in the output of certain commands; the task ID number can always be safely ignored.)

The complete syntax for the command is

jobs [**-help**]

The **-help** option displays the online help for the command.

If no operations are executing or pending, the **bak>** prompt returns immediately. Otherwise, the output includes one line for each operation, reporting

- The job ID number.
- A name describing the operation.
- The number of kilobytes transferred so far (from file system to tape for a dump operation, from tape to file system for a restore operation).
- For a dump operation, the string **fileset** followed by the name of the fileset currently being dumped; for a restore operation, the string **fileset** followed by the name of the fileset currently being restored.
- A message indicating the status of the operation. No message is displayed if the operation is executing normally.

The bak kill Command

The **bak kill** command terminates a currently running dump, restore, or tape labeling operation. If the command interrupts a backup operation, all filesets written to the tape before the kill signal is received are complete and usable. The fileset being written when the signal is received may not be complete and *should not be used*. It is best not to use any of the filesets from an interrupted dump.

If the command interrupts a restore operation, all completely restored filesets are online and usable. Because complete restoration of a fileset usually requires data from multiple tapes (a full dump tape and one or more incremental dump tapes), most

filesets are usually not completely restored. If the kill signal occurs before the system accesses all of the necessary tapes, most filesets are not restored to the desired date or version and *should not be used*.

If the interrupted restore is overwriting one or more existing filesets, the filesets can be lost entirely; however, the data being restored still exists on tape. In general, to avoid the inconsistencies that can result from an interrupted restore operation, reinitiate the restore operation.

The complete syntax for the command is

kill
-job {*jobID* | *dump_set*}
[**-help**]

The **-job** option identifies the operation to kill. It can be

- The job ID of the operation, as displayed in the output of the **bak jobs** command.
- The name of the operation, as displayed in the output of the **bak jobs** command if the operation is a dump. Dump set names associated with dump operations have the form *fileset_family_name* **.***dump_level*. It is not possible to distinguish restore operations by name.

The **-help** option displays the online help for the command. All other valid options specified with the **-help** option are ignored.

The bak quit Command

The **bak quit** command exits interactive mode; the regular shell prompt replaces the **bak>** prompt.

The complete syntax for the command is

quit [**-help**]

The **-help** option displays the online help for the command.

Command and Monitoring Windows

When using the Backup System, you can use a single terminal session as the command window in which to issue **bak** commands to the Tape Coordinators on all Tape Coordinator machines. In addition, you must open a separate monitoring session for each Tape Coordinator process running on a Tape Coordinator machine. The Tape Coordinator process runs in the foreground; any prompts from the Backup System appear in this window.

Aliases

An alias is an alternate way of entering a command. Each alias is either shorter than the original command or it is unique within the command's suite. (Because only the number of characters sufficient to uniquely identify a command need to be entered to execute the command, unique aliases require less typing.)

The **bak** suite is the only command suite with aliases. The following commands in the **bak** suite can also be entered as specified:

bak restoredb
Can be entered as **bak dbrestore**.

bak restoredisk
Can be entered as **bak dkrestore**.

bak restoreft
Can be entered as **bak ftrestore**.

bak restoreftfamily
Can be entered as **bak familyrestore**.

Cautions

Specific cautionary information is included with individual commands.

Receiving Help

There are several different ways to receive help about DFS commands. The following examples summarize the syntax for the different help options:

$ **man bak**
Displays the reference page for the command suite.

$ **man bak_** *command*
Displays the reference page for an individual command. You must use an _ (underscore) to connect the command suite to the command name. *Do not use the underscore when issuing the command in DFS.*

$ **bak help**
Displays a list of commands in a command suite.

$ **bak help** *command*
Displays the syntax for a single command.

$ **bak apropos -topic** *string*
Displays a short description of any commands that match the specified *string*.

Consult the **dfs_intro(8dfs)** reference page for complete information about the DFS help facilities.

Privilege Required

It is recommended that all system administrators using the Backup System be included in the following lists: the **admin.bak** file on all machines that house the Backup Database, the **admin.fl** file on all machines that house the Fileset Location Database (FLDB), and the **admin.ft** file on all File Server machines. The issuer of a **bak** command must be included in the **admin.bak** list on all machines that house the Backup Database.

Related Information

Commands: **bak adddump(8dfs)**, **bak addftentry(8dfs)**, **bak addftfamily(8dfs)**, **bak addhost(8dfs)**, **bak apropos(8dfs)**, **bak deletedump(8dfs)**, **bak dump(8dfs)**, **bak dumpinfo(8dfs)**, **bak ftinfo(8dfs)**, **bak help(8dfs)**, **bak labeltape(8dfs)**, **bak lsdumps(8dfs)**, **bak lsftfamilies(8dfs)**, **bak lshosts(8dfs)**, **bak readlabel(8dfs)**, **bak restoredb(8dfs)**, **bak restoredisk(8dfs)**, **bak restoreft(8dfs)**, **bak restoreftfamily(8dfs)**, **bak rmdump(8dfs)**, **bak rmftentry(8dfs)**, **bak rmftfamily(8dfs)**, **bak rmhost(8dfs)**, **bak savedb(8dfs)**, **bak scantape(8dfs)**, **bak setexp(8dfs)**, **bak status(8dfs)**, **bak verifydb(8dfs)**, **dfs_intro(8dfs)**.

bak adddump

Purpose

bak adddump – Defines a dump level in the dump hierarchy

Synopsis

bak adddump -level *dump_level...* [**-expires** *date*]… [**-help**]

Options

-level *dump_level*

Names each new dump level to be added to the dump hierarchy. Specify a full pathname for each dump level. Precede the name of each level by a / (slash); the / (slash) is a metacharacter that separates each level in a dump level name. When defining a full dump level, precede the name of the level with a / (slash). When defining an incremental dump level, precede the name of each dump level in the name with a / (slash); the elements in the pathname preceding the last one must already exist in the dump hierarchy. The complete pathname of each dump level must be unique within the Backup Database of the local cell.

Dump level names can have any number of elements. Each element cannot contain more than 28 characters. Complete dump level names cannot contain more than 256 characters. They can include any characters. (To avoid confusion when dump set names are created, the name should not include a dot. When a dump set is transferred to tape, the fileset family name and the last component of the dump level name are joined by a dot to form the name of the dump set.) When including regular expression characters, escape each character with a \ (backslash) or " " (double quotes).

-expires *date*

Defines the expiration date to be associated with each new dump level. Expiration dates can be specified as absolute or relative values. Absolute expiration dates have the format

at*mm*/*dd*
/*yy*

[*hh***:***mm*]

The word **at** is followed by a date (*month*/*day*/*year*) and, optionally, a time (*hours*:*minutes*). When the system creates a dump set at this level, it assigns the specified date as the expiration date of the tape that contains the dump set.

Valid values for *yy* are 00 to 37, which are interpreted as the years 2000–2037, and 70 to 99, which are interpreted as 1970–1999. Values between 38 and 69 cannot be interpreted because the years to which they correspond (2038–2069) exceed the capacity of the standard UNIX representation of dates (the number of seconds since 12:00 a.m. on 1 January 1970). Values between 38 and 69 are reduced to 2038.

If specified, the time must be in 24-hour format (for example, **20:30** for 8:30 p.m.). The default time is **00:00** (12:00 a.m.).

Relative expiration dates have the format

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

The word **in** is followed by a number of years (maximum 9999), months (maximum 11), and days (maximum 30), or a combination of these arguments. When the system creates a dump set at this level, it adds the specified values to the current date to calculate the expiration date of the tape that contains the dump set. At least one of the three values must be specified, and the appropriate unit abbreviation (**y**, **m**, or **d**) must always accompany a value. If more than one of the three is specified, they must appear in the order shown. As with absolute dates, a number of years that causes the relative time to exceed the year 2038 is effectively truncated to the number of years remaining until 2038.

If you omit this option, tapes created at the specified dump levels have no expiration dates, meaning they can be overwritten by appropriately named dump sets at any time. Although the **-expires** option is followed by an ellipsis, you can specify only one expiration date. (The ellipsis is included to accommodate the DFS command parser.)

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

Description

The **bak adddump** command defines one or more dump levels in the dump hierarchy that is stored in the Backup Database and names them as specified by **-level**. Precede each different level in a dump level name with a / (slash) metacharacter. If a dump level is for full dumps, provide only its name preceded by a / (slash) (for example, / **full**).

If a dump level is for incremental dumps, its name resembles a pathname listing the dump levels that serve as its parents, starting with a full dump level and proceeding (in order) down the hierarchy. The dump level's immediate parent (named by the next-to-last element in the pathname) is the reference point that determines which files are included in dump sets made at the dump level. Files with modification time stamps later than the date and time when the volume was dumped at the parent dump level are included.

The optional **-expires** option associates an expiration date with each dump level. The expiration date is applied to tapes containing dump sets made at the dump level; after the specified date, the Backup System overwrites the tape's contents with acceptably named dump sets without question.

An attempt to overwrite an unexpired tape fails until the issuer relabels the tape with the **bak labeltape** command. (Because the label records the unexpired expiration date or unacceptable name, erasing the label removes the obstacle to overwriting.) If no expiration date is defined for a tape, the Backup System overwrites the dump set on the tape with a dump set of the same name without question. Expiration dates can be either absolute or relative; see the Options section for details.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

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

```
$ bak addd -level /yearly -expires in 1y
```

The following command defines an incremental dump called **/full/incr1** with a relative expiration date of 3 months and 15 days:

```
$ bak addd -l /full/incr1 -e in 3m 15d
```

The following command defines two dump levels, **week1** and **week2**; both are incremental from the parent, **monthly**, and both are defined to expire at 12:00 a.m. on 1 January 1992:

```
$ bak adddump -l /monthly/week1 /monthly/week2 -e at 01/01/92
```

Related Information

Commands: **bak dump(8dfs)**, **bak labeltape(8dfs)**, **bak lsdumps(8dfs)**, **bak rmdump(8dfs)**.

bak addftentry

Purpose

bak addftentry – Defines a fileset family entry in a fileset family

Synopsis

bak addftentry -family *fileset_family_name* **-server** *machine* **-aggregate** *name* **-fileset** *name* [**-help**]

Options

-family *fileset_family_name*

Names the fileset family to which this fileset family entry is to be added. The fileset family must already have been created with the **bak addftfamily** command.

-server *machine*

Indicates the File Server machines that house the filesets in the fileset family entry. Legal values for a single machine are the machine's DCE pathname, the machine's host name, or the machine's IP address. You can also specify the regular wildcard expression (**.***) to match all machine names; in noninteractive mode, surround the wildcard with double quotes ("**.***").

-aggregate *name*

Indicates the aggregates that house the filesets in the fileset family entry. Legal values are the device name or aggregate name of an aggregate (these names are specified in the first and second fields of the entry for the aggregate in the *dcelocal*/**var/dfs/dfstab** file) or the regular wildcard expression (**.***), which matches any aggregate name. In noninteractive mode, surround the wildcard with double quotes ("**.***").

-fileset *name*

Indicates the filesets to be included in the fileset family entry. Legal values are a specific fileset name, the regular wildcard expression (**.***), or a regular expression that includes the regular expression characters described in the **Description** section of this reference page. In noninteractive mode, surround the entire argument with " "

(double quotes) if it contains regular expression characters, or escape each regular expression character with a \ (backslash); otherwise, the command shell attempts to interpret the characters.

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

Description

The **bak addftentry** command adds a fileset family entry to the fileset family specified with the **-family** option. The fileset family must already have been created with the **bak addftfamily** command.

A fileset family entry can include different numbers and groupings of filesets, depending on how the **-server**, **-aggregate**, and **-fileset** options are combined. For the **-server** and **-aggregate** options, the issuer can specify either a single, specific value or the wildcard (**.***). The wildcard matches any string, so it matches every server machine name or aggregate name found in the Fileset Location Database (FLDB). The **bak** program initiates a search of the entire FLDB to resolve the wildcards.

For the **-fileset** option, a wider range of notation from the regular expression character set is acceptable and can be combined with specific character strings. Regular expression characters are case sensitive. In addition to strings of individual letters (which match any occurrence of that exact string) and the wildcard (**.***, which matches any fileset name), the acceptable notation includes the following regular expression characters. Note that these characters cannot be used for server machine or aggregate names.

***** (asterisk) Matches any number of repetitions (0 or more) of the previous character and can be combined with any other regular expression character.

[ ](brackets) Around a list of characters, matches a single instance of any of the characters, but no other characters. For example, **[abc]** matches **a** or **b** or **c** but not **d** or **A** or **ab**.

^ (caret) When used as the first character in a bracketed set, indicates a match with any single character except the characters that follow it. For example, **[^a]** matches any single character except lowercase **a**.

? (question mark)
Matches any single character or no character. For example, **?** matches **a** or **A** or **1** or *****.

. (dot) Matches any single character, but a character must be present.

\ (backslash) Can precede any of the regular expression characters in this list so that they match only their literal values. For example, the expression ***** matches a single asterisk, and the expression \\ matches a single backslash.

In the following example, the combination of letters and regular expression characters matches any string that begins with a **user.** prefix and ends with a **.bak** extension:

user\..*\.bak

The previous example is issued in interactive mode. When issuing this command in noninteractive mode, it is necessary to enclose character strings that include regular expression characters in " " (double quotes) or to escape the regular expression characters with the \ (backslash); for example, **"user\..*\.bak"** and **user\\..*\\.bak** are equivalent to the previous example. Otherwise, the command shell attempts to resolve the regular expression characters rather than pass them to the **bak** command interpreter for resolution. This can result in failure of the command or creation of incorrect fileset entries.

Possible values for the arguments of the **bak addftentry** command follow. To create a fileset family entry that includes

- Every fileset in the cell's file system, provide the **.*** wildcard for all three options.
- Every fileset on a machine, provide the DCE pathname of the machine with **-server** and the **.*** wildcard for **-aggregate** and **-fileset**.
- Every fileset on every aggregate of the same name, provide the aggregate name with **-aggregate** and the **.*** wildcard for **-server** and **-fileset**.
- Every fileset in the cell's file system that includes a common string of letters in its name (such as a **.bak** extension), provide the **.*** wildcard for **-server** and **-aggregate** and a character string/regular expression combination for **-fileset**.
- Every fileset on one aggregate, provide the DCE pathname of the machine with **-server**, the aggregate name with **-aggregate**, and the **.*** wildcard for **-fileset**.
- Every fileset on a specific machine that includes a common string of letters in its name (such as a **.bak** extension), provide the DCE pathname of the machine with **-server**, the **.*** wildcard for **-aggregate**, and a character string/regular expression combination for **-fileset**.
- Every fileset on each machine's similarly named aggregate that includes a common string of letters in its name (such as a **.bak** extension), provide the **.*** wildcard

for **-server**, the aggregate name for **-aggregate**, and a character string/regular expression combination for **-fileset**.

- Every fileset on one aggregate that includes a common string of letters in its name (such as a **.bak** extension), provide the DCE pathname of the machine with **-server**, the aggregate name with **-aggregate**, and a character string/regular expression combination for **-fileset**.
- A single fileset, provide the DCE pathname of the machine with **-server**, the aggregate name with **-aggregate**, and the fileset name with **-fileset**.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following commands add a fileset family entry that includes all filesets in the cell that begin with a **user.** prefix to the fileset family called **user**. The two commands, issued in noninteractive mode, are equivalent.

```
$ bak addftentry user ".*" ".*" "user\..*"
```

```
$ bak addftentry user ".*" ".*" user\\..\*
```

Both of the previous commands could be issued in interactive mode as

```
bak>addftentry user .* .* user\..*
```

Related Information

Commands: **bak addftfamily(8dfs)**, **bak lsftfamilies(8dfs)**, **bak rmftentry(8dfs)**.

Files: **dfstab(4dfs)**.

bak addftfamily

Purpose

bak addftfamily – Creates a new (empty) fileset family in the Backup Database

Synopsis

bak addftfamily -family *fileset_family_name* [**-help**]

Options

-family *fileset_family_name*

Names the new fileset family. The fileset family name must be unique within the Backup Database of the local cell. It can be no longer than 31 characters, and it can include any characters. (To avoid confusion when dump set names are created, the name should not include a dot. When a dump set is transferred to tape, the fileset family name and the last component of the dump level name are joined by a dot to form the name of the dump set.)

Regular expression characters used in the name of the fileset family must be escaped with a \ (backslash) to prevent the command shell from expanding them when working in noninteractive mode; for example, **usr*** for a fileset family named **usr***. Because they have no meaning in the name of a fileset family, regular expression characters are not recommended.

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

Description

The **bak addftfamily** command creates a new fileset family in the Backup Database, assigning it the name specified with the **-family** option. To make it easier to track its contents, the fileset family name should give some indication of the filesets it contains (for example, **user** for the fileset family that includes all user filesets in the file system).

Do not include dots in the fileset family name. The names of tapes that contain dump sets of this fileset family consist of the fileset family name and the final component of the dump level name joined by a dot.

After issuing this command, enter the **bak addftentry** command to define the fileset entries included in the fileset family. Use the **bak lsftfamilies** command to list the fileset families currently defined in the Backup Database. Use the **bak rmftfamily** command to remove a currently defined fileset family from the Backup Database.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command creates a fileset family called **sys**:

$bak addftf sys

Related Information

Commands: **bak addftentry(8dfs)**, **bak lsftfamilies(8dfs)**, **bak rmftfamily(8dfs)**.

bak addhost

Purpose

bak addhost – Adds a Tape Coordinator entry to the Backup Database

Synopsis

bak addhost -tapehost *machine* [**-tcid** *tc_number*] [**-help**]

Options

-tapehost *machine*

Names the machine for which the Tape Coordinator is to be added. You can specify the machine's DCE pathname (for example, */.../abc.com/hosts/bak1*), the machine's host name (for example, **bak1.abc.com**), or its IP address (for example, **11.22.33.44**).

-tcid *tc_number*

Specifies the Tape Coordinator ID (TCID) to be assigned to the Tape Coordinator. Legal values are integers from 0 to 1023. A value must match the TCID assigned to the Tape Coordinator in the *dcelocal* **/var/dfs/backup/TapeConfig** file on the **-tapehost** machine, and it must be unique among TCIDs in the Backup Database of the local cell. Each Tape Coordinator must have its own TCID, but the TCIDs need not be assigned in sequence (for example, it is legal to skip numbers or to assign them out of order). If this option is omitted, a value of **0** (zero) is used.

Issuing **bak** commands is most convenient if the Tape Coordinator used most often has a TCID of **0** (zero). The **-tcid** option can then be omitted to direct commands to that Tape Coordinator.

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

Description

The **bak addhost** command creates an entry for a Tape Coordinator in the Backup Database. The entry indicates

- The machine for which the Tape Coordinator is defined (specified by **-tapehost**).
- The Tape Coordinator's TCID (specified by **-tcid**).

Repeat the command once for each Tape Coordinator on a Tape Coordinator machine. The Backup Database allows a maximum of 1024 Tape Coordinators in the local cell.

The mapping between the TCID of a Tape Coordinator and the device name of the drive with which it is associated is recorded in the **TapeConfig** file on the Tape Coordinator machine. The **TapeConfig** file must be altered accordingly when this command is issued.

Enter the **bak lshosts** command to list the Tape Coordinators that have entries in the Backup Database. Enter the **bak rmhost** command to remove the entry for a Tape Coordinator from the Backup Database.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command creates an entry in the Backup Database for a Tape Coordinator on the machine named **bak1**. The Tape Coordinator is assigned a TCID of **0** (zero); the mapping between the TCID of the Tape Coordinator and the device name of a tape drive must appear in the **TapeConfig** file.

```
$
bak addhost /.../abc.com/hosts/bak1
```

The following command creates an entry in the Backup Database for a Tape Coordinator on the machine named **bak2**; the Tape Coordinator has a TCID of **1**.

```
$
bak addh /.../abc.com/hosts/bak2 1
```

Related Information

Commands: **bak lshosts(8dfs)**, **bak rmhost(8dfs)**.

Files: **TapeConfig(4dfs)**.

bak apropos

Purpose

bak apropos – Shows each help entry containing a specified string

Synopsis

bak apropos -topic *string* [**-help**]

Options

-topic *string* Specifies the keyword string for which to search. If it is more than a single word, surround the string with " " (double quotes) or other delimiters. Type all strings for **bak** commands in lowercase letters.

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

Description

The **bak apropos** command displays the first line of the help entry for any **bak** command containing the string specified by **-topic** in its name or short description.

To display the syntax for a command, use the **bak help** command.

Privilege Required

No privileges are required.

Output

The first line of the online help entry for a command lists the command and briefly describes its function. This command shows the first line for any **bak** command where the string specified by **-topic** is part of the command name or first line.

Examples

The following command lists all **bak** commands containing the word **tape** in their names or short descriptions:

```
$ bak ap tape
```

```
labeltape: label tape
readlabel: read label on tape
scantape: list filesets on tape
status: get tape coordinator status
```

Related Information

Commands: **bak help(8dfs)**.

bak deletedump

Purpose

bak deletedump – Deletes the record of a dump set from the Backup Database

Synopsis

bak deletedump -id *dumpID* [**-help**]

Options

-id *dumpID* The dump ID number of the dump set to be deleted from the Backup Database. Use the **bak dumpinfo** command to list the current dump IDs from the Backup Database.

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

Description

The **bak deletedump** command removes the record of the dump set associated with the specified dump ID from the Backup Database. It can be used to remove from the Backup Database the record of a dump that contains incorrect data or for which the corresponding tape is to be discarded.

After the record of a dump set is deleted from the Backup Database, dump sets for which it serves as the parent, either directly or indirectly, can no longer be used to restore data to the file system. The **bak deletedump** command can be reissued to remove the record of such dumps from the Backup Database, but leaving a record of them in the database causes no problems. Also, as long as the tape that contains the parent dump set remains available, the **bak scantape** command can be used to restore information about that dump set from the tape to the Backup Database, again making the dump sets that rely on the parent dump set usable.

Use the **bak dumpinfo** command to list the dump IDs currently recorded in the Backup Database.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command deletes the record of the dump set with dump ID **653777462** from the Backup Database:

```
$
bak del
653777462
```

Related Information

Commands: **bak dump(8dfs)**, **bak dumpinfo(8dfs)**, **bak scantape(8dfs)**.

bak dump

Purpose

bak dump – Dumps a specific fileset family at a specific dump level

Synopsis

bak dump - family *fileset_family_name* **- level** *dump_level* [**- tcid** *tc_number*] [**-noaction**][**-help**]

Options

-family *fileset_family_name*
Names the fileset family (already defined in the Backup Database using the **bak addftfamily** and **bak addftentry** commands) to be dumped.

-level *dump_level*
Indicates the dump level (already defined in the Backup Database using the **bak adddump** command) to be used in dumping the fileset family. Provide a full pathname for the dump level, including all necessary / (slashes). This option determines whether the dump is full or incremental and, in the latter case, determines which dump level serves as the parent for the dump.

-tcid *tc_number*
Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive containing the tape. If omitted, it defaults to **0** (zero).

-noaction Displays all filesets that would be included in the indicated dump without actually performing the dump. This lets you check a fileset family's size before actually dumping it so that you can calculate the correct number of tapes needed. Specify all other options as you would to actually perform the operation.

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

Description

The **bak dump** command dumps the fileset family specified by **-family** at the dump level specified as a pathname by **- level**. There are two types of dumps:

- A **full dump** records the structure of all directories in each fileset in the fileset family and includes all the data in each fileset.
- An **incremental dump** also records the structure of all directories in each fileset in the fileset family, but it includes data from only those files in the filesets that changed since the fileset family was dumped at the parent dump level; such files have modification time stamps later than the date and time at which the fileset family was dumped at the parent dump level. The program uses the next-to-last element in the **- level** pathname as the parent dump and consults the Backup Database to learn the date and time at which this fileset family was last dumped at that level.

  If the program cannot locate a dump set dumped at a parent dump level, it looks recursively in the Backup Database for a dump set created at the dump level one higher in the pathname. If it can find no dump set created at a higher dump level in the hierarchy, it creates a full dump set.

If the Backup System is unable to access a fileset (for example, because of a File Server machine or Fileset Server outage), it attempts to access the fileset three times over the course of the operation. If it cannot access the fileset after the third attempt, it omits the fileset from the dump instead of stopping the dump entirely. If the Tape Coordinator performing the dump was initialized at debug level 1, a report on the failure to include the fileset appears in the Tape Coordinator's monitoring window. The Tape Coordinator's error file also records the fileset's omission.

If the failure to access a fileset occurs during a full dump, the next incremental dump of the fileset includes the entire fileset. If the failure occurs during an incremental dump, the next incremental dump of the fileset includes all files modified since the fileset was last included in a dump set.

Before writing the dump to tape, the Tape Coordinator checks that the tape in the indicated tape drive has an acceptable name on its label. If the name on the label is not acceptable, the Backup System prompts for the correct tape. There are three acceptable types of names:

- The tape is labeled *fileset_family_name.dump_level.index* , where *fileset_family_name* and *dump_level* match the values provided on the command line (with **-family** and **-level**). The *dump_level* is the last component of the specified dump level; the *index* distinguishes this tape from others that

contain this same dump set. If a single tape contains the entire dump set, its index is 1.

- The tape is labeled as empty. The Backup System labels the tape with the correct name of the form *fileset_family_name.dump_level.index* .
- The tape is not labeled because it has never been used in the Backup System. The Backup System labels the tape with the correct name of the form *fileset_family_name.dump_level.index*.

If it finds that the name on the tape label is acceptable, the Backup System checks the expiration date on the tape before it writes data to it. If the expiration date has not expired, the system does not write data to the tape unless the issuer relabels the tape with the **bak labeltape** command (because the label records the expiration date, erasing the label removes the obstacle to overwriting). If the expiration date has expired or if no expiration date is associated with the tape, the system overwrites the contents of the tape without question (given that the tape has an acceptable name).

The tape label also tells the Tape Coordinator the size of the tape. However, the Tape Coordinator applies the capacity specified in the **TapeConfig** file for the tape drive containing the tape to any tape, regardless of the size specified in the tape's label. Make sure the tapes are at least as large as the tape size listed in the **TapeConfig** file. If a tape is larger, some of its capacity simply may not be used for the dump; if it is smaller, the dump may fail, but only after the Backup System fills the tape and determines that the tape is too small for the drive.

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

The **-noaction** option instructs the program to display a list of the filesets to be included in a dump set without actually performing the dump. This allows the issuer to determine how large the filesets are before actually dumping them; the issuer can then better calculate the required number of tapes. The command ignores a value specified with the **-tcid** option if the **-noaction** option is used with the command.

The **bak restoreft**, **bak restoredisk**, and **bak restoreftfamily** commands can be used to restore data dumped with the **bak dump** command. You can use the commands to restore data to any type of file system (DCE LFS or non-LFS), regardless of the type of file system from which it was dumped. Thus, you can dump and restore data between DCE LFS and non-LFS file systems, and between different types of non-LFS file systems. (See the documentation for the **bak restoreft**, **bak restoredisk**, and

bak restoreftfamily commands for more information about restoring data; see your vendor's documentation to verify the level of support for dump and restore operations between different types of file systems.)

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines and in the **admin.ft** files on all File Server machines from which filesets are to be dumped.

Output

The following header is displayed in the command window followed by a list of the filesets, identified by name and fileset ID number, to be included in the dump set:

```
Preparing to dump the following filesets: list of filesets
```

The following message indicates that the Backup System has passed the dump request to the indicated Tape Coordinator:

```
Starting dump.
```

It is followed by a message that reports the unique dump ID number associated with this dump operation:

```
Dump ID of dump fileset_family_name.dump_level: dump_ID_number
```

The dump ID also appears in the Tape Coordinator monitoring window if the **butc** command is issued with debug level 1. The dump ID is not the same as the job ID number visible with **(bak) jobs** when **bak dump** is issued in interactive mode.

If the issuer includes the **-noaction** option, the output is

```
Starting dump of fileset family
     'fileset family' (dump level 'dump level')
Total number of filesets : number
Would have dumped the following filesets:
     list of filesets
```

Examples

The following command dumps the filesets in the fileset family **user** according to the dump level **/full/week2/monday**. The issuer places the necessary tapes in the drive with TCID 5.

```
$ bak dump user /full/week2/monday 5
```

```
Preparing to dump the following filesets:
user.jones.bak 387623900
user.pat.bak 486219245
user.smith.bak 597315841
        .
        .
Starting dump.
Dump ID of dump user.monday: 34
```

The following command displays the list of filesets to be dumped when the **sys.rs_aix32** fileset family is dumped at the **/full** dump level:

```
$ bak dump sys.rs_aix32 /full -n
```

```
Starting dump of fileset family 'sys.rs_aix32' (dump level '/full')
Total number of filesets : 24
Would have dumped the following filesets:
     rs_aix32 124857238
     rs_aix32.bin 124857241
     rs_aix32.etc 124
```

857246

. .
. .

Related Information

Commands: **bak adddump(8dfs)**, **bak addftentry(8dfs)**, **bak addftfamily(8dfs)**, **bak deletedump(8dfs)**, **bak dumpinfo(8dfs)**, **bak ftinfo(8dfs)**, **bak labeltape(8dfs)**, **bak lsdumps(8dfs)**, **bak readlabel(8dfs)**, **bak restoredisk(8dfs)**, **bak restoreft(8dfs)**, **bak rmdump(8dfs)**, **bak rmftfamily(8dfs)**. **bak restoreftfamily(8dfs)**,

bak dumpinfo

Purpose

bak dumpinfo – Lists information about specified backups

Synopsis

bak dumpinfo [{**-ndumps** *number* | **-id** *dumpID*}] [**-verbose**][**-help**]

Options

-ndumps *number*
: Specifies the number of dumps about which information is to be displayed; information about the most recent number of dumps specified with this option is displayed. If fewer than the specified number of dumps exist, information about all existing dumps is displayed. Use this option or use **-id** ; omit both options to list information about the last 10 dumps.

-id *dumpID*
: Specifies the unique dump ID number of a specific dump operation about which information is to be displayed. Use this option or use **-ndumps**; omit both options to list information about the last 10 dumps.

-verbose
: Includes a detailed list of information about the dump specified with the **-id** option. This option can be used only with **-id**.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bak dumpinfo** command lists information about specified dump sets. If a number is specified with **-ndumps**, information about that number of dump sets is displayed (information about the most recent **-ndumps** is displayed); if a specific dump ID number is specified with **-id**, information about that dump set is displayed; if both options are omitted, information about the 10 most recent dump sets is displayed.

The command displays information from the Backup Database. It can be used to display dump IDs prior to using the **bak deletedump** command to delete the record of

one or more dump sets from the Backup Database. To view more detailed information about a specific dump set, specify both the **-id** option and the **-verbose** option.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Output

The following information is displayed for each dump listed:

DumpID The dump set's ID number. This is a unique identifier that the Backup System uses internally.

parentID The dump ID of the dump set that served as the parent for this dump set. A value of **0** (zero) means this is a full dump set and so has no parent, in which case **lvl** is also **0** (zero).

lvl The location in the dump hierarchy of the dump level used in creating the dump set. A value of **0** (zero) indicates a full dump set. A value of **1** or greater indicates an incremental dump set made at the indicated level in the hierarchy.

created The date and time at which the Backup System started executing the dump operation that created this dump set.

nt The number of tapes required to record the dump set.

nfsets The number of filesets included in the dump set.

dump_name
The name of the dump set.

Additional information is displayed if both the **-id** and **-verbose** options are specified.

Examples

The following example displays information about the last three dumps:

```
$ bak dumpinfo -ndumps 3
```

```
    DumpID   parentID lvl     created     nt nfsets dump_name
-- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---
729293644  729289323   1  02/09/93  5:34  1     43 users.tue
729287531  729286818   1  02/08/93  4:52  1     23 users.mon
729286056          0   0  02/07/93  4:27  1     31 users.wk1
```

Related Information

Commands: **bak deletedump(8dfs)**, **bak dump(8dfs)**, **bak ftinfo(8dfs)**, **bak lsdumps(8dfs)**.

bak ftinfo

Purpose

bak ftinfo – Displays a fileset's dump history from the Backup Database

Synopsis

bak ftinfo -fileset *name* [**-help**]

OPTIONS

-fileset *name*
: Names the fileset whose dump history is to be displayed. Include a **.backup** extension if the backup version of the fileset (rather than the read/write version) was dumped.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

DESCRIPTION

The **bak ftinfo** command displays a dump history for the specified fileset, detailing the dates on which the fileset was cloned (the backup version was made) and dumped and the tapes on which it resides. If the dump was made of the backup version, as is usual, then **-fileset** must include the **.backup** extension.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

OUTPUT

The output lists information about the dump sets in which **-fileset** is included, with the most recent dump set listed first. The output is displayed in six columns, as follows:

DumpID
: The dump set's ID number. This is a unique identifier that the Backup System uses internally. It allows the issuer to check that the parent ID

for an incremental dump set matches the dump ID of the dump set created previously.

parentID The dump ID of the dump set that served as the parent for this dump set. A value of **0** (zero) means this is a full dump set and so has no parent, in which case **lvl** is also **0** (zero). It normally corresponds to the dump ID of the dump set created previously (the one on the next line of the output).

lvl The location in the dump hierarchy of the dump level used in creating the dump set. A value of **0** (zero) indicates a full dump set. A value of **1** or greater indicates an incremental dump set made at the indicated level in the hierarchy.

creation date
The date and time at which the Backup System started executing the dump operation that created the dump set.

clone date The date and time at which the fileset was created. For a backup or read-only fileset, this represents the time at which it was cloned from its read/write source. For a read/write fileset, it indicates when the Backup System accessed the fileset to include it in the present dump set.

tape name The name of the tape that contains the dump set.

EXAMPLES

The following command displays dump information about the fileset named *user.smith.backup*:

```
$ bak ftinfo user.smith.backup


  DumpID   parentID  lvl creation  date   clone    date    tape name
654972910  654946323  1  10/01/91  5:07  10/01/91  4:01  users.tuesday.1
654960415  654946323  1  09/30/91  5:11  09/30/91  4:16  users.monday.1
654946323          0  0  09/29/91  5:36  09/28/91  4:31  users.week.1
```

RELATED INFORMATION

Commands: **bak dump(8dfs)**, **bak dumpinfo(8dfs)**, **bak lsdumps(8dfs)**.

bak help

Purpose

bak help – Shows syntax of specified **bak** commands or lists functional descriptions of all **bak** commands

Synopsis

bak help [**-topic** *string*]… [**-help**]

Options

-topic *string* Specifies each command whose syntax is to be displayed. Provide only the second part of the command name (for example, **dump**, not **bak dump**). If this option is omitted, the output provides a short description of all **bak** commands.

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

Description

The **bak help** command displays the first line (name and short description) of the online help entry for every **bak** command if **-topic** is not provided. For each command name specified with **-topic**, the output lists the entire help entry.

Use the **bak apropos** command to show each help entry containing a specified string.

Privilege Required

No privileges are required.

Output

The online help entry for each **bak** command consists of the following two lines:

- The first line names the command and briefly describes its function.

- The second line, which begins with **Usage:**, lists the command options in the prescribed order.

Examples

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

```
$ bak help dump
```

```
bak dump: start dump
Usage: bak dump -family <fileset_family_name> -level <dump_level>
[-tcid <tc_number>] [-noaction] [-help]
```

Related Information

Commands: **bak apropos(8dfs)**.

bak labeltape

Purpose

bak labeltape – Creates the label on a tape

Synopsis

bak labeltape [**-tape** *tape_name*] [**-size** *tape_size*] [**-tcid** *tc_number*] [**-help**]

Options

-tape *tape_name*

Specifies the name to assign to the tape. If this option is omitted, the tape is marked as empty with a null identifier.

An assigned name must reflect the dump set that is to go on the tape. It must be of the form *fileset_family_name.dump_level.index*, where *fileset_family_name* and *dump_level* constitute the name of the dump set to go on the tape. The *dump_level* is the last component of the name of the appropriate dump level; the *index* is an integer that represents the tape's place in the collection of tapes needed to contain the entire dump set. If the dump set fits on one tape, the index is 1.

-size *tape_size*

Indicates the tape capacity. Providing this option is necessary only for information purposes. The Tape Coordinator uses the capacity specified in the **TapeConfig** file for any tape used in its tape drive. If this option is omitted, the size specified in the **TapeConfig** file for the drive is used for the tape's label.

The default unit of size is kilobytes. It is also possible to express this number in megabyte or gigabyte units. To indicate megabyte units, add an uppercase or lowercase "m" with the number (with no space between the number and letter). To indicate gigabyte units, add an uppercase or lowercase "g" with the number (with no space between the number and letter).

-tcid *tc_number*

Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive containing the tape. If omitted, it defaults to **0** (zero).

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

Description

The **bak labeltape** command creates a label, readable by the Backup System, at the beginning of a tape. The issuer can either assign a name with the **-tape** option or omit the option to label the tape as empty.

The **-size** option is useful mainly for information purposes. The Tape Coordinator uses the capacity specified in the **TapeConfig** file for any tape used in its drive. It also copies the size specified in the **TapeConfig** file to the label of any tape that has no size specified in its label.

Labeling a tape is not a prerequisite to putting a dump set on it. The **bak dump** command accepts partially labeled or completely unlabeled tapes. However, the **bak labeltape** command can be used to overwrite an existing label. This is useful if the data on a tape is no longer needed, but the tape's label prevents the tape from being used (because the label bears an inappropriate name or contains an unexpired expiration date). Overwriting the label with this command removes the obstacle to the tape's reuse.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command puts the label **user.monthly.1** on the tape in the drive whose TCID is **3**:

```
$ bak la user.monthly.1 -tcid 3
```

The following three commands are equivalent in effect. They all mark the tape in the drive whose TCID is **4** with a capacity of 2 gigabytes and the default name null.

```
$
bak label -size 2g -tcid 4
```

```
$
bak label -size 2048M -tcid 4

$
bak label -size 2097152
-tcid 4
```

Related Information

Commands: **bak readlabel(8dfs)**.

bak lsdumps

Purpose

bak lsdumps – Creates the label on a tape

Synopsis

bak lsdumps [**-help**]

Options

-help Prints the online help for this command.

Description

The **bak lsdumps** command displays the dump hierarchy from the Backup Database. A dump hierarchy can contain more than one full dump level, each of which defines a separate subhierarchy of dump levels. The **bak lsdumps** command displays the multiple subhierarchies if the Backup Database contains more than one full dump level.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Output

The output depicts the parent/child relationships between full and incremental dump levels in the dump hierarchy. The names of full dump levels are displayed at the far left margin. There can be more than one full dump in the hierarchy; each defines a subhierarchy of dump levels, each of which would presumably be used for dumping different fileset families.

Incremental dump levels are displayed below and indented to the right from their parent dump level, which can be either full or incremental. Incremental dump levels need not be directly below their parent; the amount of indentation alone indicates the parent/child relationship.

Examples

The following example displays a dump hierarchy with two subhierarchies. One subhierarchy starts with the full dump level **/month**, the other with the full dump level **/monday** (their positions at the left margin indicate they are full dump levels).

$ bak lsdumps

```
/month
      /week1
            /tuesday
                    /thursday
      /week2
            /tuesday
                    /thursday
/monday
       /tuesday
               /wednesday
                        /thursday
                                /friday
       /saturday
```

In the first subhierarchy, **/month** serves as the parent for the **/month/week1** and **/month/week2** dump levels, as indicated by the indentation (**/month/week2** is an example of how an incremental level need not be directly below its parent). The **/month/week1** dump level serves as the parent for the **/month/week1/tuesday** dump level, which serves as the parent for the **/month/week1/tuesday/thursday** level. These within-week relationships are repeated under **/month/week2**.

Dump sets created at the **/month** level are full dumps. Dumps performed at the **/month/week1** level include all files modified since the fileset family was dumped at the **/month** level. Dumps performed at the **/month/week1/tuesday** level include all files modified since the fileset family was dumped at the **/month/week1** level, and dumps done at the **/month/week1/tuesday/thursday** level include all files modified since the dump done at the **/month/week1/tuesday** level.

Dumps done at the **/month/week2** level would include all files modified since the fileset family was dumped at the **/month** level. Thus, dumps done at the**/month/week2** level serve as a summary of dumps done since the dump at the **/month/week1**

level (they contain all files modified since a full dump was performed at the **/month** level).

The second subhierarchy, starting with **/monday**, is similarly constructed. The **/monday** dump level represents a full dump (it is at the far left margin); it is the parent for the **/monday/tuesday** level. The **/monday/tuesday** level is the parent for **/monday/tuesday/wednesday**, and so on. The **/monday/saturday** level's parent is **/monday**, so dumps performed at that level represent a summary of all the dumps performed at the intervening levels.

Related Information

Commands: **bak adddump(8dfs)**, **bak dump(8dfs)**, **bak dumpinfo(8dfs)**, **bak ftinfo(8dfs)**, **bak rmdump(8dfs)**.

bak lsftfamilies

Purpose

bak lsftfamilies – Lists fileset families defined in the Backup Database

Synopsis

bak lsftfamilies [**-family** *fileset_family_name*] [**-help**]

Options

-family *fileset_family_name*
Names the fileset family to be displayed with the command. If omitted, all fileset families are displayed.

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

Description

The **bak lsftfamilies** command displays fileset family entry information about the specified fileset family. If **-family** is omitted, it lists all of the fileset families defined in the Backup Database. If **-family** is provided, it lists only that fileset family. In both cases, the fileset family entries in each fileset family are displayed.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Output

The output includes a separate entry for each fileset family. The entry lists the fileset family entries that make up the fileset family. Each fileset family entry is assigned an index number; the issuer of the **bak rmftentry** command uses these index numbers to identify the fileset family entries to delete.

Examples

The following command shows the fileset family entries in the two fileset families currently defined in the Backup Database:

$ bak lsftfamilies

```
Fileset family user:
    Entry 1: server .*, aggregate .*, filesets: user.*\.bak
Fileset family aix31:
    Entry 1: server .*, aggregate .*, filesets: aix31
```

Related Information

Commands: **bak addftentry(8dfs)**, **bak addftfamily(8dfs)**, **bak rmftentry(8dfs)**, **bak rmftfamily(8dfs)**.

bak lshosts

Purpose

bak lshosts – Lists Tape Coordinator entries in the Backup Database

Synopsis

bak lshosts [**-help**]

OPTIONS

-help Prints the online help for this command.

DESCRIPTION

The **bak lshosts** command lists the Tape Coordinator entries currently defined in the Backup Database. The list includes the Tape Coordinators defined for all Tape Coordinator machines in the cell. Each Tape Coordinator is defined in the Backup Database and is, by implication, available for use. However, a Tape Coordinator process does not have to be running on a Tape Coordinator machine at the time this command is issued for its entry to be displayed.

Enter the **bak addhost** command to add an entry for a Tape Coordinator to the Backup Database. Enter the **bak rmhost** command to remove an entry for a Tape Coordinator from the Backup Database.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

OUTPUT

The command first displays a **Tape hosts:** header. It then reports the following information for each Tape Coordinator:

- The name of the machine on which the Tape Coordinator is defined. (The format of the machine name depends on the form specified by the issuer of the **bak addhost** command.)

- The TCID of the Tape Coordinator. Valid TCIDs for Tape Coordinators are integers from 0 to 1023.

EXAMPLES

The following command displays the Tape Coordinators currently defined in the Backup Database:

```
$ bak lshosts


Tape hosts:
Host /.../abc.com/hosts/bak1, TCID 0
Host /.../abc.com/hosts/bak1, TCID 1
Host /.../abc.com/hosts/bak2, TCID 2
Host /.../abc.com/hosts/bak3, TCID 8
Host /.../abc.com/hosts/bak3, TCID 6
Host /.../abc.com/hosts/bak3, TCID 7
```

RELATED INFORMATION

Commands: **bak addhost(8dfs)**, **bak rmhost(8dfs)**.

bak readlabel

Purpose

bak readlabel – Displays the name and size from a tape's label

Synopsis

bak readlabel [**-tcid** *tc_number*] [**-help**]

Options

-tcid *tc_number*
: Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive containing the tape. If omitted, it defaults to **0** (zero).

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bak readlabel** command displays the name and size from the label of the tape in the indicated tape drive. These values are placed on the tape with either the **bak dump** command or the **bak labeltape** command.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Output

For tapes with complete labels, a message appears listing the name and size of the tape. The tape name is of the form *fileset_family_name.dump_level.index*. If a tape has no name, the output reads **NULL**.

The tape size is expressed as follows: If an uppercase or lowercase "g" follows the size, it is a number of gigabytes; if an uppercase or lowercase "m" follows the size, it is a number of megabytes; if a lowercase "k" or the string **Kbytes** follows the size, it is a number of kilobytes.

If the tape is completely unlabeled or if the drive is empty, the output reads **Failed to read tape label**.

Examples

The following command shows the output for the tape with the label **sys.Monthly.3**. The capacity is 2 megabytes (expressed in kilobyte units). The tape is currently in the drive with a TCID of **6**.

```
$ bak read 6
```

```
Tape read was labelled : sys.Monthly.3 size : 2097152 Kbytes
```

The following command shows that the unlabeled tape in the drive with a TCID of **0** (zero) has a capacity of 5 gigabytes:

```
$ bak read
Tape read was labelled : NULL size : 5G
```

Related Information

Commands: **bak dump(8dfs)**, **bak labeltape(8dfs)**.

bak restoredb

Purpose

bak restoredb – Restores a backup copy of the Backup Database

Synopsis

bak restoredb [**-tcid** *tc_number*] [**-help**]

Alias

bak dbrestore

Options

-tcid *tc_number*
: Specifies the TCID of the Tape Coordinator for the tape drive from which the Backup Database is to be restored. If omitted, it defaults to **0** (zero).

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bak restoredb** command restores a backup copy of the entire Backup Database. If the Backup Database becomes damaged, you should delete the old database; then use this command to restore an entirely new version from its backup tape (which must be named **bak_db_dump.1**). The Backup Database is damaged if the disk housing the database becomes damaged or the **bak verifydb** command fails.

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

After the database is removed, restart all **bakserver** processes on the machines on which they were running. Use the **bak addhost** command to add a tape host for the

Tape Coordinator from which you plan to restore the Backup Database. Then use the **bak restoredb** command to restore the new version of the database; the **-tcid** option specifies the TCID of the Tape Coordinator from which to restore the Backup Database (the Tape Coordinator just added with the **bak addhost** command).

Use the **bak savedb** command to save the Backup Database to tape.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Related Information

Commands: **bak savedb(8dfs)**, **bak verifydb(8dfs)**.

bak restoredisk

Purpose

bak restoredisk – Restores the contents of an entire aggregate from tape

Synopsis

bak restoredisk -server *machine* **-aggregate** *name* [**-tcid** *tc_number*] [**-newserver** *machine*] [**-newaggregate** *name*] [**-noaction**][**-help**]

Alias

bak dkrestore

Options

-server *machine*

Names the File Server machine that houses the aggregate you want to restore. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate *name*

Specifies the device name or aggregate name of the aggregate on the machine indicated with the **-server** option that you want to restore. These names are specified in the first and second fields of the entry for the aggregate in the *dcelocal*/**var/dfs/dfstab** file.

-tcid *tc_number*

Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive in which you are placing the necessary tapes. If omitted, it defaults to **0** (zero).

-newserver *machine*

Names the File Server machine to which to restore the data. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. Use this option only if the destination server is different from the server specified with the **-server** option.

-newaggregate *name*
Specifies the device name or aggregate name of the aggregate to which to restore the data. These names are specified in the first and second fields of the entry for the aggregate in the **dfstab** file. Use this option only if the name of the destination aggregate is different from the name of the aggregate specified with the **-aggregate** option.

-noaction Directs the command to display the list of tapes necessary to perform the indicated restore without actually performing the operation.

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

Description

The **bak restoredisk** command restores the contents of the aggregate specified with the **-server** and **-aggregate** options to the file system. To do this, the **bak** program contacts the Fileset Location Server (FL Server) for a listing from the Fileset Location Database (FLDB) of all the filesets that reside on the specified aggregate. It then consults the Backup Database to learn which tapes contain the full and incremental dumps needed to restore every fileset from the aggregate. This command is useful if a disk or machine failure destroys the data on an entire aggregate.

To restore filesets from the specified aggregate to the same site (the site specified with the **-server** and **-aggregate** options), omit the **-newserver** and **-newaggregate** options. The data in the restored filesets overwrites the filesets' current contents; there is no change in the Fileset Location Database (FLDB) entries for the filesets.

To restore the filesets to an alternate site, include the **-newserver** option, the **-newaggregate** option, or both. The filesets continue to use their existing FLDB entries and fileset ID numbers, and the filesets' FLDB entries are updated to record the new site. The current contents of each fileset are replaced with the data restored from tape. The command allows you to restore filesets to a new site as follows:

- To restore the filesets to a different aggregate on the same File Server machine, specify the new aggregate with the **-newaggregate** option.
- To restore the filesets to an aggregate of the same name on a different File Server machine, specify the new File Server machine with the **-newserver** option.
- To restore the filesets to a completely different site, specify the new File Server machine with the **-newserver** option and the new aggregate with the **-newaggregate** option.

If you specify a new site and the filesets to be restored currently exist at their old site, you must use the **fts zap** command to delete the existing filesets before issuing the **bak restoredisk** command. The **bak restoredisk** command fails if you do not use the **fts zap** command to delete the existing filesets before using the **bak restoredisk** command to restore the filesets to the new site.

Note: Do not use the **fts delete** command to delete the existing filesets and their FLDB entries before issuing the **bak restoredisk** command. If you use the **fts delete** command instead of the **fts zap** command, you cannot use the **bak restoredisk** command to restore the filesets; you can restore the filesets only with the **bak restoreft** command.

The **-noaction** option instructs the command to produce a list of the tapes the Backup System would need to perform the indicated restore without actually performing the operation. To do so, include the **-noaction** option with all of the other options to be used with the actual command.

Data can be dumped and restored between different types of file systems. For example, data dumped from a DCE LFS fileset can be restored to a DCE LFS fileset or to any type of nonLFS fileset; likewise, data dumped from a nonLFS fileset can be restored to a DCE LFS fileset or to a different type of nonLFS fileset. (See your vendor's documentation to verify the level of support for dump and restore operations between different types of file systems.)

Restored data is translated into the appropriate format for the file system to which it is restored. Note that incompatible information may be lost when a fileset is dumped and restored between different types of file systems. For example, ACLs on objects in a DCE LFS fileset may be lost if the fileset is restored to a file system that does not support ACLs.

Use the **bak restoreft** command to restore one or more filesets to a single site. Use the **bak restoreftfamily** command to restore a fileset family or to restore one or more filesets to the same site or to different sites.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines and in the **admin.ft** file on the File Server machine to which filesets are to be restored.

Output

If you do not include the **-noaction** option, the **bak restoredisk** command returns the unique dump ID number associated with the restore operation. The dump ID is displayed in the command window following the command line and in the Tape Coordinator's monitoring window if the **butc** command is issued with debug level 1. The dump ID is not the same as the job ID number visible with the **(bak) jobs** command if the **bak restoredisk** command is issued in interactive mode.

If you include the **-noaction** option, a **Tapes needed:** header is displayed, followed by a list of the tapes necessary to complete the restore operation. No dump ID number is reported because none is assigned.

Examples

The following command restores the filesets listed in the FLDB as residing on the aggregate named **/dev/lv01** on the File Server machine named **fs5**. The filesets are restored to the same aggregate and server machine. Tapes are placed in the drive with a TCID of **3**.

```
$ bak restored /.../abc.com/hosts/fs5 /dev/lv01 3
```

```
Starting restore
bak: dump ID of restore operation: 253
bak: Finished doing restore
```

The following command restores the filesets listed in the FLDB as stored on the aggregate named **/dev/lv02** on the File Server machine named **fs1**. The filesets are restored to a new site, the aggregate **/dev/lv01** on the File Server machine **fs3**. The **fts zap** command is used to delete existing filesets from the current site before the **bak restoredisk** command is issued. Tapes are placed in the drive with a TCID of **0** (zero).

```
$ bak restored /.../abc.com/hosts/fs1 /dev/lv02 -news /.../abc.com/hosts/fs3 \
-newa /dev/lv01
```

```
Starting restore
bak: dump ID of restore operation: 256
bak: Finished doing restore
```

Related Information

Commands: **bak dump(8dfs)**, **bak restoreft(8dfs)**, **bak restoreftfamily(8dfs)**, **fts delete(8dfs)**, **fts zap(8dfs)**.

Files: **dfstab(4dfs)**.

bak restoreft

Purpose

bak restoreft – Restores filesets from tape

Synopsis

bak restoreft -server *machine* **-aggregate** *name* **-fileset** *name…* [**-extension** *name_extension*] [**-date** *date*] [**-tcid** *tc_number*] [**-noaction**][**-help**]

Alias

bak ftrestore

Options

-server *machine*

Names the File Server machine to which to restore each specified fileset. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. If the fileset currently exists at a site other than the one specified with this option and the **-aggregate** option, you must delete the existing fileset before restoring it to the specified site.

-aggregate *name*

Specifies the device name or aggregate name of the aggregate to which to restore each specified fileset. These names are specified in the first and second fields of the entry for the aggregate in the *dcelocal* **/var/dfs/dfstab** file. If the fileset currently exists at a site other than the one specified with this option and the **-server** option, you must delete the existing fileset before restoring it to the specified site.

-fileset *name*

Names each fileset to be restored. Provide the name of the read/write version of each fileset, even if (because of its fileset entry definition in a fileset family) the backup version of a fileset was actually dumped. The command automatically appends a **.backup** extension to the name

of a fileset if it can find no record in the Backup Database of a backup performed for the fileset's read/write version.

-extension *name_extension*
Specifies an extension to add to the restored fileset's name to distinguish it from a fileset of the same name that currently exists in the file system. This causes the Backup System to restore the data from tape into a new fileset independent of the existing one. Any string other than **.readonly** or **.backup** is acceptable; if a period is to precede the extension, include it in the string provided.

-date *date* Specifies the date prior to which a dump must have been made to be included in the restore. The **-date** option indicates a date-specific restore; only dump sets dated before the specified date are restored. If omitted, this option defaults to **0** (zero) and a full restore of the most recently dumped version of the fileset occurs. Otherwise, there are two types of legal values:

mm/*dd*/*yy* Specifies 00:00 (12:00 a.m.) on the indicated date. A value of this type causes a date-specific restore containing only data from dumps done before the indicated date (for example, **11**/**22**/**91**).

mm/*dd*/*yy hh*:*mm*
Specifies a time on the indicated date. A value of this type causes a date-specific restore containing only data from dumps done before the indicated date and time. The time must be in 24-hour format (for example, **20:30** is 8:30 p.m.). Surround the entire argument with **" "** (double quotes) because it contains a space.

-tcid *tc_number*
Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive in which you are placing the necessary tapes.

-noaction Directs the command to produce the list of tapes necessary to perform the indicated restore without actually performing the operation.

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

Description

The **bak restoreft** command restores the contents of each fileset indicated with the **-fileset** option from tape to the indicated site (File Server machine and aggregate). By default, restores are full, recreating the fileset as it existed when it was last dumped. A full restore includes data from the last full dump and all subsequent incremental dumps (if any). If incremental dumps exist, you are prompted to insert the necessary tapes into the tape drive. To have the command produce a list of the tapes that the Backup System would need to perform the indicated restore without actually performing the operation, include the **-noaction** option with the command.

You can also choose to do a date-specific restore by including the **-date** option. A date-specific restore returns the fileset to the state it was in at its last dump before the indicated date. Rather than including all dumps to the final one done, it includes only the last full dump and any incremental dumps done before the indicated date.

The precise effect of a restore depends on whether the fileset currently exists in the file system and whether you want to preserve its current state. To replace the current contents of a fileset with data restored from tape, omit the **-extension** option. The results are as follows:

- If the **-server** and **-aggregate** options specify the fileset's current site, the restored data overwrites the fileset's current contents. There is no change in the Fileset Location Database (FLDB) entry for the fileset.
- If the **-server** and **-aggregate** options specify a new site, the restored data is stored in a new fileset at the indicated site. If you name a new site and the fileset to be restored currently exists at its old site, you must do one of the following before issuing the command:
  - — Use the **fts zap** command to delete the existing fileset. The fileset continues to use its existing FLDB entry and fileset ID number, and the fileset's FLDB entry is updated to record the new site.
  - — Use the **fts delete** command to delete the existing fileset and its FLDB entry. The fileset receives a new FLDB entry and a new fileset ID number.

  Using the **fts zap** command is the better approach because it preserves the fileset's existing ID number, which allows Cache Managers to continue to access the fileset without updating their tables of mappings between fileset names and fileset ID numbers. The **bak restoreft** command fails if you do not use the **fts zap** or **fts delete** command to delete the existing fileset before using the **bak restoreft** command to restore the fileset to the new site.

To preserve a fileset's current contents but also introduce a restored version into the file system, use the **-extension** option. A new fileset at the site specified with the **-server** and **-aggregate** options then contains the restored data. It has the same name as the current fileset, with the addition of the distinguishing extension. The Fileset Location (FL) Server automatically assigns the new fileset a fileset ID number and a new FLDB entry, which records all of the appropriate information about the new fileset.

You can also restore a fileset that no longer exists in the file system. A new fileset at the site specified with the **-server** and **-aggregate** options is created to contain the restored data.

Data can be dumped and restored between different types of file systems. For example, data dumped from a DCE LFS fileset can be restored to a DCE LFS fileset or to any type of nonLFS fileset; likewise, data dumped from a nonLFS fileset can be restored to a DCE LFS fileset or to a different type of nonLFS fileset. (See your vendor's documentation to verify the level of support for dump and restore operations between different types of file systems.)

Restored data is translated into the appropriate format for the file system to which it is restored. Note that incompatible information may be lost when a fileset is dumped and restored between different types of file systems. For example, ACLs on objects in a DCE LFS fileset may be lost if the fileset is restored to a file system that does not support ACLs.

Use the **bak restoredisk** command to restore the contents of an entire aggregate. Use the **bak restoreftfamily** command to restore a fileset family or to restore one or more filesets to the same site or to different sites.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines and in the **admin.ft** file on the File Server machine to which filesets are to be restored.

Cautions

Overwriting an existing fileset destroys any files created in the current fileset after the date of the last dump included in the restore. It is always safer to preserve the current fileset by using the **-extension** option to restore data to a new fileset.

Output

If you do not include the **-noaction** option, the **bak restoreft** command returns the unique dump ID number associated with the restore operation. The dump ID is displayed in the command window directly following the command line and in the Tape Coordinator's monitoring window if the **butc** command is issued with debug level 1. The dump ID number is not the same as the job ID number visible with the **(bak) jobs** command if the **bak restoreft** command is issued in interactive mode.

If you include the **-noaction** option, a **Tapes needed:** header is displayed, followed by a list of the tapes necessary to complete the restore operation. No dump ID number is reported because none is assigned.

Examples

The following command restores the fileset named *user.pat* to the aggregate named **/dev/lv01** on the File Server machine named */.../abc.com/hosts/fs5*:

```
$ bak restoreft /.../abc.com/hosts/fs5 /dev/lv01 user.pat


Starting restore
bak: dump ID of restore operation: 187
bak: Finished doing restore
```

Related Information

Commands: **bak dump(8dfs)**, **bak restoredisk(8dfs)**, **bak restoreftfamily(8dfs)**, **fts delete(8dfs)**, **fts zap(8dfs)**.

Files: **dfstab(4dfs)**.

bak restoreftfamily

Purpose

bak restoreftfamily – Restores a fileset family or one or more specified filesets from tape

Synopsis

bak restoreftfamily {**-family** *fileset_family_name* | **-file** *filename*} [**-tcid** *tc_number*] [**-noaction**][**-help**]

Alias

bak familyrestore

Options

-family *fileset_family_name*

Specifies a fileset family to be restored. The command restores all of the filesets in each of the fileset entries in the specified fileset family. Refer to the section entitled **Using the -family Option** for information about using this option. Either this option or the **-file** option must be specified.

-file *filename*

Specifies the full pathname of a file from which the command is to read the name of each fileset to be restored and the site (File Server machine and aggregate) to which the fileset is to be restored. Specify each fileset and site on a separate line, using the following format:

```
machine aggregate fileset
```

Refer to the section entitled **Using the -file Option** for information about using this option. Either this option or the **- family** option must be specified.

-tcid *tc_number*
Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive in which you are placing the necessary tapes. If this option is omitted the TCID defaults to 0 (zero).

-noaction Directs the command to produce a list of filesets it would restore without actually restoring the filesets. The command also provides additional information, such as the tapes that contain dumps of the filesets and the sites to which the filesets would be restored. Include the other options as you would to actually execute the command. You can use this option with the **-family** option to write a list of filesets to a file, which you can then modify for use with the **-file** option. See the section of this reference page entitled **Output** for information about using the **-noaction** option.

-help Prints help for this command. All other valid options specified with this option are ignored.

Description

The **bak restoreftfamily** command restores the contents of specified filesets from tape to the file system. The command performs a full restore of each indicated fileset, restoring data from the last full dump and all subsequent incremental dumps (if any) of each fileset. Use the **- family** option or the **-file** option to indicate the filesets to be restored, as follows:

- The **-family** option lets you restore all of the filesets included in the fileset entries in a specified fileset family. The command reads the Fileset Location Database (FLDB) to determine the filesets to be restored and restores them to their current sites.
- The **-file** option lets you restore specific individual filesets that have entries in a specified file. The command restores each fileset to the site you specify.

The **-noaction** option instructs the command to produce a list of the filesets it would restore without actually restoring any filesets. The command also provides information about the tapes that contain dumps of the filesets. You can use the **-noaction** option with the **- file** option to determine the tapes required to restore the indicated filesets. You can also use the **-noaction** option with the **- family** option to construct a list of filesets that would be restored with a specified fileset family; you can then modify the list of filesets as necessary to produce a file for use with the **-file** option.

The **bak restoreftfamily** command is useful for recovering from catastrophic losses of data, such as the loss of all filesets on multiple aggregates of a File Server machine or the loss of multiple aggregates from multiple File Server machines. In such cases, the command provides a better approach to recovery than the **bak restoreft** command or the **bak restoredisk** command because

- It allows you to restore either individual filesets or specialized collections of filesets.
- It allows you to restore different filesets to different sites.

Conversely, the **bak restoreft** command restores one or more filesets to a single site, and the **bak restoredisk** command restores all filesets that reside on a single aggregate to a single aggregate. The **bak restoreftfamily** command provides greater breadth to a restore operation than the other commands that restore data, which instead provide convenient depth.

Regardless of the command used, data can be dumped and restored between different types of file systems. For example, data dumped from a DCE LFS fileset can be restored to a DCE LFS fileset or to any type of non-LFS fileset; likewise, data dumped from a non-LFS fileset can be restored to a DCE LFS fileset or to a different type of non-LFS fileset. (See your vendor's documentation to verify the level of support for dump and restore operations between different types of file systems.)

Restored data is translated into the appropriate format for the file system to which it is restored. Note that incompatible information may be lost when a fileset is dumped and restored between different types of file systems. For example, ACLs on objects in a DCE LFS fileset may be lost if the fileset is restored to a file system that does not support ACLs.

Using the -family Option

Use the **-family** option of the **bak restoreftfamily** command to restore the filesets included in a fileset family. The command reads the FLDB to determine all filesets that satisfy the fields of the entries in the specified fileset family. It then looks in the Backup Database to determine the tapes that contain the last full dump and all subsequent incremental dumps of each fileset. It restores each fileset included in an entry in the fileset family to its current site, overwriting an existing version of the fileset.

You can specify the name of an existing fileset family, or you can define a new fileset family and add entries that correspond to the filesets that need to be restored. For example, suppose you need to restore all filesets that reside on the File Server machines named **fs1.abc.com** and **fs2.abc.com**. You can use the **bak addftfamily** command to

create a new fileset family. You can then use the **bak addftentry** command to add the following entries to the new fileset family:

```
/.../abc.com/hosts/fs1  .*  .*
/.../abc.com/hosts/fs2  .*  .*
```

These entries indicate all filesets on all aggregates on the machines named **fs1.abc.com** and **fs2.abc.com**. Once the new fileset family is defined, you can issue the **bak restoreftfamily** command, specifying the name of the fileset family with the command's **- family** option.

When you create fileset families for use with the **bak restoreftfamily** command, define entries that match the read/write versions of filesets. The command automatically appends a **.backup** extension to the name of a fileset if it can find no record in the Backup Database of a backup performed for the read/write version. You can include a **.backup** extension to match the backup versions of filesets only if the backup versions were dumped to tape and the backup versions are still valid in the FLDB entries for the filesets.

Using the -file Option

Use the **-file** option of the **bak restoreftfamily** command to restore each fileset that has an entry in a specified file. The command examines the Backup Database to determine the tapes that contain the last full dump and all subsequent incremental dumps of each specified fileset and each fileset to the site indicated in the specified file. It does not consult the FLDB.

An entry for a fileset in a file to be used with the command must have the following format:

```
machine aggregate fileset [comments ...]
```

The entry provides the following information:

machine Specifies the File Server machine to which the fileset is to be restored. Identify the machine by its DCE pathname (for example, */.../abc.com/hosts/fs1*), its host name (for example, **fs1.abc.com**), or its IP address (for example, **11.22.33.44**).

aggregate Specifies the aggregate to which the fileset is to be restored. Identify the aggregate by its device name (for example, **/dev/lv01**) or by its aggregate name (for example, **lfs1**). These names are specified in the first and second fields of the entry for the aggregate in the *dcelocal*/**var/dfs/dfstab** file.

fileset Specifies the fileset to be restored. Specify the name of the read/write version of the fileset, even if the backup version of the fileset was actually dumped. The command automatically appends a **.backup** extension to the name of the fileset if it can find no record in the Backup Database of a backup performed for the read/write version. (Note that you can specify the name of the backup version of the fileset if the backup version was dumped to tape.)

comments ... All remaining text. The command treats any other text provided with the entry for the fileset as a comment and ignores it. Any additional text is optional.

Do not use wildcards (for example, **.***) in an entry. Also, do not include a newline character in an entry. Each entry must appear on a single line of the file. The command uses only the first line for a given fileset; it ignores all subsequent lines for the fileset.

If you restore a fileset to the site at which it currently exists, the command overwrites the existing version of the fileset. If you restore a fileset to a site other than the site at which it currently exists, you must do one of the following before issuing the command:

- Use the **fts zap** command to delete the existing fileset. The restored fileset continues to use its existing FLDB entry and fileset ID number, and the fileset's FLDB entry is updated to record the new site.
- Use the **fts delete** command to delete the existing fileset and its FLDB entry. The restored fileset receives a new FLDB entry and a new fileset ID number.

Using the **fts zap** command is the better approach because it preserves a fileset's existing ID number, which allows Cache Managers to continue to access the fileset without updating their tables of mappings between fileset names and fileset ID numbers. The **bak restoreftfamily** command fails if you do not use the **fts zap** or **fts delete** command to delete an existing fileset before using the **bak restoreftfamily** command to restore the fileset to a new site.

Privileges Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines

and in the **admin.ft** file on each File Server machine to which one or more filesets are to be restored.

Output

If you do not include the **-noaction** option, the **bak restoreftfamily** command returns the unique dump ID number associated with the restore operation. The dump ID is displayed in the command window directly following the command line and in the Tape Coordinator's monitoring window if the **butc** command is issued with debug level 1. The dump ID number is not the same as the job ID number visible with the **(bak) jobs** command if the **bak restoreftfamily** command is issued in interactive mode. Note that the dump ID and job ID numbers are not assigned to the operation until the command actually begins to restore filesets.

If you include the **-noaction** option, the command displays on standard output the number of filesets that would be restored, followed by a separate line of information about each fileset. For each fileset, the command provides the following output:

```
machine aggregate fileset_dumped # as fileset_restored; tape
tape_name; pos position_number; date
```

The output provides the following information:

machine The host name of the File Server machine to which the fileset would be restored (for example, **fs1.abc.com**).

aggregate The aggregate name of the aggregate to which the fileset would be restored (for example, **lfs1**).

fileset_dumped
The name of the fileset that was dumped (for example, *user.frost*). The command can display the name of the backup version of the fileset (for example, *user.frost.backup*) if that version was dumped.

fileset_restored
The name with which the fileset would be restored (for example, *user.frost*). The command always displays the name of the read/write version of the fileset.

tape_name The name of the tape that contains the dump set with which the fileset was dumped (for example, **user.full.1**).

position_number
The position of the fileset with respect to other filesets on the tape that contains the dump set (for example, **31**).

date The date and time at which the fileset was dumped (for example, **Wed Jul 13 05:59:01 1994**).

The command displays information only for filesets that have been dumped to tape; for each fileset that has not been dumped, the command displays an error message on standard error output. The command reads the Backup Database to determine everything but the *machine*,*aggregate*, and *fileset_dumped* information. If you use the **- noaction** option with the **-file** option, the *machine* , *aggregate*, and *fileset_dumped* information must be provided in the specified file; if you use the **-noaction** option with the **-family** option, the command examines the FLDB to determine this information, so it provides information only for filesets that have entries in the FLDB.

The command displays multiple lines of information for a fileset if one or more incremental dumps were performed since the last full dump of the fileset. The command displays one line of output for the last full dump and one line of output for each incremental dump. It displays the lines in the order in which the dumps would need to be restored, beginning with the full dump. It does not necessarily present all of the lines for a fileset consecutively.

If you intend to write the output of the **-family** and **- noaction** options to a file for use with the **-file** option, include only a single line for each fileset; the command ignores any additional lines for a fileset. You can include any line for the fileset; all lines name the fileset's current site. You do not need to remove the **#** (number sign) and the information that follows it; the command ignores any characters that follow the third argument on a line.

When the **-noaction** option is included, no dump ID and job ID numbers are reported because none are assigned.

Notes

The amount of time required for the **bak restoreftfamily** command to complete depends on the number of filesets to be restored. However, a restore operation that includes a large number of filesets can take hours to complete. To reduce the amount of time required for the operation, you can execute multiple instances of the command simultaneously, specifying disjoint fileset families with each command if you use the **-family** option, or indicating files that list different filesets with each command if you use the **- file** option. Depending on how the filesets to be restored were dumped to

tape, specifying disjoint fileset families can also enable you to make the most efficient use of your backup tapes when many filesets need to be restored.

Examples

The following command restores all filesets included in entries in the fileset family **data.restore**, which was created expressly to restore data to a pair of File Server machines on which all data was corrupted due to a software error. All filesets are restored to the sites recorded in their entries in the FLDB.

```
$ bak restoreftfam data.restore
```

```
Starting restore
bak: dump ID of restore operation: 112
bak: Finished doing restore
```

The following command restores all filesets that have entries in the file named **/tmp/restore**:

```
$ bak restoreftfam -file /tmp/restore
```

```
Starting restore
bak: dump ID of restore operation: 113
bak: Finished doing restore
```

The file **/tmp/restore** has the following contents:

```
/.../abc.com/hosts/fs1 /dev/lv01 user.abhijit
/.../abc.com/hosts/fs1 /dev/lv01 user.vijay
/.../abc.com/hosts/fs1 /dev/lv01 user.pierette
/.../abc.com/hosts/fs2 /dev/lv01 user.frost
/.../abc.com/hosts/fs2 /dev/lv01 user.wvh
```

. . .
. . .

Related Information

Commands: **bak addftentry(8dfs)**, **bak addftfamily(8dfs)**, **bak dump(8dfs)**, **bak restoredisk(8dfs)**, **bak restoreft(8dfs)**, **fts delete(8dfs)**, **fts zap(8dfs)**

Files: **dfstab(4dfs)**

bak rmdump

Purpose

bak rmdump – Deletes a dump level from the Backup Database

Synopsis

bak rmdump -level *dump_level* [**-help**]

Options

-level *dump_level*
Names the dump level to be deleted; specify the complete pathname for the dump level to be removed, including any necessary / (slashes).

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

Description

The **bak rmdump** command deletes the dump level indicated with the **-level** option from the dump hierarchy in the Backup Database. If the dump level is a parent, all dump levels that are its children (and their children, if any) are also deleted.

Examples

The following command deletes the dump level called **week3** from the dump hierarchy:

```
$ bak rmd /week3
```

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Related Information

Commands: **bak adddump(8dfs)**, **bak dump(8dfs)**, **bak lsdumps(8dfs)**.

bak rmftentry

Purpose

bak rmftentry – Deletes a fileset family entry from a fileset family

Synopsis

bak rmftentry -family *fileset_family_name* **-entry** *fileset_entry_index* [**-help**]

Options

-family *fileset_family_name*
Names the fileset family from which to delete the entry.

-entry *fileset_entry_index*
Identifies the fileset family entry to delete. The legal value is the fileset entry index number, a positive integer. The **bak lsftfamilies** command displays the index number of each fileset family entry in a fileset family (the first entry defined has index 1, the second 2, and so on).

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

Description

The **bak rmftentry** command deletes the indicated fileset family entry from the fileset family specified with **-family**. Use **-entry** to identify the fileset family entry by its index number.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command deletes the fourth fileset family entry from the fileset family called **sys**. The issuer first used the **bak lsftfamilies** command to determine that the index number of the fileset family entry to be deleted is 4.

```
$ bak rmfte sys 4
```

Related Information

Commands: **bak addftentry(8dfs)**, **bak lsftfamilies(8dfs)**.

bak rmftfamily

Purpose

bak rmftfamily – Deletes a fileset family from the Backup Database

Synopsis

bak rmftfamily -family *fileset_family_name*… [**-help**]

Options

-family *fileset_family_name*
Names each fileset family to be deleted.

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

Description

The **bak rmftfamily** command deletes each fileset family specified by **-family** from the Backup Database. It also deletes the fileset family entries contained in each deleted family. The **bak addftfamily** command is used to add fileset families to the Backup Database.

Use the **bak lsftfamilies** command to list the fileset families currently defined in the Backup Database. Use the **bak rmftentry** command to remove a currently defined fileset family entry from the Backup Database.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command deletes the fileset family called **user**:

```
$ bak rmftf user
```

Related Information

Commands: **bak addftfamily(8dfs)**, **bak lsftfamilies(8dfs)**, **bak rmftentry(8dfs)**.

bak rmhost

Purpose

bak rmhost – Removes a Tape Coordinator entry from the Backup Database

Synopsis

bak rmhost [**-tcid** *tc_number*] [**-help**]

Options

-tcid *tc_number*
: Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator to be removed. Legal values are integers from 0 to 1023. If this option is omitted, a value of **0** (zero) is used.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bak rmhost** command deletes the indicated Tape Coordinator entry from the Backup Database. The Backup Server no longer sends requests to that Tape Coordinator, even if it is still operational on the machine. Repeat this command once for each Tape Coordinator whose entry is to be deleted.

The mapping between the TCID of a Tape Coordinator and the device name of the drive with which it is associated is recorded in the **TapeConfig** file on the Tape Coordinator machine. Remove the entry for a Tape Coordinator from the **TapeConfig** file when you remove its entry from the Backup Database.

Enter the **bak addhost** command to add an entry for a Tape Coordinator to the Backup Database. Enter the **bak lshosts** command to list the Tape Coordinators that have entries in the Backup Database.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command removes the entry for the Tape Coordinator with a TCID of **1** from the Backup Database:

```
$ bak rmhost 1
```

Related Information

Commands: **bak addhost(8dfs)**, **bak lshosts(8dfs)**.

Files: **TapeConfig(4dfs)**.

bak savedb

Purpose

bak savedb – Creates a backup copy of the Backup Database

Synopsis

bak savedb [**-tcid** *tc_number*] [**-help**]

Options

-tcid *tc_number*
: Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive to which the database is to be backed up. If omitted, it defaults to **0** (zero).

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bak savedb** command creates a backup copy of the entire Backup Database. Designate one tape as the Backup Database tape; label it with the name **bak_db_dump.1** (it must have this name). The **-tcid** option specifies the TCID of the Tape Coordinator to which to save the Backup Database; this option is necessary only if the TCID is not **0** (zero).

If the Backup Database is damaged, delete the old database and use the **bak restoredb** command to restore a new version from tape. Use the **bak verifydb** command to determine if the Backup Database is damaged.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command backs up the Backup Database to a tape in the Tape Coordinator with a TCID of **3**:

```
$ bak save 3
```

Related Information

Commands: **bak restoredb(8dfs)**, **bak verifydb(8dfs)**.

bak scantape

Purpose

bak scantape – Extracts dump set information from a tape

Synopsis

bak scantape [**-dbadd**][**-tcid** *tc_number*] [**-help**]

Options

-dbadd Adds the information extracted from the tape to the Backup Database if the tape is completely undamaged and the Backup Database does not already contain an entry with the same dump ID number.

-tcid *tc_number*
Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive containing the tape. If omitted, it defaults to **0** (zero).

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

Description

The **bak scantape** command reads the tape in the drive controlled by the Tape Coordinator indicated by **-tcid**, extracting information from the tape label and from the fileset header of each fileset on the tape. It does not extract actual data from the filesets, though the information it does extract is needed to restore the data using the Backup System.

The Tape Coordinator displays the information about each fileset in its monitoring window as it extracts the information. The Tape Coordinator checks for damage to the tape medium by checking for the presence of special markers it expects to find at the start and end of each fileset. If the Tape Coordinator does not find an expected marker, it concludes that the tape medium is damaged, and the command aborts.

If the **-dbadd** option is provided, the program creates a Backup Database entry for the tape and records the extracted information in the entry. It is not possible to extract information about only specific filesets on a tape. Because only data about all of the

filesets on a tape can be extracted, this command works only if a tape is completely undamaged.

The Tape Coordinator does not require that the issuer insert all of the tapes containing a dump set unless a fileset is split across two tapes. In that case, it automatically prompts for the tape with the next highest index to extract complete information about the fileset. If **-dbadd** is used, information from both tapes is added to the database.

If a tape contains only complete filesets, the Tape Coordinator reads the tape and prompts

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

If the issuer responds **n**, the command exits, adding the information from the tape to the Backup Database if **-dbadd** is used. If the issuer responds **y**, the Tape Coordinator prompts for the tape with the next highest index.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Cautions

Using the **-dbadd** option with this command introduces the possibility that two database entries will appear almost the same; you will need to track which physical tape corresponds to which entry.

Database entries are identified by three elements: the tape name, the dump level pathname, and a dump ID number, which is unique for every dump set. This command creates a database entry for the dump set on the tape as long as its dump ID number is different from any existing entry's ID number, even if the entry has the same tape name and dump level name as an existing entry.

Output

The **bak scantape** command first displays the following information from the label of the tape:

name The tape label, in the format *fileset_family_name.dump_level.index*.

createTime The date and time at which the Backup System started executing the dump operation that created this dump set.

cell The cell in which the dump set was created.

size The tape's capacity in kilobytes (not the amount of data on the tape). The value comes from the tape label and is derived from **bak labeltape** or the **TapeConfig** file rather than from a measurement of the tape.

dump path The dump level used in creating the dump set.

dumpID The dump ID number of the dump on the tape.

useCount The number of times data has been dumped to this tape.

The command then displays an entry for each fileset. The entries appear in the order in which the filesets are encountered on the tape. If a fileset is split across two tapes, there is a separate entry for both fragments. Each entry includes the following information:

fileset name The name of the fileset, with a **.backup** or **.readonly** extension if appropriate.

fileset ID The fileset ID number of the fileset.

dumpSetName
The dump set to which the fileset belongs.

dumpID The dump ID number of the dump set named by **dumpSetName**.

level The depth in the dump hierarchy of the dump level used in creating the dump set. A value of **0** (zero) indicates a full dump set. A value of **1** or greater indicates an incremental dump set made at the indicated depth in the hierarchy. The value reported is for the entire dump, not necessarily for the fileset itself. (For example, it is possible for an individual fileset to be dumped at a higher level if it was omitted from a previous dump set.)

parentID The dump ID number of **dumpSetName**'s parent dump set. (A parent dump set is a dump set made at the level that serves as the parent for a dump level.) This should be **0** (zero) if **level** is **0** (zero).

endTime Should always be **0** (zero); it is for internal use only.

clonedate The date and time at which the fileset was created. For a backup or read-only fileset, this represents the time when it was cloned from its read/write source fileset. For a read/write fileset, it indicates when the Backup System accessed the fileset to include it in **dumpSetName**.

The following error message (usually preceded by other, more specific messages) indicates that the program has not encountered one of the markers it expects to find at the start and end of each fileset and has concluded that the tape is damaged. No data from this tape can be incorporated into the Backup Database.

```
aborting - this dump cannot be processed correctly
```

Examples

The following command shows the output from a tape's label and for the first fileset on the tape:

$ bak scantape

```
Tape label
-- ---- ----
name =          guests.monthly.1
createTime =    Fri Nov 22 05:59:31 1990
cell =          /.../abc.com
size =          20103324 Kbytes
dump path =     /monthly
dump id =       729369701
useCount =      1
-- End of tape label --


-- fileset --
fileset name: user.guest10.backup
fileset ID 0,,112262
dumpSetName: guests.monthly
dumpID 729369701
level 0
parentID 0
endTime 0
clonedate Fri Nov 22 05:36:29 1991
```

Related Information

Commands: **bak deletedump(8dfs)**, **bak dump(8dfs)**, **bak restoredisk(8dfs)**, **bak restoreft(8dfs)**, **kill** (see the **bak(8dfs)** reference page for information about the **kill** command).

bak setexp

Purpose

bak setexp – Sets the expiration date on an existing dump level

Synopsis

bak setexp -level *dump_level*… [**-expires** *date*]… [**-help**]

Options

-level *dump_level*

Names each dump level whose expiration date is to be set. Provide the full pathname for each dump level, including all necessary / (slashes).

-expires *date*

Defines the expiration date to be associated with each dump level. Expiration dates can be specified as absolute or relative values. Absolute expiration dates have the format

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

The word **at** is followed by a date (*month*/*day*/*year*) and, optionally, a time (*hours*:*minutes*). Values that can be interpreted for *yy* run from 00 to 37, which are interpreted as the years 2000–2037, and from 70 to 99, which are interpreted as 1970–1999. Values between 38 and 69 cannot be interpreted because the years to which they correspond (2038–2069) exceed the capacity of the standard UNIX representation of dates (the number of seconds since 12:00 a.m. on 1 January 1970). Values between 38 and 69 are reduced to 2038.

If provided, the time must be in 24-hour format (for example, **20:30** for 8:30 p.m.). If omitted, the time defaults to **00:00** (12:00 a.m.).

Relative expiration dates have the format

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

The word **in** is followed by a number of years (maximum 9999), months (maximum 11), and days (maximum 30), or a combination of these arguments. At least one of the three must be provided, and the appropriate unit abbreviation (**y**, **m**, or **d**) must always accompany a value. If more than one of the three is provided, they must appear in the order shown. As with absolute dates, a number of years that causes the relative time to exceed the year 2038 is effectively truncated to the number of years remaining until 2038.

If you omit this option, tapes created at the specified dump levels have no expiration dates, meaning they can be overwritten by appropriately named dump sets at any time. Although the **-expires** option is followed by an ellipsis, you can specify only one expiration date. (The ellipsis is included to accommodate the DFS command parser.)

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

Description

The **bak setexp** command sets the expiration date on each dump level specified with **-level**. Each dump level must already exist in the dump hierarchy stored in the Backup Database.

The expiration date is applied to tapes containing dump sets made at the dump level; after the specified date, the Backup System overwrites a tape's contents with acceptably named dump sets without question. The Backup System's attempts to overwrite an unexpired tape fail until the issuer relabels the tape with the **bak labeltape** command. (Because the label records the unexpired expiration date or unacceptable name, erasing the label removes the obstacle to overwriting.) If no expiration date is defined for a tape, the Backup System overwrites the dump set on the tape with a dump set of the same name without question.

Expiration dates can be either absolute or relative:

- Absolute expiration dates are defined as a specific month/day/year and, optionally, hours and minutes. A tape with an absolute expiration date expires at that time,

regardless of when the dump set on it was created. (If the expiration predates the dump set's creation, the tape is immediately treated as expired.)

- Relative dates are defined as a number of years, months, days, or any combination of the three. When the Backup System creates a dump set at the dump level, it calculates the tape's actual expiration date by adding the relative date to the start time of the dump operation.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Examples

The following command associates an absolute expiration date of 10:00 p.m. on 31 December 1990 with the dump level **/90/december**:

```
$ bak setexp /90/december -e at 12/31/90 22:00
```

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

```
$ bak set /monthly/week1 /monthly/week2 -exp 7d
```

Related Information

Command: **bak adddump(8dfs)**, **bak dump(8dfs)**, **bak labeltape(8dfs)**.

bak status

Purpose

bak status – Reports on the operation that a Tape Coordinator is executing

Synopsis

bak status [**-tcid** *tc_number*] [**-help**]

Options

-tcid *tc_number*
: Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for which status information is to be displayed.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bak status** command displays information about the operation currently being performed by the indicated Tape Coordinator. The command displays information about only the Tape Coordinator's current job. It does not display information about any pending jobs waiting for the Tape Coordinator. Use the **jobs** command in interactive mode to display information about the currently executing job and any pending jobs for a Tape Coordinator.

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Output

If the indicated Tape Coordinator is not currently performing an operation, the output reports **Tape coordinator is idle**. Otherwise, it reports the following:

- An operation name describing the operation. One of the following operation names is displayed:

Dump (*dump_set***)**
For a dump operation, where *dump_set* is the name of the dump set in the form *fileset_family_name.dump_level*. Dump operations are initiated with the **bak dump** command.

Restore For a restore operation. Restore operations are initiated with the **bak restoreft**, **bak restoredisk**, or **bak restoreftfamily** command.

Labeltape (*tape_label***)**
For a tape labeling operation, where *tape_label* is the label being placed on the tape. Tape labeling operations are started with the **bak labeltape** command.

Scantape For a tape scanning operation. Tape scanning operations are initiated with the **bak scantape** command.

SaveDb For a database saving operation. Operations that save the Backup Database to tape are started with the **bak savedb** command.

RestoreDb For a database restoring operation. Operations that restore the Backup Database from tape are initiated with the **bak restoredb** command.

- The number of kilobytes transferred so far (from file system to tape for a dump operation, from tape to file system for a restore operation).
- The string **fileset** followed by the name of the fileset currently being restored if the operation is a restore; the string **fileset** followed by the name of the fileset currently being dumped if the operation is a dump.
- Status information about the operation. If the operation is executing normally, no message is displayed; otherwise, one of the following messages is displayed:

[abort requested]
The **kill** command was issued, but the operation is not yet canceled.

[abort sent] The operation is canceled, but its execution is not yet stopped.

[operator wait]
The Tape Coordinator is waiting for the operator monitoring the operation to insert a tape in the drive.

Examples

The following command displays status information about the operation being performed by the Tape Coordinator with a TCID of **4**. The operation is a dump of the dump set whose name is **usersys.monday**. So far, 23,597 bytes have been dumped to tape. The fileset named *user.terry* is currently being dumped. No status message is displayed, indicating the operation is proceeding normally.

```
$ bak status 4


Dump (usersys.monday): 23597 Kbytes transferred, fileset user.terry.
```

bak_restoreftfamily command"

Related Information

Commands: **bak(8dfs)**, **bak dump(8dfs)**, **bak labeltape(8dfs)**, **bak restoredb(8dfs)**, **bak restoredisk(8dfs)**, **bak restoreft(8dfs)**, **bak restoreftfamily(8dfs)**, **bak savedb(8dfs)**, **bak scantape(8dfs)**,

bak verifydb

Purpose

bak verifydb – Checks the status of the Backup Database

Synopsis

bak verifydb [**-verbose**][**-help**]

Options

-verbose Directs the command to provide more information about the Backup Database.

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

Description

The **bak verifydb** command checks the status of the Backup Database. It displays a message indicating whether the Backup Database is undamaged or damaged. If the Backup Database is undamaged, it can be accessed. If it is damaged, it must be restored from tape with the **bak restoredb** command (provided it has been backed up previously with the **bak savedb** command).

Privilege Required

The issuer must be listed in the **admin.bak** files on all Backup Database machines.

Output

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

`Database OK.` Indicates that the database is undamaged and can be used.

`Database not OK.` Indicates that the database is damaged. The database must be deleted and then restored from tape.

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

The **-verbose** option also causes the command to display the name of the machine on which the command is issued.

Examples

The following command verifies that the Backup Database is undamaged:

```
$ bak verifydb
```

```
Database OK.
```

Related Information

Commands: **bak dumpinfo(8dfs)**, **bak ftinfo(8dfs)**, **bak lsdumps(8dfs)**, **bak restoredb(8dfs)**, **bak savedb(8dfs)**.

bakserver

Purpose

bakserver – Initializes the Backup Server

Synopsis

bakserver [adminlist *filename*] [**-verbose**][**-help**]

Options

-adminlist *filename*
: Specifies the file that contains principals and groups authorized to execute **bakserver** RPCs (usually using **bak** commands). If this option is omitted, the **bakserver** obtains the list of authorized users from the default administrative list file, *dcelocal* **/var/dfs/admin.bak**.

-verbose
: Directs the **bakserver** process to provide a detailed report on what it is doing by displaying messages on standard error.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

  The **help** and **apropos** commands available with all command suites are also available with the **bakserver** command. See the **bos help** and **bos apropos** pages for examples of using these commands.

Description

The Backup Server (**bakserver** process) communicates with the Backup Database to perform dump and restore operations. The **bakserver** process must run on all machines that house a copy of the Backup Database. It is usually started and controlled by the BOS Server; if it is not, execute the **bakserver** process as a background process. The binary file for the **bakserver** process resides in *dcelocal* **/bin/bakserver**.

The first time it is initialized, the **bakserver** process creates the Backup Database in *dcelocal* **var/dfs/backup**; all Backup Database files have a root name of **bkdb**. The **bakserver** process also creates the *dcelocal*/**var/dfs/admin.bak** administrative list file if the file does not already exist.

The principals and members of groups in the **admin.bak** administrative list are authorized to issue **bak** commands (which are used for tasks such as examining the database and dumping and restoring data). The list must also include the abbreviated DFS server principals of all Backup Database machines to allow for the proper distribution of changes via the Ubik database synchronization mechanism.

Because the Backup Database is a replicated database, the **admin.bak** administrative lists for all **bakserver** processes in a cell must contain the same principals and groups.

It is recommended that all system administrators using the Backup System be included on the following lists: the **admin.bak** file on all machines housing the Backup Database, the **admin.fl** file on all machines housing the Fileset Location Database (FLDB), and the **admin.ft** file on all File Server machines.

When it is started, the **bakserver** process makes a **ubik_ServerInit** call to register its existence as a server process with the Ubik coordinator. It then listens for incoming RPCs to which to respond.

Each time it is started, the **bakserver** process also creates the *dcelocal***/var/dfs/adm/BakLog** event log file if the file does not already exist. It then appends messages to the file. If the file exists when the **bakserver** process is started, the process moves it to the **BakLog.old** file in the same directory (overwriting the current **BakLog.old** file if it exists) before creating a new version to which to append messages.

Privilege Required

The issuer must be logged in as **root** on the local machine.

Output

If problems are encountered during initialization, the **bakserver** process displays error messages on standard error output. The **bakserver** process keeps an event log in the file *dcelocal* **/var/dfs/adm/BakLog**.

If run with the **-verbose** option, the **bakserver** process provides a detailed report on what it is doing by displaying messages on standard error.

Related Information

Files: **admin.bak(4dfs)**, **BakLog(4dfs)**.

bos

Purpose

bos – Introduction to the **bos** command suite

Options

The following options are used with many **bos** commands. They are also listed with the commands that use them.

-server *machine*

Names the machine running the BOS Server that is to execute the command. To run a privileged **bos** command (a **bos** command that requires the issuer to have some level of administrative privilege) using a privileged identity, always specify the full DCE pathname of the machine (for example, */.../abc.com/hosts/fs1*).

To run an unprivileged **bos** command, you can use any of the following to specify the machine:

- The machine's DCE pathname (for example, */.../abc.com/hosts/fs1*)
- The machine's host name (for example, **fs1.abc.com** or **fs1**)
- The machine's IP address (for example, **11.22.33.44**)

Note: If you specify the host name or IP address of the machine, the command executes using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option); unless DFS authorization checking is disabled on the specified machine, a privileged **bos** command issued in this manner fails. If you specify the machine's host name or IP address, the command displays the following message (using the **-noauth** option suppresses the message):

```
bos:  WARNING: short form for server used;
no authentication information will be sent to
the bosserver
```

-noauth Directs the **bos** program to use the unprivileged identity **nobody** as the identity of the issuer of the command. Generally, the **-noauth** option is included with a command if DFS authorization checking is disabled on the server machine whose BOS Server is to execute the command or if the Security Service is unavailable. If DFS authorization checking is disabled, the BOS Server requires no administrative privilege to issue any command; any user, even the identity **nobody**, has sufficient privilege to perform any operation. If the Security Service is unavailable, a user's security credentials cannot be obtained.

DFS authorization checking is disabled with the **bos setauth** command or by including the **-noauth** option when the **bosserver** process is started on a machine. DFS authorization checking is typically disabled

- During initial DFS installation
- If the Security Service is unavailable
- During server encryption key emergencies
- To view the actual keys stored in a keytab file

Include the **-noauth** option with a command that requires administrative privilege only if DFS authorization checking is disabled on the necessary machine. A command that requires administrative privilege fails if the **-noauth** option is included and DFS authorization checking is not disabled. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal of the machine on which the command is issued as the identity of the issuer. Each DFS server machine has a DFS server principal stored in the Registry Database. A DFS server principal is a unique, fully qualified principal name that ends with the string **dfs-server**; for example, */.../abc.com/hosts/fs1/dfs-server*. (Do not confuse a machine's DFS server principal with its unique **self** identity.)

Use this option only if the command is issued from a DFS server machine. You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-help Prints the online help for the command. All other valid options specified with this option are ignored. For complete details about receiving help, see the **dfs_intro(8dfs)** reference page.

Description

Commands in the **bos** command suite are used by system administrators to contact the Basic OverSeer (BOS) Server. The BOS Server runs on every DFS server machine to monitor the other DFS server processes on the machine. It restarts processes automatically if they fail. The BOS Server also provides an interface through which system administrators can start and stop processes and check on server status.

The files described in the following sections are used to store configuration, administrative, and security information.

The BosConfig File

The *dcelocal*/**var/dfs/BosConfig** file on the local disk of each DFS server machine contains information about the processes the BOS Server is to monitor. This information includes the process type, the command parameters associated with the process, and a status flag that tells the BOS Server to start the process at initialization or restart the process if the process fails. Whenever the BOS Server starts or restarts, it reads the file to learn which processes to monitor; this information is transferred into memory and the file is not read again until the BOS Server next restarts.

The administrator can change the process status in the BOS Server's memory with specific **bos** commands; therefore, it is possible for a process to stop running even if its status flag in the BosConfig file is set to **Run**. Similarly, an administrator can start a process without setting its status flag in the **BosConfig** file to **Run** by changing its memory state flag to **Run**.

Never edit the **BosConfig** file directly; always use the appropriate **bos** commands. Editing the file directly can introduce changes of which the BOS Server is unaware. The BOS Server does not recognize such changes until it is restarted and again reads the file.

The admin.bos File

The *dcelocal*/**var/dfs/admin.bos** file on the local disk of each File Server machine contains the names of users who are allowed to issue **bos** commands on that machine. All users can list the contents of the file with the **bos lsadmin** command; only administrative users can edit the contents of the file with the **bos addadmin** and **bos rmadmin** commands. Because the **admin.bos** file is a binary file, you cannot edit it directly; you must use the appropriate **bos** commands.

The Keytab File

A **/krb5/v5srvtab** keytab file is stored on the local disk of each File Server machine. A keytab file contains the list of server encryption keys used by a server process

on that machine to decrypt tokens presented by clients. The server process interacts only with clients possessing tokens encrypted with server encryption keys listed in the appropriate keytab file.

The keys in a keytab file are marked with a unique key version number. All tokens presented by clients are also marked with a key version number; a server process uses the key version number to determine which key to use to decrypt a token.

Only administrative users can examine, add, and remove keys in the keytab file. Never edit a keytab file directly; always use the appropriate **bos** commands.

Receiving Help

There are several different ways to receive help about DFS commands. The following examples summarize the syntax for the different help options:

$ **man bos** Displays the reference page for the command suite.

$ **man bos_** *command*
Displays the reference page for an individual command. You must use an _ (underscore) to connect the command suite to the command name. *Do not use the underscore when issuing the command in DFS.*

$ **bos help** Displays a list of commands in a command suite.

$ **bos help***command*
Displays the syntax for a single command.

$ **bos apropos -topic** *string*
Displays a short description of any commands that match the specified *string*.

Consult the **dfs_intro(8dfs)** reference page for complete information about the DFS help facilities.

Privilege Required

All **bos** commands can be issued by users listed in the **admin.bos** file on the machine whose BOS Server is executing the command. Specific privilege information is listed with each command's description. In addition, if the BOS Server is running with DFS authorization checking disabled, no privilege is required to issue **bos** commands.

Cautions

Never directly edit a **BosConfig** file, a keytab file, an **admin.bos** file, or any administrative (**admin**) file; always use the appropriate commands from the **bos** command suite.

Related Information

Commands: **bos addadmin(8dfs)**, **bos addkey(8dfs)**, **bos apropos(8dfs)**, **bos create(8dfs)**, **bos delete(8dfs)**, **bos gckeys(8ds)**, **bos genkey(8dfs)**, **bos getdates(8dfs)**, **bos getlog(8dfs)**, **bos getrestart(8dfs)**, **bos help(8dfs)**, **bos install(8dfs)**, **bos lsadmin(8dfs)**, **bos lscell(8dfs)**, **bos lskeys(8dfs)**, **bos prune(8dfs)**, **bos restart(8dfs)**, **bos rmadmin(8dfs)**, **bos rmkey(8dfs)**, **bos setauth(8dfs)**, **bos setrestart(8dfs)**, **bos shutdown(8dfs)**, **bos start(8dfs)**, **bos startup(8dfs)**, **bos status(8dfs)**, **bos stop(8dfs)**, **bos uninstall(8dfs)**, **dfs_intro(8dfs)**, **keytab(8dce)**.

Files: **admin.bak(4dfs)**, **admin.bos(4dfs)**, **admin.fl(4dfs)**, **admin.ft(4dfs)**, **admin.up(4dfs)**, **BosConfig(4dfs)**, **v5srvtab(5sec)**.

bos addadmin

Purpose

bos addadmin – Adds a user, group, or server to an administrative list

Synopsis

bos addadmin -server *machine* **-adminlist** *filename* [**-principal** *name*…] [**-group** *name*…] [**-createlist**][{**-noauth** | | | **-localauth** }] [**-help**]

Options

-server *machine*

Names the server machine that houses the administrative list to which principals, groups, or both are to be added. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-adminlist *filename*

Names the administrative list to which principals, groups, or both are to be added. The complete pathname is unnecessary if the list is stored in the default configuration directory (*dcelocal*/**var/dfs**).

-principal *name*

Specifies the principal name of each user or server machine to be added to the administrative list. A user from the local cell can be specified by a full or abbreviated principal name (for example, /**...**/*cellname*/*username* or just *username*); a user from a foreign cell can be specified only by a full principal name. A server machine from the local cell can be specified by a full or abbreviated principal name (for example, /**...**/*cellname* /**hosts**/*hostname*/**self** or just **hosts**/*hostname* /**self**); a server machine from a foreign cell can be specified only by a full principal name.

-group *name*
Specifies the name of each group to be added to the administrative list. A group from the local cell can be specified by a full or abbreviated group name (for example, **/...**/*cellname /group_name* or just *group_name*); a group from a foreign cell can be specified only by a full group name.

-createlist Specifies that the file indicated with **-adminlist** is to be created if it is not found. Any principals or groups specified with the command are added to the new file; if no principals or groups are specified, the command creates an empty file. This option has no effect if the specified file already exists.

Note: Because the **admin.bos** list must already exist to issue this command, this option is ignored if **admin.bos** is specified with the **-adminlist** option.

-noauth Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos addadmin** command adds the specified users, groups, and servers to the administrative list specified by the **-adminlist** option on the server machine indicated by the **-server** option. The principal (login) names of users and the principal names of server machines to be added to the administrative list are specified with the **-principal** option; the names of groups to be added to the list are specified with the **-group** option. Principals added to the administrative list either directly (with the **-principal** option) or indirectly (as members of groups indicated with the **-group** option) can then issue administrative commands for the DFS server process associated with the list.

The default path for administrative lists is the configuration directory (*dcelocal*/**var/dfs**). If the specified list is stored in the default directory, only the specific filename is required. If the specified list is stored elsewhere, the pathname to the file that was used when the associated server process was started is required.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples

The following command adds the user names **jones** and *smith* to the **admin.bos** file on **fs1**. The administrative list is stored in the default configuration directory.

```
$ bos adda -server /.../abc.com/hosts/fs1 -adminlist admin.bos -principal jones smith
```

Related Information

Commands: **bos lsadmin(8dfs)**, **bos rmadmin(8dfs)**.

Files: **admin.bak(4dfs)**, **admin.bos(4dfs)**, **admin.fl(4dfs)**, **admin.ft(4dfs)**, **admin.up(4dfs)**.

bos addkey

Purpose

bos addkey – Converts a string into a server encryption key and adds it to a keytab file

Synopsis

bos addkey -server *machine* **-kvno** *+_or_version_number* **-password** *string* [**-principal** *name*] [**-localonly**][{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*

Names the server machine whose keytab file is to have a new key added to it. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-kvno *+_or_version_number*

Defines the key version number of the new key. The version number must be one of the following:

- An integer in the range 1 to 255. The command uses the specified integer as the version number of the new key. The integer must be unique for the principal indicated by **-principal** in the keytab file on the server machine specified by **-server**.
- **+** or **0** (zero). The command chooses an integer to serve as the version number of the new key. The integer it chooses is unique for the principal indicated by **-principal** in the Registry Database. However, it may not be unique for the indicated principal in the keytab file on the specified machine, in which case it replaces the key currently associated with the principal/version number pair in the keytab file.

Unless the **-localonly** option is used, the new key and its version number replace the key and version number currently stored in the Registry Database for the indicated principal.

-password *string*
Defines a character string to be converted into an octal string for use as the key. The string serves as a password for the indicated principal. It can include any characters; it can also include spaces if the entire string is enclosed in " " (double quotes).

-principal *name*
Provides the principal name with which the key is to be associated. The default is the DFS principal name of the machine specified by **-server**.

-localonly Specifies that the key is to be added to the keytab file on the machine indicated by **-server**, but that the Registry Database is not to be updated. The default is to add the key to the local keytab file and update the Registry Database accordingly.

-noauth Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos addkey** command associates a new server encryption key with the principal name indicated by **-principal** in the **/krb5/v5srvtab** keytab file on the server machine specified by **-server** and, by default, in the Registry Database. The key is derived from the string specified by **-password** and is given the version number specified by **-kvno**. The issuer can either specify a version number or have the command choose one that is unique for the indicated principal in the Registry Database. If the **-localonly** option

is omitted, the server encryption key and version number for the indicated principal are automatically updated both in the keytab file on the specified server machine and in the Registry Database; if the **-localonly** option is specified, the keytab file is updated, but the Registry Database is not.

The **bos genkey** command is a more secure way of adding a key to a keytab file because it generates a random key. It also always updates the Registry Database. The keytab file must already exist before the **bos addkey** or **bos genkey** command can be used to add a key to it. (Keytab files are created with the **dcecp keytab create** command.)

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**, and, unless the **-localonly** option is used, the DFS server principal of the machine specified by **-server** must have the permissions necessary to alter entries in the Registry Database.

Output

If the packet privacy protection level is not available to you, the command displays the following message reporting that the BOS Server is using the packet integrity protection level instead:

```
Data encryption unsupported by RPC. Continuing without it.
```

Examples

The following command adds a new server encryption key with key version number **14** to the keytab file on **fs1** without updating the Registry Database. Because **-principal** is omitted, the key is associated with the DFS principal name of **fs1** (the machine specified with **-server**). The password string **fourteenth new key** is converted into an octal key before being placed in the keytab file.

```
$ bos addk /.../abc.com/hosts/fs1 14 "fourteenth new key" -localonly
```

Related Information

Commands: **bos gckeys(8dfs)**, **bos genkey(8dfs)**, **bos lskeys(8dfs)**, **bos rmkey(8dfs)**, **keytab(8dce)**.

Files: **v5srvtab(5sec)**.

bos apropos

Purpose

bos apropos – Shows each help entry containing a specified string

Synopsis

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

Options

-topic *string* Specifies the keyword string for which to search. If it is more than a single word, surround the string with " " (double quotes) or other delimiters. Type all strings for **bos** commands in lowercase letters.

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

Description

The **bos apropos** command displays the first line of the help entry for any **bos** command containing the string specified by **-topic** in its name or short description.

To display the syntax for a command, use the **bos help** command.

Privilege Required

No privileges are required.

Output

The first line of an online help entry for a command lists the command and briefly describes its function. This command displays the first line for any **bos** command where the string specified by **-topic** is part of the command name or the first line.

Examples

The following command lists all **bos** commands that have the word **restart** in their names or short descriptions:

$ bos apropos restart

```
getrestart: get restart times
restart: restart all processes
setrestart: set restart times for server processes
```

Related Information

Commands: **bos help(8dfs)**.

bos create

Purpose

bos create – Creates a new process in the **BosConfig** file and starts it

Synopsis

bos create - server *machine* **-process** *server_process* **-type** *process_type* **-cmd** *cmd_line…* [{**-noauth** | | | **-localauth** }] [**-help**]

Options

-server *machine*

Names the server machine on which to create the new process. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-process *server_process*

Names the server process to be created. You can choose any name for a process, but it is recommended that you give the process the same name as its binary file (and use the same name on every machine running that process). The recommended names are

ftserver For the Fileset Server process

flserver For the Fileset Location Server process

upclient For the client portion of the Update Server, which brings common configuration files and binary files from the System Control and Binary Distribution machines

upserver For the server portion of the Update Server process

repserver For the Replication Server process

bakserver For the Backup Server process

Each process runs under the local identity **root** and the DCE identity **self**. However, the process is unauthenticated as far as DFS is concerned.

-type*process_type*

Specifies the process type. Legal values are **simple** and **cron**. Specify **simple** for continuous processes and **cron** for processes that are to run only at specified times.

-cmd *cmd_line*

Specifies the commands the BOS Server runs to start the process and, if **-type** is **cron**, the time the BOS Server executes the command.

For a **simple** *process*, this must be the complete pathname to the binary file for the process (for example, *dcelocal* **/bin/flserver** for the Fileset Location Server). The commands for some **simple** processes take options, in which case the entire argument must be surrounded with " " (double quotes).

For a **cron** *process*, provide two parameters. The first parameter is either the pathname to a binary file to be executed or the complete pathname of a command from one of the DFS suites (complete with all of the necessary arguments). Surround this parameter with " " (double quotes) if it contains spaces.

If the specified executable file does not exist, the **bos create** command does not create an entry in the **BosConfig** file. Instead, the command displays the following message:

```
bos:  failed to create a new server instance instance of
type process_type (specified executable not found)
```

The second parameter for a **cron** process specifies the time when the BOS Server is to execute the command specified by the first parameter. Use a day and time together to execute the command weekly at the specified time; use a time alone to execute the command daily at the specified time. Day and time specifications have the following format:

[*day*] *hh*:*mm*

Enter the name of the day in all lowercase letters, giving either the whole name or the first three letters as an abbreviation (for example, **sunday** or **sun**). Specify the time of day by separating the hours from the minutes with a **:** (colon). Use 24-hour time (for example, **14:30**), or use 1:00 to 12:00 with **am** or **pm** (for example, **"2:30 pm"**). The

time part of the option is optional if the day is specified; if the time is excluded, it defaults to 00:00 on the specified day. As shown in the example, enclose the entire entry in " " (double quotes) if it contains a space.

To execute the command only once, specify **now** instead of a day or a day and time, or issue the command directly; the process entry is removed from the **BosConfig** file after the command is executed. To place the process entry in the **BosConfig** file without ever executing it, specify **never** instead of a time or a day and time.

-noauth Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **- localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos create** command creates a new server process on the server machine specified by **-server** by creating an entry in the **BosConfig** file on the local disk of the machine. The status of the new process entry in both the **BosConfig** file and memory is set to **Run**, and the process is started on the server machine (unless the process is a **cron** process and the second parameter of the **-cmd** option is **never**).

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples

The following command creates the **simple** process **flserver** on the machine named **fs3**:

$ bos create /.../abc.com/hosts/fs3 flserver simple*dcelocal***/bin/flserver**

The following command creates the **cron** process named **backup** on the machine named **fs3**. The **-localauth** option allows the process (which runs unauthenticated) to use the DFS server principal of **fs3** to execute the privileged **fts clonesys** command.

$ bos create /.../abc.com/hosts/fs3 backup cron "*dcelocal***/bin/fts clonesys **
-s /.../abc.com/hosts/fs3 -localauth" 5:30

Related Information

Commands: **bos delete(8dfs)**.

Files: **BosConfig(4dfs)**.

bos delete

Purpose

bos delete – Deletes server processes from the **BosConfig** file

Synopsis

bos delete -server *machine* **-process** *server_process…* [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine from which to delete one or more server processes. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-process *server_process*
: Names each process to delete. Use the name assigned with the **-process** option in the **bos create** command; if necessary, use the **bos status** command to list the possible process names.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bos delete** command removes each indicated server process entry from the **BosConfig** file on the server machine specified by **-server**. Before issuing this command, the issuer must use the **bos stop** command to stop each indicated process, both **simple** and **cron**, running on **-server**. An error message results if the status flag of a process being deleted is **Run** when this command is issued.

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples

The following command removes the **flserver** process entry from the **BosConfig** file on the machine named **fs3**:

```
$ bos delete /.../abc.com/hosts/fs3 flserver
```

Related Information

Commands: **bos create(8dfs)**.

Files: **BosConfig(4dfs)**.

bos gckeys

Purpose

bos gckeys – Removes obsolete server encryption keys from a keytab file

Synopsis

bos gckeys -server *machine* [**-principal** *name*] [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine whose keytab file is to have obsolete keys removed from it. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-principal *name*
: Provides the principal name for which obsolete keys are to be removed from the keytab file. The default is the DFS principal name of the machine specified by **-server**.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bos gckeys** command removes obsolete server encryption keys from the **/krb5/v5srvtab** keytab file on the server machine specified by **-server**. Obsolete keys associated only with the principal name specified by **-principal** are removed from the keytab file; the DFS principal name of the server machine specified with **-server** is used by default.

Keys are removed based on age and lack of use. The removal process, referred to as *garbage collection*, affects only the keytab file stored on the local disk of the machine indicated by **-server**; it has no effect on the Registry Database.

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**.

Output

If the packet privacy protection level is not available to you, the command displays the following message reporting that the BOS Server is using the packet integrity protection level instead:

```
Data encryption unsupported by RPC. Continuing without it.
```

Examples

The following command removes obsolete keys associated with the principal **hosts/fs1/dfs-server** from the keytab file on the server machine named */.../abc.com/hosts/fs3*. Note that the keys being removed are associated with the principal name of a machine different from the one whose BOS Server is executing the command.

```
$ bos gckeys /.../abc.com/hosts/fs3 hosts/fs1/dfs-server
```

Related Information

Commands: **bos addkey(8dfs)**, **bos genkey(8dfs)**, **bos lskeys(8dfs)**, **bos rmkey(8dfs)**, **keytab(8dce)**.

Files: **v5srvtab(5sec)**.

bos genkey

Purpose

bos genkey – Generates a random key and adds it to a keytab file

Synopsis

bos genkey -server *machine* **-kvno** *+_or_version_number* [**-principal** *name*] [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*

Names the server machine whose keytab file is to have a new key added to it. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-kvno *+_or_version_number*

Defines the key version number of the new key. The version number must be one of the following:

- An integer in the range 1 to 255. The command uses the specified integer as the version number of the new key. The integer must be unique for the principal indicated by **-principal** in the keytab file on the server machine specified by **-server**.
- **+** or **0** (zero). The command chooses an integer to serve as the version number of the new key. The integer it chooses is unique for the principal indicated by **-principal** in the Registry Database. However, it may not be unique for the indicated principal in the keytab file on the specified machine, in which case it replaces the key currently associated with the principal/version number pair in the keytab file.

The new key and its version number always replace the key and version number currently stored in the Registry Database for the indicated principal.

-principal*name*
Provides the principal name with which the key is to be associated. The default is the DFS principal name of the machine specified by **-server**.

-noauth Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos genkey** command associates a new server encryption key with the principal name indicated by **-principal** in the **/krb5/v5srvtab** keytab file on the server machine specified by **-server** and in the Registry Database. The command generates a random key and assigns it the version number indicated by **-kvno**. The issuer can either specify a version number or have the command choose one that is unique for the indicated principal in the Registry Database. The server encryption key and version number for the specified principal are automatically updated both in the keytab file on the specified server machine and in the Registry Database.

The **bos addkey** command can also be used to add a key to a keytab file with or without updating the Registry Database. However, it is less secure because the issuer must specify a string to be converted into the server encryption key. The keytab file must already exist before the **bos genkey** or **bos addkey** command can be used to add a key to it. (Keytab files are created with the **dcecp keytab create** command.)

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**, and the DFS server principal of the machine specified by **-server** must have the permissions necessary to alter entries in the Registry Database.

Output

If the packet privacy protection level is not available to you, the command displays the following message reporting that the BOS Server is using the packet integrity protection level instead:

```
Data encryption unsupported by RPC. Continuing without it.
```

Examples

The following command generates a new server encryption key with key version number **14** and adds it to the keytab file on **fs1**. Because **-principal** is omitted, the key is associated with the DFS principal name of **fs1** (the machine specified with **-server**). The Registry Database is updated automatically.

```
$ bos genkey /.../abc.com/hosts/fs1 14
```

Related Information

Commands: **bos addkey(8dfs)**, **bos gckeys(8dfs)**, **bos lskeys(8dfs)**, **bos rmkey(8dfs)**, **keytab(8dce)**.

Files: **v5srvtab(5sec)**.

bos getdates

Purpose

bos getdates – Lists time stamps on versions of binary files

Synopsis

bos getdates -server *machine* **-file** *binary_file…* [**-dir** *alternate_dest*] [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine that houses the binary files whose time stamps are to be displayed. The BOS Server on this machine executes the command. Specify the machine's DCE pathname, its host name, or its IP address.

-file *binary_file*
: Names the current version of each binary file whose time stamps are to be displayed. The time stamps on the current, **.BAK**, and **.OLD** versions of each file are displayed. All specified files must reside in the same directory (*dcelocal*/**bin**, by default, or an alternate directory specified with the **-dir** option). Specify only filenames; if a pathname is provided for a file, the command ignores all but the final element.

-dir *alternate_dest*
: Provides the pathname of the directory in which all specified files reside. Omit this option if the files reside in the default directory, *dcelocal* / **bin**; otherwise, provide a full or relative pathname. Relative pathnames (pathnames that do not begin with a slash) are interpreted relative to the *dcelocal* directory on the machine specified by **-server**.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a

machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos getdates** command displays the time stamps for the current, **.BAK**, and **.OLD** versions of each binary file whose current version is specified with the **-file** option. The time stamps record when the files were installed. The command displays a message for any version of a specified file that does not exist. Use the **-server** option to specify the name of the server machine on which the files reside. The **-dir** option can be used to specify the name of the directory in which the files reside if it is different from *dcelocal*/**bin**.

The BOS Server automatically creates **.BAK** and **.OLD** versions when new binaries are installed with **bos install**. Use the **bos uninstall** command to replace the current version with its next-oldest version (**.BAK** or, if the **.BAK** version does not exist, **.OLD**) or to remove all versions of a binary file. Use the **bos prune** command to remove **.BAK** and **.OLD** files from the *dcelocal*/**bin** directory. (This command can also be used to remove core files from the *dcelocal* **/var/dfs/adm** directory.)

Privilege Required

No privileges are required.

Output

For each file specified with the **-file** option, the output reports the time stamps on the current, **.BAK**, and **.OLD** versions. The output displays a message to indicate any version that does not exist.

Examples

The following command displays the time stamps on the three versions of the **flserver** binary file stored in the default directory on the server machine named **fs2**:

```
$ bos getdates /.../abc.com/hosts/fs2 flserver
```

Related Information

Command: **bos install(8dfs)**, **bos prune(8dfs)**, **bos uninstall(8dfs)**.

bos getlog

Purpose

bos getlog – Examines the log file for a server process

Synopsis

bos getlog -server *machine* **-file** *log_file* [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine from which to retrieve the log file. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-file *log_file*
: Names the log file to display. If a simple filename is provided, with no slashes, the file is assumed to reside in *dcelocal* **/var/dfs/adm**; the standard choices from that directory are **BakLog**, **BosLog**, **DfsgwLog**, **FlLog**, **FtLog**, **RepLog**, and **UpLog**.

  Pathnames are interpreted relative to *dcelocal* **/var/dfs/adm**; absolute pathnames are also allowed. In cases where a / (slash) appears in the specified filename, the issuer's username must appear in the **admin.bos** file on the machine specified by the **-server** option.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If the filename specified by **-file** contains a / (slash), the command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database).

You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos getlog** command displays the contents of the log file specified by **-file** that resides on the machine specified by **-server**. It can be used to view any of the following log files:

BakLog Generated by the Backup Server process on each Backup Database machine

BosLog Generated by the BOS Server process on each server machine

DfsgwLog Generated by the Gateway Server process on each Gateway Server machine

FlLog Generated by the Fileset Location Server process on each Fileset Database machine

FtLog Generated by the Fileset Server process on each File Server machine

RepLog Generated by the Replication Server process on each server machine

UpLog Generated by the **upserver** process on each server machine running the server portion of the Update Server

By default, the command looks in the *dcelocal* **/var/dfs/adm** directory for the log file it is to display. It is not necessary to specify the full pathname of a log file if it resides in the default directory. However, if the file resides elsewhere, the full pathname of the log file must be provided. (The command can also be used to view the **.old** version of a log file created by the associated server process.)

Privilege Required

No privilege is required if the filename specified by **-file** does not contain a / (slash). If the name contains a / (slash), the issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples

The following example displays the contents of the **BosLog** file located in the default directory (*dcelocal* **/var/dfs/adm**) on the server machine named **fs1**:

```
$ bos getl /.../abc.com/hosts/fs1 BosLog
```

Related Information

Files: **BakLog(4dfs)**, **BosLog(4dfs)**, **DfsgwLog(4dfs)**, **FlLog(4dfs)**, **FtLog(4dfs)**, **RepLog(4dfs)**, **UpLog(4dfs)**.

bos getrestart

Purpose

bos getrestart – Lists automatic restart times for server processes

Synopsis

bos getrestart -server *machine* [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine on which to check the restart times. The BOS Server on this machine executes the command. Specify the machine's DCE pathname, its host name, or its IP address.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bos getrestart** command displays the following two restart times from the **BosConfig** file on the server machine specified by the **-server** option:

- The general restart time, which is the time each week when the BOS Server process automatically restarts itself and all processes that have the status flag **Run** in the **BosConfig** file

- The new binary restart time, which is the time each day when the BOS Server automatically restarts any process executed from a binary file in the *dcelocal*/**bin** directory whose time stamp is later than the last restart time for the process

Either of these times can be daily (consist only of a time) or weekly (consist of a day and time). By default, the general restart time is once a week, while the new binary restart time occurs once a day. Both restart times are set with the **bos setrestart** command.

Privilege Required

No privileges are required.

Output

The output consists of the following two lines:

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

where *machine* indicates the name of the server machine whose restart times are displayed, and possible values for *time* include the following:

never Indicates that the BOS Server never performs that type of restart

A specified day and time
: Indicates that the BOS Server performs that type of restart once per week

A specified time
: Indicates that the BOS Server performs that type of restart once per day

Examples

The following command displays the restart times for the server machine **fs2**:

```
$ bos getr /.../abc.com/hosts/fs2
```

```
Server fs2 restarts at sun 4:00 am
Server fs2 restarts for new binaries at 2:15 am
```

Related Information

Commands: **bos setrestart(8dfs)**.

Files: **BosConfig(4dfs)**.

bos help

Purpose

bos help – Shows syntax of specified **bos** commands or lists functional descriptions of all **bos** commands

Synopsis

bos help [**-topic** *string*]… [**-help**]

Options

-topic *string* Specifies each command whose syntax is to be displayed. Provide only the second part of the command name (for example, **status**, not **bos status**). If this option is omitted, the output provides a short description of all **bos** commands.

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

Description

The **bos help** command displays the first line (name and short description) of the online help entry for every **bos** command if **-topic** is not provided. For each command name specified with **-topic**, the output lists the entire help entry.

Use the **bos apropos** command to show each help entry containing a specified string.

Privilege Required

No privileges are required.

Output

The online help entry for each **bos** command consists of the following two lines:

- The first line names the command and briefly describes its function.

- The second line, which begins with **Usage:**, lists the command options in the prescribed order.

Examples

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

```
$ bos help status


bos status: show server process status
Usage: bos status -server <machine> [-process <server_process>...]
[-long] [{-noauth | -localauth}] [-help]
```

Related Information

Commands: **bos apropos(8dfs)**.

bos install

Purpose

bos install – Installs new versions of binary files

Synopsis

bos install **-server** *machine* **-file** *binary_file…* [**-dir** *alternate_dest*] [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine on which the new binary files are to be installed. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-file *binary_file*
: Specifies the pathname of each binary file to be installed on the machine specified by the **-server** option. For each file, specify either the full pathname or a relative pathname (one that does not begin with a slash); relative pathnames are interpreted relative to the current working directory. The name of each file remains the same when it is installed on the specified machine; the command automatically preserves an existing file of the same name as a file that is installed.

-dir*alternate_dest*
: Provides the pathname of the directory on the machine specified by the **-server** option in which all specified files are to be installed. Omit this option to install the files in the default directory, *dcelocal* **/bin**; otherwise, provide a full or relative pathname. Relative pathnames are interpreted relative to the *dcelocal* directory on the machine specified by **-server**.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and

DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos install** command installs each binary file specified with the **-file** option on the server machine specified with the **-server** option. The **-file** option provides the pathname of each file to be installed on the specified machine. By default, the command installs the files in the *dcelocal* **/bin** directory on the specified machine; use the **-dir** option to indicate a different installation directory on the specified machine.

The command does not change the names of files when it installs them. To preserve the current version of a binary file that has the same name as a file being installed, the command adds a **.BAK** extension to the name of the existing file. If there is a **.BAK** version at least 7 days old, the command adds a **.OLD** extension to its name and it replaces the current **.OLD** version (if one exists). If there is a **.BAK** version less than 7 days old, it is overwritten when the current version receives a **.BAK** extension. If there is no **.OLD** version, the current **.BAK** version becomes the **.OLD** version automatically, regardless of its age.

The command is typically used to install new versions of binary files on Binary Distribution machines. The machine specified with the **-server** option should be the Binary Distribution machine for its CPU/operating system type. If it is not, newly installed binary files are overwritten the next time the **upclient** process on the specified machine copies new (or different) versions of binary files via the **upserver** process on the Binary Distribution machine of its CPU/operating system type. (Note that the Update Server propagates binary files from Binary Distribution machines, but the BOS Server installs files when the **bos install** command is issued; by default, it takes the Update Server 5 minutes to propagate binary files from a Binary Distribution machine.)

To make the machine specified by **-server** immediately start using new binary files for server processes controlled by the BOS Server, issue the **bos restart** command.

Otherwise, new binaries are not used until the BOS Server restarts the affected processes at the new binary restart time specified in the *dcelocal* **/var/dfs/BosConfig** file. Use the **bos getrestart** and **bos setrestart** commands to inspect and set the new binary restart time. (The information in this paragraph applies *only* to affected processes already under the control of the BOS Server.)

The **bos install** command installs all files with the UNIX mode bits set to **755** (**rwxr–xr–x**), regardless of the mode bits associated with a version of the file that currently exists in the destination directory. These permissions are subject to the **umask** associated with the BOS Server on the machine on which the files are installed (because the BOS Server on the specified machine actually executes the command). The mode bits associated with the current version of the file are preserved when it becomes the **.BAK** version, as are those of the **.BAK** version when it becomes the **.OLD** version. (The command does not preserve the access control list, or ACL, permissions of a file installed from a DCE LFS fileset, nor does it directly manipulate the ACL permissions of a file installed into a DCE LFS fileset.)

Use the **bos uninstall** command to replace the current version of a binary file with the next-oldest version of the file: the **.BAK** version, if it exists; otherwise, the **.OLD** version. If both the **.BAK** and **.OLD** versions exist, the **.OLD** version replaces the **.BAK** version when the latter becomes the current version. Use the **-all** option with the **bos uninstall** command to remove all versions of a file; use the **bos prune** command to remove **.BAK** and **.OLD** files from the *dcelocal* **/bin** directory. (This command can also be used to remove core files from the *dcelocal* **/var/dfs/adm** directory.) Use the **bos getdates** command to check the time stamps on binary files.

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**.

Related Information

Commands: **bos create(8dfs)**, **bos getdates(8dfs)**, **bos getrestart(8dfs)**, **bos prune(8dfs)**, **bos restart(8dfs)**, **bos setrestart(8dfs)**, **bos uninstall(8dfs)**, **upclient(8dfs)**, **upserver(8dfs)**.

Files: **BosConfig(4dfs)**.

bos lsadmin

Purpose

bos lsadmin – Lists the users, groups, and servers from an administrative list

Synopsis

bos lsadmin -server *machine* **-adminlist** *filename* [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine that houses the administrative list whose principals and groups are to be displayed. The BOS Server on this machine executes the command. Specify the machine's DCE pathname, its host name, or its IP address.

-adminlist *filename*
: Names the administrative list whose principals and groups are to be displayed. The complete pathname is unnecessary if the list is stored in the default configuration directory (*dcelocal*/**var**/**dfs**).

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bos lsadmin** command lists the principal names of users and server machines and the names of groups found in the administrative list specified by the **-adminlist** option on the server machine specified by the **-server** option. Principals whose names are specified in the administrative list or that are members of groups specified in the list can issue administrative commands for the DFS server process associated with the list.

The default path for the administrative lists is the configuration directory (*dcelocal*/**var**/**dfs**). If the specified list is stored in the default directory, only the specific filename is required. If the specified list is stored elsewhere, the pathname to the file that was used when the associated server process was started is required.

Use the **bos addadmin** command to add principals and groups to an administrative list. Use the **bos rmadmin** command to remove principals and groups from an administrative list.

Privilege Required

No privileges are required.

Output

The command displays the output

```
Admin Users are:
```

followed by the principal name of each user and machine and the name of each group contained in the administrative list. Names from the local cell are displayed in an abbreviated form (for example, *username* for **/.../***cellname /username*); names from foreign cells are displayed in full. Each name is preceded by one of the following strings:

`user:` Precedes the principal name of each user or machine from the local cell

`foreign_user:`
Precedes the principal name of each user or machine from a foreign cell

`group:` Precedes the name of each group from the local cell

`foreign_group:`
Precedes the name of each group from a foreign cell

Examples

The following command lists the members of the **admin.bos** file on the server machine named **fs1**. The administrative list contains two users, a server machine, and two groups, all of which are from the local cell.

$ bos lsa -server /.../abc.com/hosts/fs1 -adminlist admin.bos

```
Admin Users are: user: jones, user: smith,
user: hosts/fs1/self, group: dfs-admin, group: fs1-admin
```

Related Information

Commands: **bos addadmin(8dfs)**, **bos rmadmin(8dfs)**.

Files: **admin.bak(4dfs)**, **admin.bos(4dfs)**, **admin.fl(4dfs)**, **admin.ft(4dfs)**, **admin.up(4dfs)**.

bos lscell

Purpose

bos lscell – Lists the cell in which the BOS Server is running

Synopsis

bos lscell -server *machine* [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine on which the BOS Server whose cell is to be listed is running. Specify the machine's DCE pathname, its host name, or its IP address.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bos lscell** command reports the name of the cell in which the BOS Server on the machine specified with the **-server** option is running. The command extracts the information from the local configuration file, *dcelocal*/**dce_cf.db**, on the specified machine. If the command fails after being issued from the machine specified by **-server** (if **-server** is the local machine), the failure may indicate that the local **dce_cf.db** file is corrupted; use the **cat** or **more** command (or a similar command appropriate to

your operating system) to display the contents of the file, and ensure that it is not corrupted.

Privilege Required

No privileges are required.

Output

The **bos lscell** command displays the following line reporting the name of the cell in which the BOS Server is running:

```
Cell name is cellname
```

Examples

The following command displays the name of the cell in which the BOS Server on the machine named **fs1** is running:

```
$ bos lscell /.../abc.com/hosts/fs1

Cell name is abc.com
```

bos lskeys

Purpose

bos lskeys – Displays server encryption key information from a keytab file

Synopsis

bos lskeys - server *machine* [**- principal** *name*] [{**-noauth** | **-localauth** }] [**- help**]

Options

-server *machine*
: Names the server machine whose keytab file is to have keys listed. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-principal*name*
: Provides the principal name for which associated keys are to be listed. The default is the DFS principal name of the machine specified by **-server**.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **- server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bos lskeys** command formats and displays information about server encryption keys kept in the **/krb5/v5srvtab** keytab file on the server machine specified by **-server**. It displays information for keys associated with the principal name indicated by **-principal**; the DFS principal name of the server machine specified with **-server** is used by default.

DFS authorization checking must be disabled on the machine specified with **-server** to display the string of octal numbers that compose the key. (Use the **bos setauth** command to disable DFS authorization checking.) Disabling DFS authorization checking is required for two reasons. First, it implies that only someone authorized to issue the **bos setauth** command or someone with **root** access to **- server**'s local disk (presumably a system administrator) is able to see actual encryption keys. Second, it makes it clear that the system is in a compromised state of security while server encryption keys are being examined. (Both turning off DFS authorization checking and displaying keys on a screen are serious security risks.)

If DFS authorization checking is enabled on **-server** (the normal case), a **checksum** appears in place of the octal numbers. A checksum is a decimal number derived by encrypting a constant with each key.

Privilege Required

If DFS authorization checking is enabled, you must be listed in the **admin.bos** file on the machine specified by **-server**; checksums are displayed instead of the actual keys. Because DFS authorization checking must be disabled with the **bos setauth** command before the actual keys (rather than just checksums) can be displayed, no privilege is required to see the keys. However, you must be listed in the **admin.bos** file on a machine to use the **bos setauth** command to disable DFS authorization checking on it.

Output

The **bos lskeys** command displays one line for each server encryption key associated with **-principal** in the keytab file on the machine specified by **-server**. Each key is identified by its key version number. If DFS authorization checking is enabled on the machine, a checksum is displayed with each version number; if checking is disabled, the octal numbers that constitute the key are displayed.

A line specifying when the key in the Registry Database (at the Registry Server) was last changed follows the list of keys or checksums. The words **All done** indicate the end of the output.

If the packet privacy protection level is not available to you, the command displays the following message reporting that the BOS Server is using the packet integrity protection level instead:

```
Data encryption unsupported by RPC. Continuing without it.
```

Examples

The following command shows the checksums for the keys associated with the principal name of **fs3** in the keytab file on that machine. The checksums appear instead of the actual keys because DFS authorization checking is *not* disabled.

```
$ bos lsk /.../abc.com/hosts/fs3
```

```
key 1 has cksum 972037177
key 3 has cksum 282517022
key 4 has cksum 260617746
Keys last changed (at the registry server) on Thu Jun 6 11:24:46 1991.
All done.
```

The following command lists the keys associated with **fs3** after DFS authorization checking is disabled with the **bos setauth** command:

```
$ bos setauth /.../abc.com/hosts/fs3 off
```

```
$ bos lsk /.../abc.com/hosts/fs3
```

```
key 1 is '\040\205\211\241\345\002\023\211'
key 2 is '\343\315\307\227\255\320\135\244'
key 3 is '\310\310\255\253\265\236\261\211'
Keys last changed (at the registry server) on Thu Jun 6 11:24:46 1991.
All done.
```

Related Information

Commands: **bos addkey(8dfs)**, **bos gckeys(8dfs)**, **bos genkey(8dfs)**, **bos rmkey(8dfs)**, **bos setauth(8dfs)**, **keytab(8dce)**.

Files: **v5srvtab(5sec)**.

bos prune

Purpose

bos prune – Removes old binary and core files from *dcelocal*/**bin** and *dcelocal*/**var/dfs/adm**

Synopsis

bos prune -server *machine* [**-bak**][**-old**][**-core**][**-all**][{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine from which to remove the indicated files. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-bak
: Removes all files with a **.BAK** extension from *dcelocal* **/bin**. Use this option with **-old**, **-core**, or both, or use **-all**.

-old
: Removes all files with an .**OLD** extension from *dcelocal*/**bin**. Use this option with **-bak**, **-core**, or both, or use **-all**.

-core
: Removes all core files from *dcelocal*/**var/dfs/adm**. Use this option with **-bak**, **-old**, or both, or use **-all**.

-all
: Removes all **.BAK** and **.OLD** files from *dcelocal*/**bin** and all core files from *dcelocal*/**var/dfs/adm**. Use this option or use some combination of **-bak**, **-old**, and **-core**.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this

option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos prune** command removes obsolete versions of binary and core files from the *dcelocal*/**bin** and *dcelocal*/**var/dfs/adm** directories on the server machine specified with the **-server** option. Binary files should only need to be removed from the Binary Distribution machine for a CPU/operating system type; core files may need to be removed from any server machine. Specify the files to be removed with the command's other options as follows:

- Use the **-bak** option to remove all **.BAK** files from *dcelocal*/**bin**.
- Use the **-old** option to remove all **.OLD** files from *dcelocal*/**bin**.
- Use the **-core** option to remove all core files from *dcelocal*/**var/dfs/adm**.
- Use the **-all** option to remove all three types of files.

The **-bak**, **-old**, and **-core** options can be combined to remove different types of files with the same command. The **-all** option can also be used with any of the three options, but using the **-all** option alone is sufficient to remove all three types of files.

Binary files with **.BAK** and **.OLD** extensions are created when new versions of binary files are installed with the **bos install** command. Core files are created when a process that the BOS Server is monitoring goes down.

Use the **bos uninstall** command to replace the current version of a binary file with its next-oldest version (**.BAK** or, if the **.BAK** version does not exist, **.OLD**) or to remove all versions of a binary file. Use the **bos getdates** command to determine the time stamps on binary files.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Related Information

Commands: **bos getdates(8dfs)**, **bos install(8dfs)**, **bos uninstall(8dfs)**.

bos restart

Purpose

bos restart – Restarts processes on a server machine

Synopsis

bos restart -server *machine* [{**-bosserver** | **-process** *server_process*}] [] [**-help**]

Options

-server*machine*
: Names the server machine on which to stop and restart indicated processes. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-bosserver
: Indicates that all processes, including the current BOS Server, are to stop running. A new BOS Server immediately starts; it then restarts all processes with the status flag **Run** in the *dcelocal* **/var/dfs/BosConfig** file.

  Provide this option or provide the **-process** option. Omit both options to stop all processes except the BOS Server; those with the status flag **Run** in the **BosConfig** file are immediately restarted.

-process *server_process*
: Specifies each process to be stopped and immediately restarted. The BOS Server stops all specified processes that are currently running; it then restarts all of the specified processes, regardless of their status flags in the **BosConfig** file. Refer to a process by the name assigned with the **-process** option of the **bos create** command (this name appears in the output from the **bos status** command). *Do not include* **bosserver** *in the list of processes*; use the **-bosserver** option instead.

  Provide this option or provide the **-bosserver** option. Omit both options to stop all processes except the BOS Server; those with the status flag **Run** in the **BosConfig** file are immediately restarted.

-noauth Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos restart** command instructs the BOS Server running on the server machine specified by **-server** to stop all indicated processes on the machine. The BOS Server then immediately restarts some or all of the processes, depending on the options included with the command. The processes to be stopped and possibly restarted are specified with the following options:

- The **-bosserver** option causes the BOS Server to stop all processes, including itself. A new BOS Server immediately starts; it then restarts all processes with the status flag **Run** in the **BosConfig** file.
- The **-process** option causes the BOS Server to stop and immediately restart all specified processes, regardless of their status flags in the **BosConfig** file. All restarted processes with the status flag **NotRun** in the **BosConfig** file have the status **temporarily enabled** in the output of the **bos status** command.
- The absence of both the **-bosserver** and **-process** options causes the BOS Server to stop all processes except itself. The BOS Server then immediately restarts all processes with the status flag **Run** in the **BosConfig** file.

This command can be used to stop only those processes the BOS Server controls. Also, it does *not* change the status flag for a process in the **BosConfig** file.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples

The following command instructs the BOS Server on */.../abc.com/hosts/fs3* to stop all processes, including itself. A new BOS Server immediately starts, after which it restarts all processes with the status flag **Run** in the **BosConfig** file.

```
$ bos restart /.../abc.com/hosts/fs3 -bos
```

The following command instructs the BOS Server on */.../abc.com/hosts/fs5* to stop all processes except itself. The BOS Server then restarts all processes with the status flag **Run** in the **BosConfig** file.

```
$ bos res /.../abc.com/hosts/fs5
```

Related Information

Commands: **bos create(8dfs)**, **bos status(8dfs)**.

Files: **BosConfig(4dfs)**.

bos rmadmin

Purpose

bos rmadmin – Removes a user, group, or server from an administrative list

Synopsis

bos rmadmin -server *machine* **-adminlist** *filename* [**-principal** *name*…] [**-group** *name*…] [**-removelist**][{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*

Names the server machine that houses the administrative list from which principals, groups, or both are to be removed. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-adminlist *filename*

Names the administrative list from which principals, groups, or both are to be removed. The complete pathname is unnecessary if the list is stored in the default configuration directory (*dcelocal*/**var/dfs**).

-principal *name*

Specifies the principal name of each user or server machine to be removed from the administrative list. A user from the local cell can be specified by a full or abbreviated principal name (for example, **/.../***cellname*/*username* or just *username*); a user from a foreign cell can be specified only by a full principal name. A server machine from the local cell can be specified by a full or abbreviated principal name (for example, **/.../***cellname* **/hosts/***hostname***/self** or just **hosts/***hostname* **/self**); a server machine from a foreign cell can be specified only by a full principal name.

-group *name*
Specifies the name of each group to be removed from the administrative list. A group from the local cell can be specified by a full or abbreviated group name (**/.../***cellname***/** *group_name* or just *group_name*); a group from a foreign cell can be specified only by a full group name.

-removelist Specifies that the file indicated with **-adminlist** is to be removed if it is empty either when the command is issued or after any principals or groups specified with the command are removed. This option has no effect if the specified file is not empty when the command is issued or after any indicated principals or groups are removed.

-noauth Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos rmadmin** command removes the specified users, groups, and servers from the administrative list specified by the **-adminlist** option on the server machine specified by the **-server** option. The principal (login) names of users and the principal names of server machines to be removed from the administrative list are specified with the **-principal** option; the names of groups to be removed from the list are specified with the **-group** option. Principals removed from the administrative list either directly (with the **-principal** option) or indirectly (as members of groups indicated with the **-group** option) can no longer issue administrative commands for the DFS server process associated with the list.

The default path for administrative lists is the configuration directory (*dcelocal***/var/dfs**). If the specified list is stored in the default directory, only the specific filename

is required. If the specified list is stored elsewhere, the pathname to the file that was used when the associated server process was started is required.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples

The following command removes the former administrative users *smith* and **jones** from the **admin.bos** file on **fs1**:

```
$ bos rmadmin -server /.../abc.com/hosts/fs1 -adminlist admin.bos \
-principal smith jones
```

Related Information

Commands: **bos addadmin(8dfs)**, **bos lsadmin(8dfs)**.

Files: **admin.bak(4dfs)**, **admin.bos(4dfs)**, **admin.fl(4dfs)**, **admin.ft(4dfs)**, **admin.up(4dfs)**.

bos rmkey

Purpose

bos rmkey – Removes server encryption keys from a keytab file

Synopsis

bos rmkey -server *machine* **-kvno** *version_number*… [**-principal** *name*] [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine whose keytab file is to have keys removed from it. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-kvno *version_number*
: Specifies the key version number of each key to be removed from the keytab file. The command removes each key that is associated with a specified key version number and the principal indicated by **-principal**. Each version number must be an integer in the range 1 to 255.

-principal*name*
: Provides the principal name associated with the keys to be removed from the keytab file. The default is the DFS principal name of the machine specified by **-server**.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database).

You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos rmkey** command removes server encryption keys from the **/krb5/v5srvtab** keytab file on the server machine specified by **-server**. It removes each key associated with a key version number indicated by **-kvno** and the principal indicated by **-principal**. The command has no effect on the Registry Database. Once a key is removed from the keytab file, it can no longer be used to establish communication between clients and the server to which it applied.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Output

If the packet privacy protection level is not available to you, the command displays the following message reporting that the BOS Server is using the packet integrity protection level instead:

```
Data encryption unsupported by RPC. Continuing without it.
```

Examples

The following command removes two keys from the keytab file on **fs1**: the keys with key version numbers **5** and **6** that are associated with the DFS principal name of **fs1**.

```
$ bos rmk /.../abc.com/hosts/fs1 -kvno 5 6
```

Related Information

Commands: **bos addkey(8dfs)**, **bos gckeys(8dfs)**, **bos genkey(8dfs)**, **bos lskeys(8dfs)**, **keytab(8dce)**.

Files: **v5srvtab(5sec)**.

bos setauth

Purpose

bos setauth – Enables or disables DFS authorization checking for all DFS server processes on a machine

Synopsis

bos setauth -server *machine* **-authchecking** {on | off} [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine on which the status of DFS authorization checking is to change. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-authchecking
: Determines whether or not server processes on the machine check for DFS authorization. A value of **on** enables DFS authorization checking; a value of **off** disables it.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. (The option can be used only when enabling authorization checking.) If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos setauth** command enables or disables DFS authorization checking on the server machine specified by the **-server** option. If DFS authorization checking is enabled on a server machine, all DFS server processes running on the machine check that the issuer of a command is correctly authorized (is included in the necessary administrative lists) to execute the command. If DFS authorization checking is disabled on a server machine, the DFS server processes on the machine perform any action for any user, even the unprivileged user **nobody**.

By default, DFS authorization checking is enabled on every server machine. Disabling it on a server machine is a serious security risk. It is typically disabled for the briefest possible time and only in the following situations:

- During initial DFS installation
- If the Security Service is unavailable
- During server encryption key emergencies
- To view the actual keys stored in a keytab file

To indicate to all DFS server processes (including itself) that DFS authorization checking is disabled on a server machine, the BOS Server creates the zero-length file *dcelocal*/**var/dfs/NoAuth** on the local disk of the machine. All DFS server processes, including the BOS Server, check for the presence of this file when they are requested to perform an operation; they do not check for the necessary administrative privilege for a requested operation when the file is present. To indicate that DFS authorization checking is enabled, the BOS Server removes the file.

Enter this command with the **-authchecking** option and an argument of **off** to disable DFS authorization checking on a server machine. (DFS authorization checking can also be disabled by including the **-noauth** option with the **bosserver** command used to start the BOS Server.) Issue the command with the **-authchecking** option and an argument of **on** to enable DFS authorization checking on a server machine. It is not necessary to restart currently running server processes when you change the state of DFS authorization checking; server processes immediately obey the current state of DFS authorization checking and act accordingly.

The **bos status** command can be used to determine whether DFS authorization checking is enabled or disabled on a server machine. The command displays the following message if DFS authorization checking is disabled on a machine. (It does not display the message if DFS authorization checking is enabled.)

```
Bosserver reports machine is not checking authorization.
```

The **-noauth** option available with many **bos** and **fts** commands is used when authentication information is unnecessary or unavailable. Use the **-noauth** option if DFS authorization checking is disabled on a server machine on which administrative privilege is required or if the Security Service is unavailable.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server** to disable DFS authorization checking on that machine. (No privilege is required to enable DFS authorization checking if it is currently disabled.)

Cautions

Always use the **bos setauth** command to create the *dcelocal*/**var/dfs/NoAuth** file. Do not create the file directly except when explicitly told to do so by instructions for dealing with emergencies (such as emergencies involving server encryption keys). Creating the file directly requires logging into the local operating system of a machine as **root** and using the **touch** command (or its equivalent).

Examples

The following command disables DFS authorization checking for all DFS server processes on the server machine named **fs7**:

```
$ bos seta /.../abc.com/hosts/fs7 off
```

Related Information

Commands: **bos status**, **bosserver(8dfs)**.

Files: **NoAuth(4dfs)**.

bos setrestart

Purpose

bos setrestart – Sets the date and time at which the BOS Server restarts all processes or only those with new binaries

Synopsis

bos setrestart -server *machine* {**-general** *time* | **-newbinary** *time*} [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*

Specifies the server machine for which restart times are to be set. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-general*time*

Sets the time at which the BOS Server restarts first itself and then each server process that has an entry in the **BosConfig** file with a status flag of **Run**. Specify a day and time to perform the restart weekly at that time; specify a time to perform the restart daily at that time. Day and time specifications have the following format:

[*day*] *hh*:*mm*

Enter the name of the day in all lowercase letters, giving either the whole name or the first three letters as an abbreviation (for example, **sunday** or **sun**). Specify the time of day by separating the hours from the minutes with a **:** (colon). Use 24-hour time (for example, **14:30**), or use 1:00 through 12:00 with **am** or **pm** (for example, **"2:30 pm"**). As shown in the example, enclose the entry in " " (double quotes) if it contains a space.

You can use either of two additional specifications instead of a time or a day and time:

never Directs the BOS Server never to perform the indicated type of restart

now Directs the BOS Server to use the day and time at which the command is issued (for example, Sunday at 2:00 a.m.) as the day and time for the indicated type of restart

If this option is never used to set the general restart time, the default general restart time is Sunday at 4:00 a.m. If you change the general restart time, the recommended frequency for this type of restart is once per week.

-newbinary *time*
Sets the time at which the BOS Server restarts any server process whose binary file was installed in *dcelocal*/**bin** after the current instance of the process started running. The recommended frequency for this type of restart is once per day, so it is standard to specify only a time of day. Use the conventions described for the **-general** option to express the time of day. The remarks for the **-general** option concerning **never** and **now** also apply to this option.

If this option is never used to set the binary checking time, the default binary checking time is 5:00 a.m.

-noauth Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos setrestart** command sets the times at which the BOS Server running on the server machine specified by the **-server** option is to perform one of two types of restarts. The command records the time settings in the **BosConfig** file. The two types of restart times are

- The time each week when the BOS Server restarts itself and any processes marked with the status flag **Run** in the **BosConfig** file. This is equivalent to executing **bos restart** with the **-bosserver** option. The default setting is 4:00 a.m. each Sunday morning.
- The time each day when the BOS Server restarts any process currently running for which the binary file in *dcelocal*/**bin** was modified since the process was last started (or restarted). The default setting is 5:00 a.m. each day.

To change both times, you must issue the command twice.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by the **-server** option.

Cautions

Restarting processes makes them unavailable for a period of time. It is advisable to set the restarts for times of typically low usage to inconvenience as few users as possible.

If the specified time is within one hour of the current time, the BOS Server does not restart the processes until that time on the next day.

Examples

The following command defines a general restart time in the **BosConfig** file on **fs4** that causes all processes on that machine to stop and restart each Saturday morning at 3:30 a.m.:

```
$ bos setr -s /.../abc.com/hosts/fs4 -gen "sat 3:30"
```

The following command defines a new binary restart time in the **BosConfig** file on **fs6**, instructing the BOS Server on that machine to check for new binary files each evening at 11:45 p.m. and restart any processes for which it finds a new file at that time:

```
$ bos setr -s /.../abc.com/hosts/fs6 -new 23:45
```

Related Information

Commands: **bos getrestart(8dfs)**, **bos restart(8dfs)**.

Files: **BosConfig(4dfs)**.

bos shutdown

Purpose

bos shutdown – Stops processes without changing their status flags in the **BosConfig** file

Synopsis

bos shutdown -server *machine* [**-process** *server_process*]… [**-wait**][{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine on which the indicated processes are to be stopped. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-process *server_process*
: Specifies each process to be stopped. If this option is omitted, the BOS Server stops all server processes other than itself on the server machine. Refer to a process by the name assigned with the **-process** option of the **bos create** command; this name appears in the output of the **bos status** command.

-wait
: Indicates that the command shell prompt is not to return until the shutdown is complete (until all processes actually stop running). If this option is omitted, the prompt returns almost immediately, even if all of the processes are not yet stopped.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this

option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos shutdown** command instructs the BOS Server running on the server machine specified by **-server** to stop either all processes (except itself) running on the machine *or* only the processes specified by **-process**. The command does not change a process's status flag in the **BosConfig** file, only in the BOS Server's memory state.

Processes stopped with this command do not run again until they are started using the **bos start**, **bos startup**, or **bos restart** commands, or until the BOS Server itself restarts.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples

The following command instructs the BOS Server running on **fs3** to stop running all processes except itself:

```
$ bos shutdown -s /.../abc.com/hosts/fs3
```

Related Information

Commands: **bos create(8dfs)**, **bos status(8dfs)**.

bos start

Purpose

bos start – Starts processes after setting their status flags to **Run** in the **BosConfig** file and in memory

Synopsis

bos start -server *machine* **-process** *server_process*… [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine whose processes are to be started. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-process*server_process*
: Specifies each process to be started after its status flag in the **BosConfig** file and in memory is set to **Run**. Refer to a process by the name assigned with the **-process** option of the **bos create** command; this name appears in the output from the **bos status** command.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos start** command changes the status flag for each server process specified by **-process** from **NotRun** to **Run** in the **BosConfig** file and in memory on the server machine specified by **-server**. It then starts each specified process running on that machine.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Cautions

If an instance of a process is already running, the only effect is to guarantee that its status flag is set to **Run** in both the **BosConfig** file and memory; it does not start a new instance of the process. Issue the **bos restart** command after this command to start a new instance.

Examples

The following command causes the BOS Server on **fs3** to start the Replication Server (**repserver** process) on that machine by changing its status flags to **Run** in both the **BosConfig** file and memory:

```
$ bos start /.../abc.com/hosts/fs3 repserver
```

Related Information

Commands: **bos create(8dfs)**, **bos restart(8dfs)**, **bos startup(8dfs)**, **bos status(8dfs)**.

Files: **BosConfig(4dfs)**.

bos startup

Purpose

bos startup – Starts processes by changing their status flags to **Run** in memory without changing their status flags in the **BosConfig** file

Synopsis

bos startup -server *machine* [**-process** *server_process*…] [{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine whose processes are to be started. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-process*server_process*
: Specifies each process to be started after its status flag in memory is set to **Run**. Refer to a process by the name assigned with the **-process** option of the **bos create** command; this name appears in the output from the **bos status** command.

  If this option is omitted, all server processes with a status flag of **Run** in the **BosConfig** file that are not running are started after their status flags in memory are set to **Run**.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database).

You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

This command instructs the BOS Server running on the server machine specified by **-server** to start *either* all server processes with a status flag of **Run** in the **BosConfig** file that are not running (if **-process** is omitted) *or* each process specified by **-process**, even if its status flag in the **BosConfig** file is **NotRun**. The status flags of all started processes are changed from **NotRun** to **Run** in memory.

Using **-process** is useful for testing server processes without enabling them permanently. This command does *not* change the status flag for a process in the **BosConfig** file.

Cautions

If an instance of a process is already running, the only effect is to guarantee that its status flag is set to **Run** in memory; it does not start a new instance of the process. Issue the **bos restart** command after this command to start a new instance.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples

The following command causes the BOS Server on **fs3** to start all processes on that machine marked with a status flag of **Run** in the **BosConfig** file that are not currently running. The status flags of all such processes are set to **Run** in memory; their status flags remain set to **Run** in the **BosConfig** file.

```
$ bos startup /.../abc.com/hosts/fs3
```

The following command causes the BOS Server on **fs3** to start the Replication Server (**repserver** process) on that machine by changing its status flag to **Run** in memory. The process's status flag remains unchanged in the **BosConfig** file, regardless of its current setting (**Run** or **NotRun**).

```
$ bos startup /.../abc.com/hosts/fs3 repserver
```

Related Information

Command: **bos create(8dfs)**, **bos restart(8dfs)**, **bos shutdown(8dfs)**, **bos start(8dfs)**, **bos status(8dfs)**, **bos stop(8dfs)**.

Files: **BosConfig(4dfs)**.

bos status

Purpose

bos status – Displays the statuses of server processes on a server machine

Synopsis

bos status -server *machine* [**-process** *server_process*]… [**-long**][{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine about whose processes status information is to be displayed. The BOS Server on this machine executes the command. Specify the the machine's DCE pathname, its host name, or its IP address.

-process *server_process*
: Specifies each process whose status is to be displayed; refer to a process by the name assigned with the **-process** option of the **bos create** command. If this option is omitted, the statuses of all of the processes on the specified server are listed.

-long
: Directs the BOS Server to provide more detailed information about the specified processes.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **bos status** command lists status information about the processes on the server machine specified by the **-server** option. Use the **-process** option to indicate the specific processes about which information is to be displayed, or omit the option to display information about all the processes on the server machine. The command also displays appropriate messages if DFS authorization checking is disabled on the machine or if the machine's *dcelocal* directory or a directory or file beneath it has inappropriate protections.

Use the **-long** option to display more information about each specified process. The additional information can be used to determine the role of a server machine in a domain. (See Part 1 of this manual for instructions on using this command to determine the role of a server machine.)

Privilege Required

No privileges are required.

Output

The command first displays the following line if DFS authorization checking is disabled on the machine. (It does not display the line if DFS authorization checking is enabled.)

```
Bosserver reports machine is not checking authorization.
```

It then displays the following line if the BOS Server finds that the *dcelocal* directory or a directory or file under it on the machine has protections that the BOS Server believes are inappropriate:

```
Bosserver reports inappropriate access on server directories.
```

The message can indicate, for example, that users who should not be able to write to the *dcelocal* directory and its subdirectories have write access. The BOS Server displays the message if the UNIX mode bits on the following objects do not enforce the indicated protections. Provided the mode bits do not violate the specific restrictions

cited in the list, a directory or file can grant more permissions than those shown in the list, but it should not grant fewer.

dcelocal At least **755**, and **other** cannot have write access

dcelocal/**bin** At least **755**, and **other** cannot have write access

dcelocal/**var** At least **755**, and **other** cannot have write access

dcelocal/**var/dfs**
At least **701**, and **other** cannot have write access

dcelocal/**var/dfs/adm**
At least **755**, and **other** cannot have write access

dcelocal/**var/dfs/admin.bos**
At least **600**, and **other** cannot have write or execute access

The BOS Server also displays the message if all of these objects are not owned by **root**. The BOS Server displays the message only as a courtesy to the user. It does nothing to change the protections on these objects, nor does it fail if these protections are violated.

Note: The protections just described are the default protections enforced by the BOS Server. Your vendor can modify the required owner of the indicated directories and the permissions those directories must have. Refer to your vendor's documentation to determine the protections that apply for your version of DFS.

The command then displays a separate entry for each specified process. The first line of an entry shows the current status of the process. The possible statuses for any process include the following:

currently running normally
For a **simple** process, this means it is currently running; for a **cron** process, this means it is scheduled to run.

temporarily enabled
The status flag for the process in the *dcelocal* **/var/dfs/BosConfig** file is **NotRun**, but the process has been enabled with the **bos startup** or **bos restart** command.

temporarily disabled
Either the **bos shutdown** command was used to stop the process, or the BOS Server quit trying to restart the process, in which case the message **stopped for too many errors** also appears.

disabled The status flag for the process in the **BosConfig** file is **NotRun**, and the process has not been enabled.

has core file The process failed or produced a core file at some time. This message can appear with any of the other messages. Core files are stored in *dcelocal* **/var/dfs/adm**. The name of the core file indicates the process that failed (for example, **core.ftserver**).

The output for a **cron** process includes an auxiliary status message that reports when the command is next scheduled to execute.

The command displays the following additional information when the **-long** option is used:

- The process type (**simple** or **cron**).
- How many **proc starts** occurred (**proc starts** occur when the process is started or restarted by the current BOS Server).
- The time of the last **proc start**.
- The exit time and error exit time when the process last failed. This appears only if the process failed while the BOS Server was running. (Provided the BOS Server was running both when the process was started and when it failed, the BOS Server can provide this information for any process that has an entry in the **BosConfig** file.)
- The command and its options used by the BOS Server to start the process.

Examples

The following command displays the statuses of all server processes on the File Server machine named **fs4**:

```
$ bos status /.../abc.com/hosts/fs4
```

```
Instance ftserver, currently running normally.
Instance repserver, currently running normally.
```

If the **-long** option is included with the command, the following additional information is displayed:

```
Instance ftserver, (type is simple) currently running normally.
Process last started at Fri Nov 22 05:36:02 1991 (1 proc starts)
Parameter 1 is 'dcelocal/bin/ftserver'

Instance repserver, (type is simple) currently running normally.
Process last started at Fri Nov 22 05:36:48 1991 (1 proc starts)
Parameter 1 is 'dcelocal/bin/repserver'
```

Related Information

Commands: **bos create(8dfs)**, **bos restart(8dfs)**, **bos shutdown(8dfs)**, **bos start(8dfs)**, **bos startup(8dfs)**, **bos stop(8dfs)**.

Files: **BosConfig(4dfs)**.

bos stop

Purpose

bos stop − Stops processes after changing their status flags in the **BosConfig** file to **NotRun**

Synopsis

bos stop -server *machine* **-process** *server_process*… [**-wait**][{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine on which to stop the processes. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-process*server_process*
: Specifies each process that the BOS Server is to stop. The BOS Server stops a process after setting its status flag in the **BosConfig** file to **NotRun**. Refer to a process by the name assigned with the **-process** option of the **bos create** command; this name appears in the output from the **bos status** command.

-wait
: Indicates that the command shell prompt is not to return until all specified processes actually stop running. If this option is omitted, the prompt returns almost immediately, even if all of the processes are not yet stopped.

-noauth
: Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this

option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos stop** command sets the status flag for each server process specified by **-process** to **NotRun** in the **BosConfig** file on the server machine specified by **-server**; it then stops each process.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Related Information

Commands: **bos create(8dfs)**, **bos shutdown(8dfs)**, **bos status(8dfs)**.

Files: **BosConfig(4dfs)**.

bos uninstall

Purpose

bos uninstall – Installs the former versions of binary files

Synopsis

bos uninstall -server *machine* **-file** *binary_file…* [**-dir** *alternate_dest*] [**-all**][{**-noauth** | **-localauth** }] [**-help**]

Options

-server *machine*
: Names the server machine on which the former versions of specified binary files are to be used. The BOS Server on this machine executes the command. To run this command using a privileged identity, specify the full DCE pathname of the machine. To run this command using the unprivileged identity **nobody** (the equivalent of running the command with the **-noauth** option), specify the machine's host name or IP address.

-file *binary_file*
: Names each binary file to be replaced with its next-oldest version (**.BAK**, if it exists; otherwise, **.OLD**). All specified files must reside in the same directory (*dcelocal* **/bin**, by default, or an alternate directory specified with the **-dir** option). Specify only filenames; if a pathname is provided for a file, the command ignores all but the final element.

-dir *alternate_dest*
: Provides the pathname of the directory in which all specified files reside. Omit this option if the files reside in the default directory, *dcelocal* **/bin**; otherwise, provide a full or relative pathname. Relative pathnames (pathnames that do not begin with a slash) are interpreted relative to the *dcelocal* directory on the machine specified by **-server**.

-all
: Directs the BOS Server on the indicated machine to remove all versions (current, **.BAK**, and **.OLD**) of the specified files. Only versions of the files that reside in the *dcelocal* **/bin** directory (or an alternate directory specified with the **-dir** option) are removed.

-noauth Directs **bos** to use the unprivileged identity **nobody** as the identity of the issuer of the command. The command fails if you use this option and DFS authorization checking is not disabled on the machine specified by **-server**. If you use this option, do not use the **-localauth** option.

-localauth Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

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

Description

The **bos uninstall** command replaces each binary file specified with the **-file** option with its next-oldest version. Use the **-server** option to specify the name of the server machine that houses the files to be manipulated. All specified files must reside in the same directory. By default, the command looks for the files in the *dcelocal* **/bin** directory; use the **-dir** option to name a different directory. Versions of the files in other directories on the specified machine are not affected.

The command applies the following algorithm to the removal of each binary file:

- If current, **.BAK**, and **.OLD** versions of the file exist, the command makes the **.BAK** version the current version and it makes the **.OLD** version the **.BAK** version.
- If any version of the file does not exist, the command omits it from the algorithm. For example, if no **.BAK** version exists, the command makes the **.OLD** version the current version.
- If the **-all** option is included with the command, the command removes all versions (current, **.BAK**, and **.OLD**) of the file that exist.
- The command displays the following message if no version of the file exists, or if only the current version exists and the **-all** option is omitted:

```
bos: failed to uninstall filename (there is no earlier
version present to reinstall)
```

where *filename* is the name of the file that cannot be replaced or removed.

The machine specified with the **-server** option should be the Binary Distribution machine for its CPU/operating system type. If it is not, the binary files are overwritten the next time the **upclient** process on the specified machine copies new (or different) versions of binary files via the **upserver** process on the Binary Distribution machine of its CPU/operating system type. (Note that the Update Server propagates binary files from Binary Distribution machines, but the BOS Server manipulates files when the **bos uninstall** command is issued; by default, it takes the Update Server 5 minutes to propagate binary files from a Binary Distribution machine.)

To make the machine specified by **-server** start using the reinstalled binary files immediately, issue the **bos restart** command. Otherwise, the binaries are not used until the BOS Server restarts the affected process at the new binary restart time specified in the *dcelocal* **/var/dfs/BosConfig** file. Use the **bos getrestart** and **bos setrestart** commands to inspect and set the new binary restart time. (The information in this paragraph applies *only* to affected processes already under the control of the BOS Server.)

Use the **bos install** command to install new versions of binary files on a server machine. Use the **bos prune** command to remove **.BAK** and **.OLD** files from the *dcelocal*/**bin** directory. (This command can also be used to remove core files from the *dcelocal* **/var/dfs/adm** directory.) Use the **bos getdates** command to check the time stamps on binary files.

Privilege Required

The issuer must be listed in the **admin.bos** file on the machine specified by **-server**.

Related Information

Commands: **bos getdates(8dfs)**, **bos getrestart(8dfs)**, **bos install(8dfs)**, **bos prune(8dfs)**, **bos restart(8dfs)**, **bos setrestart(8dfs)**, **upclient(8dfs)**, **upserver(8dfs)**.

Files: **BosConfig(4dfs)**.

bosserver

Purpose

Initializes the Basic OverSeer (BOS) Server process

Synopsis

bosserver [**-adminlist** *filename*] [**-noauth**][**-help**]

Options

-adminlist *filename*
: Specifies the file that contains principals and groups authorized to execute **bosserver** RPCs (usually using **bos** commands). If this option is omitted, the **bosserver** obtains the list of authorized users from the default administrative list file, *dcelocal* **/var/dfs/admin.bos**.

-noauth
: Invokes the **bosserver** process with DFS authorization checking turned off. In this mode, DFS processes, including the **bosserver** process, do not check to see whether issuers have the necessary privilege to enter administrative commands.

  This option is intended for use when the BOS Server is initially installed on a server machine. Because it starts the **bosserver** process with DFS authorization checking turned off, it allows the issuer to add members to the **admin.bos** administrative list and to add a key to the keytab file on the server machine.

  Use this mode sparingly, as it presents a security risk. Using this option forces all DFS server processes on the machine to run without DFS authorization checking.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

  The **help** and **apropos** commands available with all command suites are also available with the **bosserver** command. See the **bos help** and **bos apropos** pages for examples of using these commands.

Description

The Basic OverSeer Server (BOS Server) monitors other DFS server processes, such as the **flserver** and **ftserver** processes, running on the machine and restarts failed processes automatically (without the intervention of a system administrator). The BOS Server, or **bosserver** process, monitors each server process that has a process entry in the local **BosConfig** file. The **bosserver** process must be run on all DFS server machines. The **bosserver** command, which resides in *dcelocal*/**bin/bosserver**, is usually added to the proper system start-up file (/**etc/rc** or its equivalent); the process places itself in the background after it starts.

When it is started, the **bosserver** creates the *dcelocal*/**var/dfs/adm/BosLog** event log file if the file does not already exist. It then appends messages to the file. If the **BosLog** file exists when the **bosserver** process is started, the process moves it to the **BosLog.old** file in the same directory (overwriting the current **BosLog.old** file if it exists) before creating a new version to which to append messages.

The principals and groups in the **admin.bos** administrative list are authorized to issue BOS commands to stop, start, create, and modify server processes on that machine. For simplified administration, the same **admin.bos** administrative list can be used by all **bosserver** processes in the administrative domain.

The first time the **bosserver** process is initialized, it creates several directories (such as the *dcelocal* **/var/dfs/adm** directory and any nonexistent directories along this path), sets the owners to the appropriate identities, and sets the mode bits to provide appropriate access. The **bosserver** process also creates the *dcelocal*/**var/dfs/admin.bos** administrative list file and the *dcelocal* **/var/dfs/BosConfig** configuration file if either file does not already exist. On subsequent restarts, the process writes the following message to the **BosLog** file if the owners and mode bits of these objects are not set appropriately:

```
Bosserver reports inappropriate access on server directories.
```

See the reference page for the **bos status** command for information about the protections the BOS Server wants to see enforced.

Note: Your vendor can modify the owner of directories created by the BOS Server and the permissions those directories are created with. Refer to your vendor's documentation to determine the protections that apply for your version of DFS.

When initially installing the BOS Server on a server machine, use the **-noauth** option to initialize the **bosserver** process with DFS authorization checking disabled. This creates the **NoAuth** file in the *dcelocal***/var/dfs** directory on the local disk; when the file is present, DFS authorization checking is disabled on the machine.

With DFS authorization checking disabled, add members to the **admin.bos** list and add a key to the keytab file on the server machine. When these steps are complete, use the **bos setauth** command to enable DFS authorization checking. Because running with DFS authorization checking disabled is a serious security risk, enable DFS authorization checking as soon as the previous steps are complete. The **bos status** command can be used to determine whether DFS authorization checking is enabled or disabled on a machine; it displays the following message if DFS authorization checking is disabled on a machine (it does not display the message if DFS authorization checking is enabled):

```
Bosserver reports machine is not checking authorization.
```

Privilege Required

The issuer must be logged in as **root** on the local machine.

Output

If problems are encountered during initialization, the **bosserver** process displays error messages on standard error output. The **bosserver** process keeps an event log in the file *dcelocal* **/var/dfs/adm/BosLog**.

Related Information

Commands: **bos setauth(8dfs)**, **bos status(8dfs)**.

Files: **admin.bos(4dfs)**, **BosConfig(4dfs)**, **BosLog(4dfs)**, **NoAuth(4dfs)**.

butc

Purpose

butc – Initializes a Tape Coordinator process

Synopsis

butc [**-tcid** *tc_number*] [**-debuglevel** *trace_level*] [**- cell** *cellname*] [**- help**]

Options

-tcid *tc_number*

Specifies the Tape Coordinator ID (TCID) associated with the Tape Coordinator to be initialized. The issuer of **bak** commands uses this number to indicate which Tape Coordinator is to execute a command.

Legal values are the integers from 0 to 1023. The value must match the value assigned to this Tape Coordinator's associated tape drive in the **TapeConfig** file. If this option is omitted, the default is **0** (zero).

-debuglevel *trace_level*

Specifies the kinds of messages the Tape Coordinator produces in its monitoring window. The following two values are legal:

- A value of **0** (zero) indicates that the Tape Coordinator prompts the issuer only to place new tapes in the drive; the process does not report on its activities (other than to display some output as necessary for operations it executes). This is the default value.
- A value of **1** indicates that the Tape Coordinator reports on its activities as it restores filesets, in addition to prompting for new tapes as necessary.

-cell *cellname*

Specifies the cell with respect to which the Tape Coordinator is to run. The Tape Coordinator communicates with the Backup Server in the specified cell. The Tape Coordinator can manipulate data in only the specified cell. A host entry must already be defined for the Tape Coordinator machine in the Backup Database of the specified cell.

If this option is omitted, the default is the local cell of the issuer of the command.

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

The **help** and **apropos** commands available with all command suites are also available with **butc**. See the **bos help** and **bos apropos** pages for examples using these commands.

Description

The **butc** command starts a Tape Coordinator process on a Tape Coordinator machine (a machine having a tape drive and an associated Tape Coordinator). The **TapeConfig** file must reside in the directory named *dcelocal*/**var/dfs/backup** on the Tape Coordinator machine, and it must contain a single line specifying information about a tape drive and its associated Tape Coordinator if the **butc** process is to start the Tape Coordinator for the drive. A machine to be configured as a Tape Coordinator machine must be a DCE client. Fewer configuration steps are required if the machine is also some type of DFS server machine. (See Part 1 of this manual for complete details about configuring a Tape Coordinator machine.)

The binary file for the **butc** process resides in *dceshared*/**bin/butc**. Depending on the operations it executes, the **butc** process that runs as a result of this command contacts the Backup Database (by way of the Backup Server process), the Fileset Location Database (by way of the Fileset Location Server process), or Fileset Server processes.

Enter the **butc** command in a separate terminal session for each Tape Coordinator. (In windowing systems, this generally means a separate window for each Tape Coordinator.) Because the Tape Coordinator must run in the foreground, the terminal session where the **butc** command is issued is unavailable for subsequent commands. Instead, the Tape Coordinator uses it as a dedicated monitoring window on which to display both trace information about filesets it restores and prompts for the insertion of additional tapes into its associated drive. The monitoring window must remain open as long as the Tape Coordinator runs. To stop a Tape Coordinator process, enter an interrupt signal (**<Ctrl-c>** or its equivalent) in the process's monitoring window.

The **butc** program also writes output to two ASCII files in the directory *dcelocal*/**var/dfs/backup** on the local disk of the Tape Coordinator machine:

TL_ *device_name*

The **TL_** *device_name* file (where *device_name* is the device name of the tape drive with which the process is associated) is a log file that contains execution information about operations performed by the **butc** process. The level of detail to which each operation is described depends on the operation.

TE_ *device_name*

The **TE_** *device_name* file (where *device_name* is again the device name of the tape drive with which the process is associated) is an error file that contains information about problems encountered by the **butc** process.

The files contain similar information. For example, if you use the **bak dump** command to dump 100 filesets, the log file lists both the names of filesets that were successfully dumped to tape and the names of filesets that, for some reason, were omitted from the dump; the error file, on the other hand, lists the names of only those filesets that were omitted from the dump.

Each time the **butc** process is started for a tape drive and Tape Coordinator pair, it automatically creates the two files. It then appends messages to the files as necessary. If the files already exist when the **butc** process is started, the process moves the current versions to files that end with **.old** extensions (for example, **TL_** *device_name* **.old**) before creating new versions of the files to which to append messages. The process overwrites current **.old** files if they exist.

No maintenance is required for the log and error files associated with any tape drive; the files are created automatically each time the **butc** process for a tape drive and Tape Coordinator pair is started. However, the files should be browsed periodically to ensure that operations such as dumps and restores are completing without problems. For example, if a file cannot be dumped because a necessary Fileset Server or Fileset Location Server is unavailable at the time of the dump, the **butc** program writes an appropriate message to the log and error files.

Privilege Required

The issuer must have write and execute permissions on the *dcelocal* **/var/dfs/backup** directory, the directory in which the **butc** process creates its log and error files.

Related Information

Commands: **bak(8dfs)**.

Files: **TapeConfig(4dfs)**, **TE(4dfs)**, **TL(4dfs)**.

cm

Purpose

cm – Introduction to the **cm** command suite

Options

The following options are used with many **cm** commands. They are also listed with the commands that use them.

-path {*filename* | *directory_name*}
Names the files, directories, or both to be used with the command.

-help Prints the online help for the command. All other valid options specified with this option are ignored. For complete details about receiving help, see the **dfs_intro(8dfs)** reference page.

Description

Commands in the **cm** command suite are issued by administrative users to set cache parameters and to update cached information on local workstations. Certain commands in the **cm** suite are available to all users to determine machine and cell information.

The files described in the following sections are used by the Cache Manager to determine its initial configuration and to store and track cached data. Each DFS client machine stores machine-specific versions of these files on its local disk.

The CacheInfo File

The *dcelocal*/**etc/CacheInfo** file specifies the Cache Manager's initial configuration. It is manually created during DFS client installation. The Cache Manager checks the file at initialization to determine certain cache configuration information.

The file is a one-line ASCII file that contains three fields separated by colons. The fields provide the following information:

- The local directory where the Cache Manager mounts the DCE global namespace. The default is the global namespace designation (**/...**).

- The local directory to serve as the cache directory. The Cache Manager stores the **CacheItems**, **FilesetItems** , and V files in this directory. The default, *dcelocal* **/var/adm/dfs/cache**, can be overridden to direct the Cache Manager to store the files in a different directory.
- The size of the cache in 1024-byte (1-kilobyte) blocks.

The CacheItems File

The *dcelocal*/**var/adm/dfs/cache/CacheItems** file is a binary file created and maintained by the Cache Manager. The file records information such as the file ID number and data version number of each V file on a client machine using a disk cache. *Never directly modify or delete this file*; doing so can cause the kernel to panic.

The FilesetItems File

The *dcelocal*/**var/adm/dfs/cache/FilesetItems** file is a binary file created and maintained by the Cache Manager. The file records the fileset-to-mount-point mapping for each fileset accessed by the Cache Manager. The mappings allow the Cache Manager to respond correctly to commands such as **pwd**. *Never directly modify or delete this file*; doing so can cause the kernel to panic.

V Files

The *dcelocal*/**var/adm/dfs/cache/V** *n* files, or V files, hold chunks of cached data on a client machine using a disk cache. In the name of an actual V file, *n* is an integer; each V file has a unique name (for example, **V1**, **V2**, and so on). The format of a V file depends on the information it contains.

By default, each V file holds up to 65,536 bytes (64 kilobytes) of data. The default size can be overridden with the **dfsd** command. *Never directly modify or delete a V file*; doing so can cause the kernel to panic.

Cautions

Specific cautionary information is included with individual commands.

Receiving Help

There are several different ways to receive help about DFS commands. The following examples summarize the syntax for the different help options:

$ **man cm**

Displays the reference page for the command suite.

$ **man cm_** *command*
Displays the reference page for an individual command. You must use an _ (underscore) to connect the command suite to the command name. *Do not use the underscore when issuing the command in DFS.*

$ **cm help** Displays a list of commands in a command suite.

$ **cm help***command*
Displays the syntax for a single command.

$ **cm apropos -topic** *string*
Displays a short description of any commands that match the specified *string*.

Consult the **dfs_intro(8dfs)** reference page for complete information about the DFS help facilities.

Privilege Required

Specific privileges required by each command are listed with individual commands.

Related Information

Commands: **cm apropos(8dfs)**, **cm checkfilesets(8dfs)**, **cm flush(8dfs)**, **cm flushfileset(8dfs)**, **cm getcachesize(8dfs)** , **cm getdevok(8dfs)**, **cm getsetuid(8dfs)** , **cm help(8dfs)**, **cm lscellinfo(8dfs)** , **cm lsstores(8dfs)**, **cm resetstores(8dfs)** , **cm setcachesize(8dfs)**, **cm setdevok(8dfs)** , **cm setsetuid(8dfs)**, **cm statservers(8dfs)** , **cm sysname(8dfs)**, **cm whereis(8dfs)** , **dfs_intro(8dfs)**, **dfsd(8dfs)**.

Files: **CacheInfo(4dfs)**, **CacheItems(4dfs)**, **FilesetItems(4dfs)**, **Vn(4dfs)**.

cm apropos

Purpose

cm apropos – Shows each help entry containing a specified string

Synopsis

cm apropos -topic *string* [**-help**]

Options

-topic *string* Specifies the keyword string for which to search. If it is more than a single word, surround the string with " " (double quotes) or other delimiters. Type all strings for **cm** commands in lowercase letters.

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

Description

The **cm apropos** command displays the first line of the online help entry for any **cm** command containing the string specified by **-topic** in its name or short description.

To display the syntax for a command, use the **cm help** command.

Privilege Required

No privileges are required.

Output

The first line of an online help entry for a command names the command and briefly describes its function. This command displays the first line for any **cm** command where the string specified by **-topic** is part of the command name or the first line.

Examples

The following command lists all **cm** commands that have the word **cache** in their names or short descriptions:

$ cm apropos cache

```
flush: flush file data and status information from cache
getcachesize: get cache usage info
setcachesize: set cache size
```

Related Information

Commands: **cm help(8dfs)**.

cm checkfilesets

Purpose

Forces the Cache Manager to update fileset-related information

Synopsis

cm checkfilesets [**-help**]

Options

-help Prints the online help for this command.

Description

The **cm checkfilesets** command forces the Cache Manager to discard its table of mappings between fileset names and fileset ID numbers. Because the Cache Manager needs the information in the table to fetch files, this command forces the Cache Manager to fetch the most recent information available about a fileset from the Fileset Location Server before the Cache Manager can fetch any more files from that fileset. (Normally, the Cache Manager flushes the table and constructs a new one every hour.)

This command is most useful if you know that a fileset name has changed or that there is a release of new read-only replicas. Issuing this command forces the Cache Manager to reference the fileset with the new name or the new read-only replica.

To force the Cache Manager to discard a cached file or directory, use the **cm flush** command. To force the Cache Manager to discard any data cached from filesets containing specified files or directories, use the **cm flushfileset** command.

Privilege Required

No privileges are required.

Related Information

Commands: **cm flush(8dfs)**, **cm flushfileset(8dfs)**.

cm flush

Purpose

cm flush – Forces the Cache Manager to discard data cached from specified files or directories

Synopsis

cm flush [**-path** {*filename* | *directory_name*}…] [**-help**]

Options

-path {*filename* | *directory_name*}
: Specifies each file or directory to be flushed. A file for which a full pathname is not specified is assumed to reside in the current working directory. In the case of a directory, all the name mappings and blocks associated with the directory are flushed; data cached from files or subdirectories that reside in the directory is not flushed. If this option is omitted, the current working directory is flushed.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **cm flush** command forces the Cache Manager to flush data cached from each file or directory specified with the **-path** option. All data cached from these files and directories is discarded. The next time the data is requested, the Cache Manager contacts the File Exporter to obtain the current version, along with new tokens and other associated status information.

This command does not discard any altered data in the cache not written to the central copy maintained by the File Exporter. It also does not affect data in the buffers of application programs.

It is also possible to flush all cached data that resides in the same fileset as a specific file or directory with the **cm flushfileset** command. To force the Cache Manager to update fileset-related information, use the **cm checkfilesets** command.

Privilege Required

No privileges are required.

Examples

The following command flushes the file **projectnotes**, which is in the current working directory, and all data from the subdirectory **plans** from the cache:

```
$ cm flush projectnotes plans/*
```

Related Information

Commands: **cm checkfilesets(8dfs)**, **cm flushfileset(8dfs)**.

cm flushfileset

Purpose

cm flushfileset – Forces the Cache Manager to discard data cached from filesets that contain specified files or directories

Synopsis

cm flushfileset [**-path** {*filename* | *directory_name*}…] [**-help**]

Options

-path {*filename* | *directory_name*}
: Specifies a file or directory from each fileset to be flushed. A file for which a full pathname is not specified is assumed to reside in the current working directory. If this option is omitted, the fileset containing the current working directory is flushed.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **cm flushfileset** command forces the Cache Manager to flush data cached from filesets that contain each file or directory specified with the **-path** option. All data cached from these filesets is discarded. The next time the data is requested, the Cache Manager contacts the File Exporter to obtain the current version, along with new tokens and other associated status information.

This command does not discard any altered data in the cache not written to the central copy maintained by the File Exporter. It also does not affect data in the buffers of application programs.

It is also possible to flush data cached from specific files or directories with the **cm flush** command. To force the Cache Manager to update fileset-related information, use the **cm checkfilesets** command.

Privilege Required

No privileges are required.

Examples

The following command flushes data cached from the fileset containing the current working directory and the directory **reports**, both of which are at the same level in the file tree:

```
$ cm flushf . ../reports
```

Related Information

Commands: **cm checkfilesets(8dfs)**, **cm flush(8dfs)**.

cm getcachesize

Purpose

cm getcachesize – Shows the current size of the cache, the amount of cache in use, and the type of cache

Synopsis

cm getcachesize [**-help**]

Options

-help Prints the online help for this command.

Description

The **cm getcachesize** command displays the current size of the cache available to the Cache Manager and the amount in use when the command is issued. It also displays the type of cache in use on the machine. The command works both on machines using disk caching and on machines using memory caching.

The information displayed by the command comes from the kernel of the workstation on which the command is issued. On machines using disk caching, the current cache size may disagree with the default setting specified in the **CacheInfo** file if the cache size was set with the **cm setcachesize** command. Regardless of the type of caching (disk or memory) in use, the size may also disagree with the default setting if it was changed with the **dfsd** command.

Privilege Required

No privileges are required.

Output

The **cm getcachesize** command displays the following output:

```
DFS using amount of the cache's available size 1K byte (type) blocks.
```

In the output, *amount* is the number of kilobyte blocks the Cache Manager is currently using, **size** is the total number of kilobyte blocks available to the Cache Manager (the current cache size), and *type* is the type of cache (disk or memory) in use on the machine.

Examples

The following command shows the output on a machine with a 25,000 kilobyte disk cache:

```
$ cm getcachesize

DFS using 22876 of the cache's available 25000 1K byte (disk) blocks.
```

Related Information

Commands: **cm setcachesize(8dfs)**, **dfsd(8dfs)**.

Files: **CacheInfo(4dfs)**.

cm getdevok

Purpose

cm getdevok – Shows whether device files from specified filesets are honored by the Cache Manager

Synopsis

cm getdevok [**-path** {*filename* | *directory_name*}…] [**-help**]

Options

-path {*filename* | *directory_name*}
Names a file or directory from each fileset whose device file status information is to be displayed. If this option is omitted, status information is displayed for the fileset containing the current working directory.

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

Description

The **cm getdevok** command reports whether the Cache Manager honors device files that reside in the indicated filesets. Indicate each fileset for which you want device file status information by specifying the name of a file or directory in the fileset with the **-path** option. This information comes from the kernel of the workstation on which the command is issued.

System administrators set whether device files are to be honored on a per-fileset and per-Cache-Manager basis with the **cm setdevok** command. By default, the Cache Manager does not honor device files from a fileset. (The UNIX kernel always honors device files stored in the **/dev** directory.)

Privilege Required

No privileges are required.

Output

The **cm getdevok** command first displays the line

```
Fileset pathname status:
```

In the output, *pathname* is the name of a file or directory specified with the **-path** option. For each specified file or directory, the following output values are possible for the fileset on which it resides:

`device files allowed`
: Indicates that device files from the fileset are honored.

`device files not allowed`
: Indicates that device files from the fileset are not honored.

`cm: the fileset on which ‘ pathname ’resides does not exist`
: Indicates that the specified pathname is invalid.

Examples

The following command indicates that device files from the fileset that contains the directory */.../abc.com/fs/usr/jlw* are not honored by the Cache Manager:

```
$ cm getdevok /.../abc.com/fs/usr/jlw
```

```
/.../abc.com/fs/user/jlw status: device files not allowed
```

Related Information

Commands: **cm setdevok(8dfs)**.

cm getpreferences

Purpose

cm getpreferences – Displays the Cache Manager's preferences for File Server or Fileset Location (FL) Server machines

Synopsis

cm getpreferences [**-path** *filename*] [**-numeric**][**-fldb**][**-help**]

Options

-path *filename*
: Specifies the full pathname of a file to which the command is to write the Cache Manager server preferences that it reports. If the specified file already exists, the command overwrites it. The command fails if the specified pathname names a directory. Omit this option to display the preferences on standard output (**stdout**).

-numeric
: Directs the command to display the IP addresses rather than the host names of the File Servers or FL Servers. Omit this option to display the host name (for example, **fs1.abc.com**) of each machine.

-fldb
: Directs the command to display the host names or IP addresses of the FL Servers and their respective ranks.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

DESCRIPTION

The **cm getpreferences** command displays the current set of entries in a Cache Manager preference list. The Cache Manager preference list is created each time a Cache Manager is initialized with the **dfsd** command (each time the client machine is rebooted). Each Cache Manager maintains its own separate preference list. Each entry in the list consists of the IP address of an FL Server or File Server and an automatically assigned preference value. New entries are automatically added to the preference list as necessary when filesets are first referenced.

In operation, when the Cache Manager needs to contact an FL Server, it consults its list of FL Servers and attempts to contact a server at the address with the lowest-ranking value in the preference list. Similarly, when a Cache Manager needs to contact a File Server, it consults its preference list and contacts a suitable File Server at the address with the lowest-ranking value.

If the Cache Manager cannot access a server at the address with the lowest preference rank (because of a problem with either the machine or the network), the Cache Manager attempts to access a similar server at the address with the next lowest rank. It continues in this way until it either succeeds in accessing an appropriate server or determines that all such servers are unavailable.

By default, the Cache Manager assigns preferences that make sensible choices based on the location of servers. Therefore, you should adjust the default values only if there is a compelling reason. The default values force the Cache Manager to attempt to connect to servers in the following order:

1. The same machine as the client (default rank of 5000).
2. The same subnetwork as the client (default rank of 20000).
3. The same network as the client (default rank of 30000).
4. Different networks (default rank of 40000).

For example, a server on the same machine as the Cache Manager receives a rank of 5000, while a server on the same subnetwork receives a rank of 20000. The entry with the lowest-ranking value has the highest "preference." Thus, a server with a preference value of 5000 will be chosen before a server with a rank of 20000.

Should two servers be assigned the same preference value, such as two File Servers on the same subnetwork both receiving a default value of 20000, the server with the lowest round-trip value is chosen. Each server is assigned a random round-trip value when the Cache Manager is initialized. The assigned round-trip value is always higher than the upper bound for stored actual round-trip values. This ensures that an actual round-trip value will always be chosen over assigned values. The **cm getpreferences** command does not display the round-trip value.

The **cm getpreferences** command displays information on standard output by default. Use the **-path** option to specify the complete pathname of a file to which the command is to write its output. If you include the **-path** option, the command displays no output on standard output.

Privilege Required

No privileges are required.

OUTPUT

The **cm getpreferences** command displays a separate line of output for each Cache Manager preference list entry. By default, each line consists of the host name of a File Server or FL Server followed by the preference value, as follows:

```
hostname          rank
```

where *hostname* is the name of a File Server or FL Server, and *rank* is the rank associated with the machine. If the **-numeric** option is included with the command, the command displays the IP address, in dotted decimal format, instead of the machine's name. The command also displays the IP address of any machine whose name it cannot determine (for example, if a network outage prevents it from resolving the address into the name).

EXAMPLES

The following command displays the preference list entries associated with the Cache Manager on the local machine. The local machine belongs to the DCE cell named **dce.abc.com**; the ranks of the File Servers from the **dce.abc.com** cell are lower than the ranks of the File Servers from the foreign cell, **dce.def.com**. The command shows the IP addresses, not the names, of two machines from the foreign cell because it cannot currently determine their names.

```
$ cm getp

fs2.abc.com              20000
fs3.abc.com              30000
fs1.abc.com              20000
fs4.abc.com              30000
server1.def.com          40000
121.86.33.34             40000
server6.def.com          40000
121.86.33.37             40000
```

The following command displays the same Cache Manager's preference list entries, but the **-numeric** option is included with the command to display the IP addresses rather than the host names of all File Servers. The IP address of the local machine is **128.21.16.221**. The two File Servers on the same subnetwork as the local machine have a rank of 20000; the two File Servers on a different subnetwork in the same network as the local machine have a rank of 30000; the remaining File Servers are in a different network, so they have a rank of 40000. The round-trip value for each preference list entry (used to select a connection when multiple entries have the same rank) is not displayed by the command.

```
$ cm getp -n


128.21.16.214            20000
128.21.18.99             30000
128.21.16.212            20000
128.21.18.100            30000
121.86.33.41             40000
121.86.33.34             40000
121.86.33.36             40000
121.86.33.37             40000
```

Related Information

Commands: **cm setpreferences(8dfs)**.

cm getprotectlevels

Purpose

cm getprotectlevels – Returns the current DCE RPC authentication level settings for communications between the Cache Manager and File Servers

Synopsis

cm getprotectlevels [**-help**]

OPTIONS

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

DESCRIPTION

The **cm getprotectlevels** command returns the current Cache Manager DCE RPC authentication level settings. The returned values include separate local and foreign cell settings for the initial and minimum authentication levels for communications with File Servers.

The Cache Manager and File Server default settings are such that communications occur at the Packet Integrity authentication level. (Packet integrity both makes certain that the data is received from the expected principal and that the data has not been modified.)

The authentication bounds for the File Server itself are set through the **fxd** command. In addition to a general pair of upper and lower bounds for all communications between the File Server and Cache Manager, administrators can also set advisory bounds on a per fileset basis. At present, these advisory levels serve only to bias the Cache Manager's selection of an initial authentication level (they may be enforced in a future version of DFS). Advisory bounds are set through the **fts setprotectlevels** command and are stored in the FLDB record for that fileset. You can display the current advisory RPC authentication bounds for a fileset through either the **fts lsfldb** or **fts lsft** commands.

Privilege Required

No privileges are required.

OUTPUT

The output consists of the following four lines:

```
Initial protection level in the local cell: level
Minimum protection level in the local cell: level
Initial protection level in non-local cells: level
Minimum protection level in non-local cells: level
```

Where *level* is one of the various DCE RPC authentication levels, whose possible values are

- **rpc_c_protect_level_default** - default : Use the DCE default authentication level.
- **rpc_c_protect_level_none** - none : Perform no authentication.
- **rpc_c_protect_level_connect** - connect : Authenticate only when the Cache Manager establishes a connection with the File Server.
- **rpc_c_protect_level_call** - call : Authenticate only at the beginning of each RPC received.
- **rpc_c_protect_level_pkt** - packet : Ensure that all data received is from the expected principal.
- **rpc_c_protect_level_pkt_integ** - packet integrity : Authenticate and verify that none of the of the data transferred has been modified.
- **rpc_c_protect_level_pkt_privacy** - packet privacy : Perform authentication as specified by all of the previous levels and also encrypt each RPC argument value.

EXAMPLES

The following command returns the current authentication levels for communications between the Cache Manager and Files Servers:

$ cm getprotectlevels

```
Initial protection level in the local cell:
                                        rpc_c_protect_level_pkt_integ
Minimum protection level in the local cell: rpc_c_protect_level_none
Initial protection level in non-local cells:
                                        rpc_c_protect_level_pkt_integ
Minimum protection level in non-local cells: rpc_c_protect_level_pkt
```

RELATED INFORMATION

Commands: **cm setprotectlevels(8dfs)**, **fxd(8dfs)**, **dfsd(8dfs)**, **fts setprotectlevels(8dfs)**

cm getsetuid

Purpose

cm getsetuid – Shows the status of **setuid** programs from specified filesets

Synopsis

cm getsetuid [**-path** {*filename* | *directory_name*}…] [**-help**]

Options

-path {*filename* | *directory_name*}
: Names a file or directory from each fileset whose **setuid** permission is to be displayed. If this option is omitted, permission information is displayed for the fileset containing the current working directory.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **cm getsetuid** command reports whether the Cache Manager allows **setuid** programs from the indicated filesets to run with **setuid** permission. Indicate each fileset whose **setuid** permission is desired by specifying the name of a file or directory in the fileset with the **-path** option. This information comes from the kernel of the workstation on which the command is issued.

Note that **setuid** programs are effective only in the local environment. A **setuid** program can change only the local identity under which a program runs; it cannot change the DCE identity with which a program executes because it provides no Kerberos tickets. DCE does not recognize the change to the local identity associated with a **setuid** program.

Because **setgid** programs on filesets are enabled or disabled along with **setuid** programs, this command also reports the status of **setgid** programs on the indicated filesets. System administrators set **setuid** and **setgid** status on a per-fileset and per-Cache Manager basis with the **cm setsetuid** command. By default, the Cache Manager does not allow **setuid** programs from a fileset to execute with **setuid** permission.

Privilege Required

No privileges are required.

Output

The **cm getsetuid** command first displays the line

```
Fileset pathname status:
```

In the output, *pathname* is the name of a file or directory specified with the **-path** option. For each specified file or directory, the following output values are possible for the fileset on which it resides:

`setuid allowed`
: Indicates that **setuid** and **setgid** programs from the fileset are enabled.

`no setuid allowed`
: Indicates that **setuid** and **setgid** programs from the fileset are disabled.

`cm: the fileset on which ‘` *pathname* `’resides does not exist`
: Indicates that the specified pathname is invalid.

Examples

The following command indicates that **setuid** and **setgid** programs from the fileset that contains the directory */.../abc.com/fs/usr/jlw* are disabled:

```
$ cm getsetuid /.../abc.com/fs/usr/jlw
```

```
Fileset /.../abc.com/fs/usr/jlw status: no setuid allowed
```

Related Information

Commands: **cm setsetuid(8dfs)**.

cm help

Purpose

cm help – Shows syntax of specified **cm** commands or lists functional descriptions of all **cm** commands

Synopsis

cm help [**-topic** *string*]… [**-help**]

Options

-topic *string* Specifies each command whose syntax is to be displayed. Provide only the second part of the command name (for example, **flush**, not **cm flush**). If this option is omitted, the output provides a short description of all **cm** commands.

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

Description

The **cm help** command displays the first line (name and short description) of the online help entry for every **cm** command if **-topic** is not provided. For each command name specified with **-topic**, the output lists the entire help entry.

Use the **cm apropos** command to show each help entry containing a specified string.

Privilege Required

No privileges are required.

Output

The online help entry for each **cm** command consists of the following two lines:

- The first line names the command and briefly describes its function.

- The second line, which begins with **Usage:**, lists the command options in the prescribed order.

Examples

The following command displays the online help entry for the **cm flush** command:

```
$ cm help flush
```

```
cm flush: flush file from cache
Usage: cm flush [-path {<filename> | <directory_name>}...] [-help]
```

Related Information

Commands: **cm apropos(8dfs)**.

cm lscellinfo

Purpose

cm lscellinfo – Shows database server machines in cells known to the Cache Manager

Synopsis

cm lscellinfo [**-help**]

Options

-help Prints the online help for this command.

Description

The **cm lscellinfo** command formats and displays the Cache Manager's kernel-resident list of Fileset Location Database (FLDB) machines in its home cell and any foreign cells the Cache Manager has accessed. This information comes from the kernel of the workstation on which the command is issued.

Privilege Required

No privileges are required.

Output

The output contains one line for the local cell and one line for each cell listed in the kernel that the Cache Manager has accessed. Each cell name is followed by a list of its database server machines (referred to as *hosts*). In a multihomed server environment (an FLDB machine can have up to four IP addresses listed in the Cache Manager's preferences), *hosts* corresponds to the IP addresses or host names that the Cache Manager is currently using to access each particular FLDB machine. Therefore, the command output lists only one machine name for each FLDB machine.

Examples

The following command shows output for several cells:

$ cm lscellinfo

```
Cell abc.com on hosts fs2.abc.com
Cell state.edu on hosts fs11.fs.state.edu
```

cm lsstores

Purpose

Lists filesets that contain data the Cache Manager cannot write back to a File Server machine

Synopsis

cm lsstores [**-help**]

Options

-help Prints the online help for this command.

Description

The **cm lsstores** command lists the fileset ID numbers of filesets that contain data the Cache Manager cannot write back to a File Server machine. This information comes from the kernel of the workstation on which the command is issued.

On occasion, a File Server machine may be unavailable to the Cache Manager (possibly because the File Server machine is down or because a network problem prevents the Cache Manager from contacting the machine). In such cases, the Cache Manager cannot write data back to the File Server machine. The Cache Manager displays a message on the screen to notify the user that it cannot write the data to the unavailable machine. If possible, it also returns a failure code to the application program using the data.

The Cache Manager keeps the unstored data in the cache and continues to attempt to contact the File Server machine until it can store the data. (The frequency with which it attempts to reach a File Server machine is defined with the **-pollinterval** option of the **fxd** command issued on that File Server machine.) In the meantime, corrective measures can be taken to alleviate the problem that prevents the data from being stored; for example, the File Server machine can be restarted. Once the problem is alleviated, the Cache Manager can reach the File Server machine and store the data.

The Cache Manager discards unstored data only when

- It needs to make room in the cache for other data. Given an average-sized cache with average usage, the Cache Manager rarely needs to discard unstored data.
- The **cm resetstores** command is issued to force it to discard unstored data from the cache.

Because unstored data discarded from the cache cannot be recovered, any problem that prevents data from being written to a File Server machine should be handled promptly.

Privilege Required

No privileges are required.

Output

If the Cache Manager cannot store data to one or more filesets, the command displays the fileset ID number of each fileset to which data cannot be stored. If the Cache Manager has been able to store all data, the command displays the following message:

```
No failed stores are being retried.
```

Related Information

Commands: **cm resetstores(8dfs)**, **fxd(8dfs)**.

cm resetstores

Purpose

Cancels attempts by the Cache Manager to contact unavailable File Server machines and discards all data the Cache Manager cannot store to such machines

Synopsis

cm resetstores [**-help**]

Options

-help Prints the online help for this command.

Description

The **cm resetstores** command cancels the Cache Manager's continued attempts to contact unavailable File Server machines. *All* data that the Cache Manager cannot store to such File Server machines is discarded; there is no way to selectively discard individual files or data from specific filesets.

Occasionally, a File Server machine may be unavailable to the Cache Manager (possibly because the File Server machine is down or because a network problem prevents the Cache Manager from contacting the machine). In such cases, the Cache Manager cannot write data back to the File Server machine. The Cache Manager displays a message on the screen to notify the user that it cannot write the data to the unavailable machine. If possible, it also returns a failure code to the application program using the data.

The Cache Manager keeps the unstored data in the cache and continues to attempt to contact the File Server machine until it can store the data. (The frequency with which it attempts to reach a File Server machine is defined with the **-pollinterval** option of the **fxd** command issued on that File Server machine.) In the meantime, corrective measures can be taken to alleviate the problem that prevents the data from being stored; for example, the File Server machine can be restarted. Once the problem is alleviated, the Cache Manager can reach the File Server machine and store the data.

The Cache Manager discards unstored data only when

- It needs to make room in the cache for other data. Given an average-sized cache with average usage, the Cache Manager rarely needs to discard unstored data.
- The **cm resetstores** command is issued to force it to discard unstored data from the cache.

Because unstored data discarded from the cache cannot be recovered, any problem that prevents data from being written to a File Server machine should be handled promptly.

Note that the **cm resetstores** command affects only data that could not be stored to a File Server machine; it does not affect other data in the cache. Nonetheless, be cautious when issuing the **cm resetstores** command. Issue the **cm lsstores** command first to list the fileset ID numbers of filesets that contain data the Cache Manager cannot write to a File Server machine; examine the output of the command to be sure that you know from which filesets unstored data will be discarded. (You may also be able to use this information to ensure that unstored data from the indicated filesets can safely be discarded.)

Privilege Required

The issuer must be logged in as **root** on the local machine.

Related Information

Commands: **cm lsstores(8dfs)**, **fxd(8dfs)**.

cm setcachesize

Purpose

cm setcachesize − Sets the size of a disk cache

Synopsis

cm setcachesize {**-size** *kbytes* | **-reset** } [**-help**]

Options

-size *kbytes* Specifies the number of 1-kilobyte blocks the Cache Manager can use for the cache. The smallest allowable value is 1. Specifying a value of 0 (zero) sets the cache size to the default specified in the **CacheInfo** file. Use this option or use the **-reset** option.

-reset Returns the cache size to the value set when the machine was last booted. Use this option or use the **-size** option.

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

Description

The **cm setcachesize** command changes the amount of local disk space the Cache Manager uses for its data cache. Specify a number of kilobyte blocks. Do not set the cache size to exceed 90% of the actual disk space available for the cache directory; the cache implementation itself requires a small amount of room on the partition. *Do not use this command on a machine using a memory cache.*

The cache size cannot be set to a value less than twice the value of the chunk size in use by the Cache Manager. If a value smaller than twice the chunk size is specified with the **-size** option, the following message is displayed:

```
path: Cache size of size is too small; value was rounded up.
```

In the message, *path* is the specified path to the **cm** program (usually just **cm**) and **size** is the size, in kilobytes, specified with the command. The standard message reporting the new cache size (the size to which the cache was rounded) is then displayed; see the section on output for an example of the message.

To return the cache size to the default value specified in the **CacheInfo** file, specify 0 (zero) as the number of kilobyte blocks with the **-size** option. To return the cache size to the value set when the machine was last booted, use the **-reset** option instead of the **-size** option; the **-reset** option also sets the size to the amount specified in the **CacheInfo** file unless the **-blocks** option was used with the **dfsd** command to override the **CacheInfo** value, in which case the value set with the **dfsd** command is used.

The **cm getcachesize** command displays the current cache size and the amount of space in use for both disk and memory caches. It also reports the type of cache (disk or memory) in use.

Privilege Required

The issuer must be logged in as **root** on the local machine.

Output

The following message is displayed whenever this command is used to set the cache size:

path: New cache size set: **size**.

In the message, *path* is the specified path to the **cm** program (usually just **cm**) and **size** is the new cache size, in kilobytes.

Examples

The following command sets the cache size to 25,000 kilobyte blocks:

```
# cm setca 25000
cm: New cache size set: 25000.
```

The following command resets the cache size to the value set when the machine was last booted (50,000 kilobyte blocks, in this case):

```
# cm setca -r
cm: New cache size set: 50000.
```

Related Information

Commands: **cm getcachesize(8dfs)**, **dfsd(8dfs)**.

Files: **CacheInfo(4dfs)**.

cm setdevok

Purpose

cm setdevok – Specifies whether device files from specified filesets are honored by the Cache Manager

Synopsis

cm setdevok [**-path** {*filename* | *directory_name*}…] [**-state** {on | off}] [**-help**]

Options

-path {*filename* | *directory_name*}
Names a file or directory from each fileset whose device file status is to be changed. If this option is omitted, the status is changed for the fileset containing the current working directory.

-state Specifies whether device files from the filesets indicated with **-path** are to be honored. Specify **on** with this option to honor device files from the indicated filesets; specify **off** with this option to prevent device files from the indicated filesets from being honored. If this option is omitted, device files from the filesets are honored. (The command has no effect if device files were already honored.)

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

Description

The **cm setdevok** command specifies whether device files from the indicated filesets are honored by the Cache Manager. Indicate each fileset whose device files are to be honored or not honored by specifying the name of a file or directory in the fileset with the **-path** option. Device files are honored on a per-fileset and per-Cache Manager basis. This command is commonly included in a start-up file (**/etc/rc** or its equivalent) to honor device files at machine startup.

If **on** is specified with the **-state** option, or if the **-state** option is omitted, the Cache Manager honors device files from the indicated filesets. If **off** is specified with the

-state option, the Cache Manager does not honor device files from the indicated filesets. By default, the Cache Manager does not honor device files from a fileset. (The UNIX kernel always honors device files stored in the **/dev** directory.)

The **cm getdevok** command displays whether the Cache Manager honors device files from indicated filesets.

Privilege Required

The issuer must be logged in as **root** on the local machine.

Examples

The following command causes device files that reside on the fileset that contains the directory */.../abc.com/fs/usr/jlw* to be honored:

```
# cm setdevok /.../abc.com/fs/usr/jlw
```

Related Information

Commands: **cm getdevok(8dfs)**.

cm setpreferences

Purpose

cm setpreferences – Sets the Cache Manager's preferences for File Server or File Location (FL) Server machines

Synopsis

cm setpreferences [**-server** *machine rank…*] [**-path** *filename*] [**-stdin**][**-fldb**] [**-help**]

Options

-server *machine rank*

Specifies File Server or FL Server preference entries, with each entry consisting of a machine specification (a host name or IP address) and a preference rank. Separate each machine specification and each rank with one or more spaces. By default, the **-server** option specifies File Server machine entries; add the **-fldb** option to specify FL Server machine entries. Each server machine can have multiple preference entries, with each entry having a unique host name or IP address. Refer to the "Specifying Preferences" section of this reference page for information about specifying File Server or FL Server entries.

-path *filename*

Specifies the full pathname of a file from which the command is to read preference entries. Each entry consists of a File Server or FL Server machine specification (a host name or IP address) and its respective rank. Separate each machine specification from its rank with one or more spaces, and include each paired machine specification and rank on a separate line. Refer to the "Specifying Preferences" section of this reference page for information about specifying File Server or FL Server entries.

-stdin

Directs the command to read File Server or FL Server preference entries from standard input (**stdin**). Each entry must consist of a machine specification (either a host name or IP address) and a ranking value. Separate each machine specification and each rank with one or more

spaces. Refer to the "Specifying Preferences" section of this reference page for information about specifying File Server or FL Server entries.

-fldb Directs the command to consider the servers specified in the **-server** option as FL Servers.

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

DESCRIPTION

The **cm setpreferences** command can be used to add preference entries to a Cache Manager preference list or modify ranking values for existing preference entries. The Cache Manager preference list is created each time a Cache Manager is initialized with the **dfsd** command (each time the client machine is rebooted). Each Cache Manager maintains its own separate preference list. Each entry in the list consists of the IP address of an FL Server or File Server and an automatically assigned preference value. New entries are automatically added to the preference list as necessary when filesets are first referenced.

In operation, when the Cache Manager needs to contact an FL Server, it consults its list of FL Servers and attempts to contact a server at the address with the lowest-ranking value in the preference list. Similarly, when a Cache Manager needs to contact a File Server, it consults its preference list and contacts a suitable File Server at the address with the lowest-ranking value.

If the Cache Manager cannot access a server at the address with the lowest preference rank (because of a problem with either the machine or the network), the Cache Manager attempts to access a similar server at the address with the next-lowest rank. It continues in this way until it either succeeds in accessing an appropriate server or determines that all such servers are unavailable.

By default, the Cache Manager assigns preferences that make sensible choices based on the location of servers. Therefore, you should adjust the default values only if there is a compelling reason to do so. The default values force the Cache Manager to attempt to connect to servers in the following order:

1. The same machine as the client (default rank of 5000).
2. The same subnetwork as the client (default rank of 20000).
3. The same network as the client (default rank of 30000).
4. Different networks (default rank of 40000).

For example, a server on the same machine as the Cache Manager receives a rank of 5000, while a server on the same subnetwork receives a rank of 20000. The entry with the lowest-ranking value has the highest "preference." Thus, a server with a preference value of 5000 will be chosen before a server with a rank of 20000.

Should two servers be assigned the same preference value, such as two File Servers on the same subnetwork both receiving a default value of 20000, the server with the lowest round-trip value is chosen. Each server is assigned a random round-trip value when the Cache Manager is initialized. The assigned round-trip value is always higher than the upper bound for stored actual round-trip values. This ensures that an actual round-trip value will always be chosen over assigned values.

The Cache Manager stores its preferences in the kernel of the local machine. The preferences are lost each time the Cache Manager is initialized. You can include the **cm setpreferences** command in a machine's initialization file to load a predefined collection of server preferences when the machine is rebooted.

Specifying Preferences

Using the **cm setpreferences** command, you specify Cache Manager preference entries as pairs of values. The first value of the pair is the machine specification (either the host name or IP address in dotted decimal format) of a File Server or FL Server; the second value is the preference rank (an integer in the range from 1 to 65,534). The FLDB can contain up to four addresses for each server machine (although the machine can have more connections); therefore, the Cache Manager preference list will normally have up to four entries for a given server machine.

You can specify preference entries

- On the command line via the **-server** option. Use this option to tune the preferences manually in response to system or network adjustments.
- From a file via the **-path** option. Use this option to configure one or more Cache Managers with a fixed set of preferences. You can use the **cm getpreferences** command to generate a file of preferences that has the proper format.
- From standard input via the **-stdin** option. Use this option to pipe preferences to the command from a user-defined process that generates preferences in an acceptable format.

The **-server**, **-path**, and **-stdin** options are not mutually exclusive. You can include any combination of these options with the command to provide input from multiple sources. Note that the command does not verify host names or IP addresses specified with any of its options. You can add a preference for an invalid host name or IP

address; the Cache Manager stores invalid preferences in the kernel, but it ignores them (the Cache Manager never needs to consult such preferences).

Privilege Required

The issuer must be logged in as **root** on the local machine.

Output

By default, the **cm setpreferences** command displays no output.

Examples

The following command uses the **-server** option to set the Cache Manager's preferences for the File Servers named **fs3.abc.com** and **fs4.abc.com**, the latter of which is specified by IP address. The two File Servers reside in a different subnetwork that is in the same network as the local machine. Therefore, the Cache Manager assigned each a default rank of 30,000. To make the Cache Manager prefer these File Servers over File Servers in other subnetworks, the **cm setpreferences** command is used to assign these machines ranks of 25,000.

```
# cm setp -se fs3.abc.com 25000 128.21.18.100 25000
```

The following command uses the **-server** option to set the Cache Manager's preferences for the same two File Servers, but it also uses the **-path** option to read a collection of preferences from a file that resides on the local machine at **/etc/cm.prefs**:

```
# cm setp -se fs3.abc.com 25000 128.21.18.100 25000 -p
/etc/cm.prefs
```

The file **/etc/cm.prefs** has the following contents and format:

```
128.21.16.214 7500
128.21.16.212 7500
121.86.33.41 39000
```

```
121.86.33.34 39000
121.86.33.36 41000
121.86.33.37 41000
```

The following command uses the **-stdin** option to read preferences from standard input. The preferences are piped to the command from a program, **calc_prefs**, which was written by the issuer to calculate preferences based on values significant to the local cell.

```
# calc_prefs | cm setp -stdin
```

Related Information

Commands: **cm getpreferences(8dfs)**, **dfsd(8dfs)**.

cm setprotectlevels

Purpose

cm setprotectlevels – Adjusts DCE remote procedure call (RPC) authentication levels for communications between the Cache Manager and File Servers

Synopsis

cm setprotectlevels [**-initiallocalprotectlevel** *level*] [**-minlocalprotectlevel** *level*] [**-initialremoteprotectlevel** *level*] [**-minremoteprotectlevel** *level*] [**-help**]

Options

-initiallocalprotectlevel *level*

Specifies the initial DCE RPC authentication level for communications between the Cache Manager and File Servers within the same cell. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-minlocalprotectlevel *level*

Specifies the minimum acceptable DCE RPC authentication level for communications between the Cache Manager and File Servers within the same cell. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-initialremoteprotectlevel *level*

Specifies the initial DCE RPC authentication level for communications between the Cache Manager and File Servers within foreign cells The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-minremoteprotectlevel *level*
Specifies the minimum acceptable DCE RPC authentication level for communications between the Cache Manager and File Servers within foreign cells. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

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

Description

The **cm setprotectlevels** command adjusts the DCE RPC security level for RPCs sent between a Cache Manager and DFS File Servers. The command adjusts two levels: an initial DCE RPC security level used as a starting point in security level negotiations between the Cache Manager and a File Server and the minimum DCE RPC security level the Cache Manager will accept for such communications. Two sets of these levels are maintained: one set specifies the security levels for communications with File Servers within the local cell and the other set specifies the security levels for communications with File Servers within foreign cells. Both sets of security levels are initially set through the **dfsd** command.

In operation, the Cache Manager and File Server interact to arrive at a mutually acceptable authentication level for communications. The negotiation starts with an RPC using the initial authentication level sent from the Cache Manager to the File Server. If the initial authentication level is outside the minimum or maximum bounds set at the File Server, the File Server returns a response to the Cache Manager specifying that the authentication level is either too low or too high. The Cache Manager then decreases or increases its authentication level accordingly and retries the RPC. This process continues until the Cache Manager either adjusts its RPCs to an acceptable security level or the File Server requests a security level below the minimum set at the Cache Manager (causing the Cache Manager to refuse communications with the File Server). Once the Cache Manager and File Server have negotiated a security level, the Cache Manager stores this information so that it does not need to renegotiate this level for further communications with the File Server.

The authentication bounds for communications at the File Server itself is set through the **fxd** command. The Cache Manager and **fxd** default settings are such that communications occur at the Packet Integrity authentication level.

In addition to a general pair of upper and lower bounds for all communications between the File Server and Cache Manager, administrators can also set advisory bounds on a per fileset basis. At present, these advisory levels serve only to bias the Cache Manager's selection of an initial authentication level (they may be enforced in a future version of DFS). Advisory bounds are set through the **fts setprotectlevels** command and are stored in the FLDB record for that fileset.

Note that the use of this command does not preclude communications with File Servers running earlier versions of DFS.

The various authentication levels are set by specifying either an integer value between 0 and 6, a complete string specifying the authentication level, or an abbreviation of that string as the *level* argument for the various command options. The following lists the various authentication levels:

- **0** or **default** or **rpc_protect_level_default**: Use the DCE default authentication level.
- **1** or **none** or **rpc_protect_level_none**: Perform no authentication.
- **2** or **connect** or **rpc_protect_level_connect**: Authenticate only when the Cache Manager establishes a connection with the File Server.
- **3** or **call** or **rpc_protect_level_call**: Authenticate only at the beginning of each RPC received.
- **4** or **pkt** or **rpc_protect_level_pkt**: Ensure that all data received is from the expected host.
- **5** or **pkt_integrity** or **rpc_protect_level_pkt_integrity**: Authenticate and verify that none of the data transferred has been modified.
- **6** or **pkt_privacy or rpc_protect_level_pkt_privacy**: Perform authentication as specified by all of the previous levels and also encrypt each RPC argument value.

Note that there is a trade-off between selecting higher security and performance. The higher levels of security require more overhead and increase the response time in file operations with File Servers.

Privilege Required

The issuer must be logged in as **root** on the local machine.

Examples

The following command sets the following authentication values:

- The initial authentication level for communications with File Servers in the local cell is set to packet integrity.
- The minimum authentication level for communications with File Servers in the local cell is set to packet.
- The initial authentication level for communications with File Servers in foreign cells is set to packet privacy.
- The minimum authentication level for communications with File Servers in foreign cells is set to packet privacy.

```
$ cm setprotectlevels -initiallocalprotectlevel 5 -minlocalprotectlevel 4
-initialremoteprotectlevel 6 -minremoteprotectlevel 6
```

Related Information

Commands: **cm getprotectlevels(8dfs)**, **fxd(8dfs)**, **dfsd(8dfs)**, **fts setprotectlevels(8dfs)**

cm setsetuid

Purpose

cm setsetuid – Enables or disables **setuid** programs from specified filesets

Synopsis

cm setsetuid [**-path** {*filename* | *directory_name*}…] [**-state** {on | off}] [**-help**]

Options

-path {*filename* | *directory_name*}
Names a file or directory from each fileset whose **setuid** status is to be changed. If this option is omitted, the status is changed for the fileset containing the current working directory.

-state Allows or disallows **setuid** programs from the filesets indicated with **-path** to execute with **setuid** permission. Specify **on** with this option to allow **setuid** programs from the indicated filesets to execute with **setuid** permission; specify **off** with this option to disallow **setuid** programs from the indicated filesets to execute with **setuid** permission. If this option is omitted, **setuid** programs from the filesets are allowed to execute with **setuid** permission. (The command has no effect if **setuid** permission was already enabled.)

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

Description

The **cm setsetuid** command enables **setuid** programs from the indicated filesets to execute with **setuid** permission or prevents them from executing with **setuid** permission. Indicate each fileset whose **setuid** permission is to be enabled or disabled by specifying the name of a file or directory in the fileset with the **-path** option. The permissions are enabled or disabled on a per-fileset and per-Cache Manager basis. This command is commonly included in a start-up file (**/etc/rc** or its equivalent) to enable **setuid** programs at machine startup.

If **on** is specified with the **-state** option, or if the **-state** option is omitted, the Cache Manager allows **setuid** programs from the indicated filesets to execute with **setuid** permission. If **off** is specified with the **-state** option, the Cache Manager does not allow **setuid** programs from the indicated filesets to execute with **setuid** permission. By default, the Cache Manager does not allow **setuid** programs from a fileset to execute with **setuid** permission.

A **setuid** program is indicated by setting a mode bit associated with an executable file. While a **setuid** program executes, the person executing the program is treated as if he or she is the owner of the program. The effective user identification number (UID) of the executing program is the UID of the person who owns the program, not the UID of the person who initiated the program's execution. Thus, the person executing the program is granted the same permissions as the person who owns the program for the duration of the program's execution.

Note that **setuid** programs are effective only in the local environment. A **setuid** program can change only the local identity under which a program runs; it cannot change the DCE identity with which a program executes because it provides no Kerberos tickets. DCE does not recognize the change to the local identity associated with a **setuid** program.

The **cm setsetuid** command enables or disables **setgid** programs from the indicated filesets at the same time that it changes the status of **setuid** programs. The **cm getsetuid** command displays whether the Cache Manager allows **setuid** and **setgid** programs from indicated filesets to execute.

Privilege Required

The issuer must be logged in as **root** on the local machine.

Examples

The following command enables **setuid** and **setgid** programs that reside on the fileset containing the directory */.../abc.com/fs/usr/jlw*:

```
# cm setsetuid /.../abc.com/fs/usr/jlw
```

Related Information

Commands: **cm getsetuid(8dfs)**.

cm statservers

Purpose

cm statservers – Checks the statuses of File Server machines

Synopsis

cm statservers [{**-cell** *cellname* | **-all** }] [**-fast**][**-help**]

Options

-cell *cellname*
: Specifies the name of the specific cell the Cache Manager is to probe for the status of each File Server machine it has contacted or has attempted to contact from that cell. The Cache Manager probes only machines in the specified cell. Use this option or use the **-all** option; omit both options to direct the Cache Manager to probe only machines in the local cell.

-all
: Directs the Cache Manager to probe all of the machines it has contacted in all cells. Use this option or use the **-cell** option; omit both options to direct the Cache Manager to probe only machines in the local cell.

-fast
: Directs the Cache Manager to display its current list of contacted File Server machines without probing the machines. This option can be combined with the **-cell** or **-all** option; it can also be used if both the **-cell** and **-all** options are omitted.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **cm statservers** command lists all File Server machines in the indicated cells that meet the following two conditions:

- The Cache Manager has been in contact with the File Exporter running on the machine and needs to contact it in the future (probably because the Cache Manager is holding tokens for data on that File Server machine).

- The File Exporter on the machine is not currently responding to the Cache Manager's probes (implying that it is not responding to the Cache Manager's requests for data either).

The Cache Manager maintains a list of File Server machines that meet the first condition, updating the list periodically by attempting to contact the File Exporter on each machine in the list. When a machine does not respond to a probe, the Cache Manager marks it as nonfunctioning. If a machine that previously did not respond begins to respond again, the Cache Manager erases the mark. The Cache Manager maintains this information in the kernel of the local machine.

Without the **-fast** option, this command forces the Cache Manager to update its information immediately (rather than waiting the standard interval). The Cache Manager probes the File Exporters on the machines in the specified cells, records those that do not respond, and reports the results. If you include the **-fast** option, the Cache Manager displays the list of nonfunctioning machines that it has at the time the command is issued; it does not probe the machines again.

By default, the Cache Manager probes machines in the local cell only. If the **-all** option is used, the Cache Manager probes all machines (from all cells) that meet the first condition. If a *cellname* is specified with the **-cell** option, the Cache Manager probes only the machines in that cell.

The execution of this command can be lengthy if a number of machines in the Cache Manager's list are unresponsive when the command is issued. The Cache Manager waits a standard timeout period before concluding that a File Exporter is not responding; this allows for the possibility of slow cross-network communication. If it is important that the command shell prompt return quickly, run this command in the background. It is harmless to interrupt the command (with **<Ctrl-c>** or another interrupt signal).

This command does not check the statuses of all File Server machines in a cell. The Cache Manager probes only those machines that meet the first condition in the previous list.

Privilege Required

No privileges are required.

Output

If the Cache Manager gets a response from all of the machines that it probes (that is, all such machines are functioning normally), the command displays the following output:

```
All servers are running.
```

This message does not imply that all File Server machines in the specified cells are running; it implies only that those machines that the Cache Manager probed are running.

If one or more machines fail to respond to the Cache Manager's probes within the timeout period, the command displays the following output:

```
These servers are still down: hostname
```

where *hostname* is the name of each File Server machine that fails to respond.

In a multihomed server environment (a File Server machine can have four IP addresses listed in the Cache Manager's preferences), the *hostname* corresponds to the host name or IP address that the Cache Manager is currently using to access each File Server machine. The output does not contain multiple machine names for the same File Server machine.

Examples

The following command uses the **-fast** option to view the Cache Manager's current list of unresponsive machines belonging to the local cell rather than waiting for the Cache Manager to probe them again. The output indicates that all machines responded to the most recent probes.

```
$ cm statservers -f
```

```
All servers are running.
```

The following command checks all File Server machines from which the Cache Manager has cached data, regardless of the cell in which a machine resides. The command reports that the machines named **fs1.abc.com** and **fs3.state.edu** did not respond to the Cache Manager's probes. The **&** (ampersand) is used to execute the command in the background.

$ cm statservers -all &

```
These servers are still down: fs1.abc.com fs3.state.edu
```

Related Information

Commands: **cm lsstores(8dfs)**, **cm whereis(8dfs)**.

cm sysname

Purpose

cm sysname – Reports or sets the CPU/OS type

Synopsis

cm sysname [**-newsys** *sysname*] [**-help**]

Options

-newsys *sysname*
: Specifies the new setting of the CPU/Operating System (**@sys**) variable for the machine on which it is issued. If this option is omitted, the output shows the current setting of the variable.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **cm sysname** command displays the current setting of the **@sys** variable or sets the variable on a client machine. If the **-newsys** option is omitted, the command reports the current setting of the **@sys** variable. If the **-newsys** option is included, the command sets the variable to the specified CPU/OS type. The value of the variable is displayed from or set in the kernel of the client machine on which the command is issued.

The Cache Manager's main use of the **@sys** variable is in pathnames used in symbolic links. As the Cache Manager interprets pathnames, it substitutes the value of the indicator for any occurrence of **@sys**. (Use the **@sys** variable sparingly; it can make the effect of changing directories confusing.)

Privilege Required

To view the current setting of **@sys** (without the **-newsys** option), no privileges are required. To change the setting of **@sys** (with the **-newsys** option), you must be logged in as **root** on the local machine.

Output

If the **-newsys** option is not specified, the output reports the system type in the following format:

```
Current sysname is ‘system_type’.
```

Examples

The following command shows the output produced on a machine running OSF/1:

```
$ cm sys
```

```
Current sysname is ‘pmax_osf1’.
```

The following commands set the system type on a machine running AIX 3.2 and use it in a symbolic link from the **/usr/local** directory on the local machine to a directory in the DFS filespace:

```
# cm sys -new rs_aix32
# ln -s /.../abc.com/fs/@sys/usr/local /usr/local
# ls -l /usr/local
```

```
lrwxrwxrwx 1 root 34 May 31 1993 /usr/local ->
/.../abc.com/fs/@sys/usr/local
```

```
# cd /usr/local
# pwd
```

```
/.../abc.com/fs/rs_aix32/usr/local
```

cm whereis

Purpose

cm whereis – Reports names of File Server machines that house specified files or directories

Synopsis

cm whereis [**-path** {*filename* | *directory_name*}…] [**-help**]

Options

-path *filename* or *directory_name*
: Specifies the pathname of each file or directory whose location is to be reported. Each file or directory must reside in DFS, not on a local disk. If a full pathname is not provided, the file or directory is assumed to reside in the current working directory. If this option is omitted, the current working directory is used.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **cm whereis** command displays information about the location of each file or directory indicated with the **-path** option. The command reports the name of the cell in which the file or directory exists, the name of the fileset in which it resides, and the name of each File Server machine that houses a copy of the fileset. This information comes from the kernel of the workstation on which the command is issued.

Privilege Required

No privileges are required.

Output

The output includes a separate line displaying the following information about each file or directory specified with the **-path** option:

```
File ‘filename’ resides in the cell ‘cellname’ in fileset
‘fileset_name’ on host(s) ‘hostname’.
```

where:

filename Specifies the complete pathname of a file or directory specified with the **-path** option.

cellname Specifies the name of the cell in which the file or directory is located.

fileset_name Specifies the name of the fileset in which the file or directory is located.

hostname Specifies the name of the File Server machine on which the fileset is located. If the fileset is a read/write or backup fileset, only one machine name is displayed; if the fileset is a read-only fileset, multiple machine names can be displayed. However, only one machine name is displayed for each File Server machine. (The Cache Manager can have up to four preferences for each File Server machine, with each preference having a different host name or IP address.)

Examples

The following command indicates that the directory named */.../abc.com/fs/bin/sysfile* is located in a replicated fileset on the File Server machines named **fs1**, **fs3**, and **fs6**, all of which are located in the cell named *abc.com*:

```
$ cm whereis /.../abc.com/fs/bin/sysfile
```

```
File ‘/.../abc.com/fs/bin/sysfile’ resides in the cell ‘abc.com’,
in fileset ‘sysfile.bin’, on hosts fs1.abc.com, fs3.abc.com,
fs6.abc.com.
```

Related Information

Commands: **cm statservers(8dfs)**.

dfs_login

Purpose Authenticates a user to DCE for access to DFS via the DFS/NFS Secure Gateway

Synopsis **dfs_login** [**- h** *hostname*] [**- S** *sysname*] [**- l** *hh* [*:mm*]] [*dce_principal*] [*dce_password*]

Options

-h *hostname*
Specifies the host name of a Gateway Server machine (a machine that is running the **dfsgwd** process) on which the DCE credentials of the specified user are to be stored. By default, the command uses the host name of the Gateway Server machine that exports the root of the DCE namespace, **/...**, to the NFS client. Use this option to name a different Gateway Server machine.

-S *sysname* Specifies the system name of the NFS client for the principal performing the login. The default system name can be overriden through the use of the **DFS_SYSNAME** variable or the **-S** option. The **-S** option takes precedence. The *sysname* argument is a unique name derived from **uname( )** that describes the machine architecture and OS type, such as **hp700_ux905** or **hp800_ux90**.

-l *hh[:mm]* Specifies the lifetime to be assigned to the DCE ticket-granting ticket (TGT) obtained by the command. Enter the lifetime as a number of hours and, optionally, minutes. For example, enter **4** for 4 hours, or enter **2:30** for 2 hours and 30 minutes. A value specified with this option is subject to the policies in effect in the registry database of the DCE cell. By default, the TGT is assigned the default lifetime assigned to tickets in the DCE cell.

Arguments

dce_principal

Provides the DCE principal name of the user who is to be authenticated to DCE. By default, the command uses the name of the user who issues the command.

dce_password

Provides the DCE password of the user indicated with the *dce_principal* argument. If you do not specify a password, the command prompts for a password if one of the following is true: You name a user other than yourself; you name yourself and you do not already have a valid TGT in the current login context; or you do not name a user and you do not already have a valid TGT in the current login context. The command does not prompt for a password if you do not name a different user and you already have a valid TGT. The command's interactive prompt provides for secure entry of the password.

Description

The **dfs_login** command authenticates a user to DCE from an NFS client. The command establishes DCE credentials for the user named with the *dce_principal* argument. If no user is specified, the command obtains credentials for the user who issues the command.

The command obtains a TGT for the user from the DCE Security Service. To obtain a TGT, a user must have a valid account in the registry database of the DCE cell. The TGT is used to create a valid login context for the user. The login context includes a Process Activation Group (PAG), which DFS stores in the kernel of the Gateway Server machine to identify the user's TGT. The TGT serves as the user's DCE credentials to provide authenticated access to files and directories in the DFS filespace from the NFS client on which the command is issued.

The **dfs_login** command also adds an entry for the user to the authentication table (AT) on the Gateway Server machine. The entry is a mapping that pairs the user's UNIX user identification number (UID) and the network address of the NFS client for which the user has DCE credentials with the user's PAG. Each Gateway Server machine maintains its own authentication table, so the DCE credentials are valid only for access via the Gateway Server machine on which they are stored. The credentials are also valid only for the NFS client from which the command is issued.

To obtain authenticated access to DCE from a different NFS client, a user must issue the command from that client.

The command does not obtain a new TGT if you do not name a user other than yourself on the command line and you already have a valid TGT in the current login context. If you do not already have an entry in the authentication table for the NFS client from which you issue the command, the command uses your existing PAG to create a new entry for you. If you already have an entry in the authentication table for the NFS client, the command has no effect. In either case, the command does not affect existing entries in the authentication table, and it does not affect the remaining ticket lifetime of your existing TGT.

The **dfs_login** command provides essentially the same functionality as the **dfsgw add** command, with the exception that the **dfs_login** command lets you request a specific ticket lifetime. Use the **dfs_logout** command (or the **dfsgw delete** command) to end an authenticated session by removing an entry from the authentication table. Both the **dfs_login** and **dfs_logout** commands require a working Kerberos 5 environment on the NFS client from which they are issued. See Part 1 of this manual for information about configuring an NFS client for use with the DFS/NFS Secure Gateway.

Privileges Required

No privileges are required.

Output

The **dfs_login** command displays the following prompt to request a DCE password:

```
Password for dce_principal: dce_password
```

where *dce_principal* is the name of the DCE principal for whom credentials are to be established, and you enter *dce_password* as the DCE password for the named user. The command displays this prompt only if you do not specify a password on the command line and if either of the following is true:

- You name a user other than yourself on the command line
- You do not name a user other than yourself on the command line and you do not already have a valid TGT

If the login succeeds, the command returns no further messages.

Files

/krb5/krb.conf
A Kerberos configuration file. The **dfs_login** command reads this file to determine the name of a DCE security server to contact.

/krb5/krb.realms
A Kerberos configuration file. The Kerberos runtime uses the information in this file to translate Internet domains to the corresponding Kerberos realms.

Variables

DFSGWSERVICE
An environment variable that can be set to specify the name of the DFS/NFS Secure Gateway service if the name of the service is changed to something other than **dfsgw**. The named service provides the login facility for the DFS/NFS Secure Gateway. The **dfs_login** command uses the service to look up the port on the Gateway Server machine at which the **dfsgwd** process is listening.

Notes

The **dfs_login** command uses the syntax conventions of all DCE commands, but it does not provide the shortcuts and help available with other DFS commands. When specifying options, you must enter the name of each option in full (you cannot abbreviate the names of options), and each option must be followed by an argument specified for it (you cannot omit options). Also, the command does not include a **-help** option.

Examples

The following command, issued on a properly configured NFS client, establishes DCE credentials for the user named **ludwig**. In the example, the DCE password of the user **ludwig** is **beethoven** .

```
dfs_login ludwig
Password for ludwig@abc.com: beethoven
```

Exit Values

The **dfs_login** command returns an exit value of **0** (zero) if it adds an entry for the user to the authentication table. Otherwise, it returns a nonzero exit value.

Related Information

Commands: **dfs_logout(8dfs)**, **dfsgw add(8dfs)**, **dfsgw delete(8dfs)**, **dfsgwd(8dfs)**.

dfs_logout

Purpose

Cancels a user's authenticated access to DFS via the DFS/NFS Secure Gateway

Synopsis

dfs_logout [**- h** *hostname*] [*dce_principal*]

Options

-h *hostname* Specifies the hostname of the Gateway Server machine (a machine that is running the **dfsgwd** process) from which the user's entry in the authentication table (AT) is to be removed. By default, the command removes the entry from the authentication table on the Gateway Server machine that exports the root of the DCE namespace, **/...**, to the NFS client. Use this option to name a different Gateway Server machine.

Arguments

dce_principal

Provides the DCE principal name of the user whose entry in the authentication table is to be removed. By default, the command removes the entry of the user who issues the command.

Description

The **dfs_logout** command cancels a user's authenticated access to DFS from an NFS client. The command ends the authenticated session of the user named with the *dce_principal* argument. If no user is specified, the command ends the session of the user who issues the command. Once the command completes, the user no longer has authenticated access to DFS from the NFS client.

The **dfs_logout** command removes the user's entry from the authentication table on the specified Gateway Server machine. The command removes the user's entry for the NFS client from which the command is issued. The command has no effect on entries

the user may have in the authentication table for other NFS clients. It also has no effect on entries the user may have in authentication tables on other Gateway Server machines.

The **dfs_logout** command provides the same functionality as the **dfsgw delete** command. To acquire DCE credentials for authenticated access to DFS from an NFS client and create an entry in the authentication table, users issue the **dfs_login** command (or the **dfsgw add** command).

Both the **dfs_logout** and **dfs_login** commands require a working Kerberos 5 environment on the NFS client from which they are issued. See Part 1 of this manual for information about configuring an NFS client for use with the DFS/NFS Secure Gateway.

Privilege Required

The issuer must be either the user whose entry is to be removed from the authentication table or a user who is logged into the local machine as **root**.

Output

If it succeeds, the **dfs_logout** command returns no messages.

Files

/krb5/krb.conf
: A Kerberos configuration file. The **dfs_logout** command reads this file to determine the name of a DCE Security Server.

/krb5/krb.realms
: A Kerberos configuration file. The Kerberos runtime uses the information in this file to translate Internet domains to the corresponding Kerberos realms.

Variables

DFSGWSERVICE
: An environment variable that can be set to specify the name of the DFS/NFS Secure Gateway service if the name of the service is changed to something other than **dfsgw**. The named service provides the login

facility for the DFS/NFS Secure Gateway. The **dfs_logout** command uses the service to look up the port on the Gateway Server machine at which the **dfsgwd** process is listening.

Notes

The **dfs_logout** command uses the syntax conventions of all DCE commands, but it does not provide the shortcuts and help available with other DFS commands. When specifying options, you must enter the name of each option in full (you cannot abbreviate the names of options), and each option must precede an argument specified for it (you cannot omit options). Also, the command does not include a **-help** option.

Examples

The following command cancels authenticated access to DFS for the user who issues it:

```
$ dfs_logout
```

Exit Values

The **dfs_logout** command returns an exit value of **0** (zero) if it removes the entry for the specified user from the authentication table. Otherwise, it returns a nonzero exit value.

Related Information

Commands: **dfsgw add(8dfs)**, **dfsgw delete(8dfs)**, **dfs_login(8dfs)**, **dfsgwd(8dfs)**.

dfsbind

Purpose

Provides user-space information to the Cache Manager and File Exporter

Synopsis

dfsbind [**-expressprocs** *number_of_express_daemons*] [**-regularprocs** *number_of_regular_daemons*] [**-junctionlife** *seconds_to_live*] [**-prefixlife** *seconds_to_live*] [**-notfoundlife** *seconds_to_live*] [**-debug**][**-help**]

Options

-expressprocs *number_of_express_daemons*

Specifies the number of express processes (user-space threads) allocated to handling requests for security information that do not require a substantial amount of time. By default, **dfsbind** uses one express process. Use this option to increase the number of express processes if the local machine encounters a large number of timeout errors. Specify an integer greater than 0 (zero) to indicate the number of express processes.

-regularprocs *number_of_regular_daemons*

Specifies the number of regular processes (user-space threads) allocated to handling requests for CDS pathname resolution and requests for security information that may require significant time. By default, **dfsbind** uses one regular process. Use this option to increase the number of regular processes if the local machine experiences a large number of timeout errors. Specify an integer greater than 0 (zero) to indicate the number of regular processes.

-junctionlife *seconds_to_live*

Specifies the length of time for which information cached about Fileset Database machines for a cell remains valid. When **dfsbind** retrieves this information from the DFS junction of a cell, it sends the information, along with a *time to live* (TTL), to the Cache Manager. The TTL specifies the length of time for which the Cache Manager is to consider the information valid. The Cache Manager caches the information and the TTL. It continues to recognize the information as valid until the TTL

expires, after which it asks **dfsbind** to refresh the information the next time it needs it.

By default, **dfsbind** assigns a TTL of 24 hours to information about Fileset Database machines. This option can be used to change the TTL that **dfsbind** assigns to such information. Specify an integer greater than or equal to 30 to indicate the new TTL in seconds.

Note: *This option has an effect only on DFS client machines, where it is useful primarily for debugging purposes.*

-prefixlife *seconds_to_live*

Specifies the length of time for which information cached about a pathname that is a valid DFS junction name prefix remains valid. When **dfsbind** successfully traverses a given path but the path is not a DFS junction name, it sends the Cache Manager the valid pathname along with a TTL. The Cache Manager caches the information and the TTL, continuing to recognize the information as valid until the TTL expires; it then contacts **dfsbind** to refresh the information the next time it needs it.

By default, **dfsbind** assigns a TTL of 24 hours to information about pathnames that are valid DFS junction name prefixes. This option can be used to change the TTL that **dfsbind** assigns to such information. Specify an integer greater than or equal to 30 to indicate the new TTL in seconds.

Note: *This option has an effect only on DFS client machines, where it is useful primarily for debugging purposes.*

-notfoundlife *seconds_to_live*

Specifies the length of time for which information cached about an invalid pathname remains valid. When **dfsbind** cannot traverse a given path, it sends the Cache Manager the invalid pathname along with a TTL. The Cache Manager caches the information and the TTL, considering the information valid until the TTL expires; it then contacts **dfsbind** to refresh the information the next time it needs it.

By default, **dfsbind** assigns a TTL of 1 hour to information about invalid pathnames. This option can be used to change the TTL that **dfsbind** assigns to such information. Specify an integer greater than or equal to 30 to indicate the new TTL in seconds.

Note: *This option has an effect only on DFS client machines, where it is useful primarily for debugging purposes.*

-debug Provides debugging information about the execution of the command. The primary usage of the information is to ensure that the process is executing properly. If this option is specified, the process does not automatically place itself in the background once it starts.

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

The **help** and **apropos** commands available with all command suites are also available with **dfsbind**. See the **bos_help(8dfs)** and **bos_apropos(8dfs)** reference pages for examples using these commands.

Description

The **dfsbind** command starts the **dfsbind** process, which provides user-space services to the Cache Manager on a DFS client machine or the File Exporter on a DFS File Server machine. (The Cache Manager and the File Exporter reside in the kernels of their respective machines.) The binary file for the **dfsbind** command resides in *dcelocal* **/bin/dfsbind**. By default, the process automatically places itself in the background after it starts.

The **dfsbind** process must be run on all client machines and File Server machines. A machine that runs the Cache Manager (which is initialized by the **dfsd** command) and the **dfsbind** process is considered a DFS client machine. A machine that runs the Fileset Server (**ftserver** process), the File Exporter (which is initialized by the **fxd** command), and the **dfsbind** process is considered a DFS File Server machine.

On either type of machine, the **dfsbind** command is usually added to the proper start-up file (**/etc/rc** or its equivalent) rather than entered at the command shell prompt. On a client machine, the **dfsbind** process must be run before the **dfsd** process in a start-up file; on a File Server machine, it must be run before the **fxd** process in a start-up file.

On a client machine, the **dfsbind** process performs the following services:

- It contacts CDS to resolve DCE pathnames (both local and foreign) that it receives from the Cache Manager. When a user on a client machine requests data that the Cache Manager has not cached, the Cache Manager employs **dfsbind** to resolve the pathname of the data. It sends **dfsbind** each element of the pathname in succession, appending each new element to the preceding elements when it

sends it—for example, it first sends **/.../** *element_one*, then **/.../** *element_one*/*element_two*, and so on. In turn, **dfsbind** determines whether each successive pathname is valid.

If the pathname of the data is valid, it eventually contains a DFS junction from which **dfsbind** can access information about the Fileset Database machines for the cell in which the data resides. If it encounters a junction for the DFS filespace, **dfsbind** returns information about the names and network addresses of the Fileset Database machines for the cell to the Cache Manager. (It actually decomposes binding handles to learn this information.)

The Cache Manager uses the information from **dfsbind** to create an RPC binding that it employs to communicate with a Fileset Location (FL) Server on an appropriate Fileset Database machine. The FL Server examines the FLDB and tells the Cache Manager which File Server machine houses the fileset that contains the data requested by the user.

For each successive pathname that it attempts to resolve for the Cache Manager, the **dfsbind** process returns one of the following error codes to the Cache Manager to indicate the result of the resolution operation:

0 (zero) Indicates that the pathname corresponds to a DFS junction that contains information about the Fileset Database machines in the cell. The process sends information about the Fileset Database machines to the Cache Manager.

EISDIR Indicates that the pathname is a valid DFS junction name prefix but is not itself a DFS junction. The process returns the valid pathname to the Cache Manager.

ENOENT Indicates that the given path could not be traversed. The process returns the invalid pathname to the Cache Manager.

ETIMEDOUT
Indicates that unexpected errors occurred. The process returns only the error code to the Cache Manager.

DCE pathname and DFS junction information that the Cache Manager receives from **dfsbind** is valid for a limited amount of time. The **dfsbind** process associates a TTL with all information it sends to the Cache Manager. The TTL defines the amount of time for which the Cache Manager is to consider the information valid. The Cache Manager caches the TTL with the information. Once its TTL has elapsed, the information becomes stale; the Cache Manager contacts **dfsbind** to refresh the information the next time it needs it.

The **dfsbind** process associates the TTLs with the information it passes to the Cache Manager as follows:

— Information about Fileset Database machines (error code **0**) receives a TTL of 24 hours by default. (The TTL of such information can be modified with the **dfsbind** command's **-junctionlife** option.)

— Information about valid DFS junction name prefixes (error code **EISDIR**) has a TTL of 24 hours by default. (The TTL of this type of information can be changed with the command's **-prefixlife** option.)

— Information about invalid pathnames (error code **ENOENT**) has a TTL of 1 hour by default. (The TTL of this type of information can be altered with the command's **-notfoundlife** option.)

For example, when the Cache Manager first needs to access data from a fileset in the local cell, it passes each successive element of the DCE pathname of the data to **dfsbind**. If the path contains a DFS junction name, **dfsbind** eventually returns information about the local cell's Fileset Database machines, and a TTL that it assigns to the information, to the Cache Manager. The Cache Manager caches the information and the TTL, using the information to contact a Fileset Database machine in the cell. If the Cache Manager needs to access data from a fileset in the local cell before the TTL has elapsed, it uses the cached information to contact a Fileset Database machine in the cell. However, if it needs to access data from a fileset in the local cell after the TTL has elapsed, it again contacts **dfsbind** to refresh its knowledge of local Fileset Database machines.

- It obtains user authentication information for the kernel RPC runtime. It communicates with the DCE Security Service of the appropriate cell to obtain authentication information about users of the client machine.

  The Cache Manager communicates with the kernel RPC runtime when it needs to create an RPC binding to a File Server machine on behalf of a user. The kernel RPC runtime then communicates with **dfsbind** to obtain authentication information about the user for use in the binding. The **dfsbind** process obtains the authentication information from the security server and sends it back to the kernel RPC runtime, which packages the information along with the other information from the Cache Manager into the RPC binding and sends it to the appropriate File Server machine.

On a File Server machine, the **dfsbind** process simply maintains user authentication information required by the File Exporter on the machine. The File Exporter uses this information to ensure that only authenticated users access data from the machine.

The command's **-expressprocs** and **-regularprocs** options can be used to change the default number of processes **dfsbind** runs on a machine as follows:

- The **-expressprocs** option specifies the number of express processes that **dfsbind** allocates for the handling of requests that require little time to complete. For example, express processes service requests for information from the local security service. The **dfsbind** process can typically handle these types of requests more quickly than it can those assigned to regular processes.
- The **-regularprocs** option specifies the number of regular processes that **dfsbind** allocates for the handling of requests that may require a substantial amount of time to complete. For example, regular processes service requests for the resolution of DCE pathnames and for information from the security service of a foreign cell. The **dfsbind** process typically requires more time to handle these types of requests than it does to handle requests assigned to express processes.

Employing two types of processes allows **dfsbind** to function more efficiently. Requests are assigned to processes according to the amount of time they require to complete. Thus, requests with short turnaround times are not queued behind requests with long turnaround times. Increase the number of express and regular daemons on a machine that experiences a large number of timeout (**ETIMEDOUT**) errors. (Note that both express and regular processes run as threads rather than processes, so neither type of process shows up in the output of the **ps** command or its equivalent.)

If the **-debug** option is included with the **dfsbind** command, the process provides debugging information as it executes. The debugging output is in the form of brief messages reporting the action currently being performed. The messages are useful primarily to ensure that the process is executing properly. If the **-debug** option is included with the command, the process does not automatically place itself in the background after it starts.

Privileges Required

The issuer must be **root** on the local machine.

Examples

The following line, entered in the appropriate initialization file (**/etc/rc** or its equivalent) on a client or File Server machine, starts the **dfsbind** process on the local machine. This line must be included before the line that starts the **dfsd** or **fxd** process on a client or File Server machine. The **dfsbind** process in the example uses two express processes and two regular processes.

dfsbind -expressprocs 2 -regularprocs 2

Related Information

Commands: **dfsd(8dfs)**, **fxd(8dfs)**.

dfsd

Purpose

Initializes the DFS Cache Manager and starts related daemons

Synopsis

dfsd [**-blocks** *number_of_cache_blocks*] [**-files** *number_of_cache_files*] [**-stat** *number_of_status_cache_entries*] [**-rootfileset** *root_fileset*] [**-cachedir** *cache_directory*] [**-mountdir** *DFS_mount_directory*] [**-rootcell** *root_cell*] [**-settime**][**-mainprocs** *number_of_background_daemons*] [**-tokenprocs** *number_of_token_daemons*] [**-ioprocs** *number_of_I/O_background_daemons*] [**-memcache**][**-dcache** *number_of_entries*] [**-chunksize** *chunk_exponent*] [**-namecachesize** *number_of_name_cache_entries*] [**-initiallocalprotectlevel** *level*] [**-minlocalprotectlevel** *level*] [**-initialremoteprotectlevel** *level*] [**-minremoteprotectlevel** *level*] [**-verbose**][**-debug**][**-help**]

OPTIONS

-blocks *number_of_cache_blocks*

Specifies the number of kilobytes to be made available for caching in the machine's cache directory (for a disk cache) or memory (for a memory cache). This value overrides the default, which must be specified in the third field of the *dcelocal*/**etc/CacheInfo** file. The unit of measurement for block size is always kilobytes.

A disk cache should not exceed 90% of the disk space available on the cache partition; a memory cache should not exceed 20 to 25% of the machine's available memory. These limits are necessary because the implementation of the cache requires a small amount of disk space or machine memory, and because a memory cache must leave enough memory for processes and applications to run.

For a memory cache, do not combine this option with the **-dcache** option.

Note: The minimum cache size you can specify with the **-blocks** option is 17 kilobytes. If you specify a cache size smaller than 17 kilobytes, the Cache Manager creates a cache of 17 kilobytes.

-files *number_of_cache_files*
Specifies the number of V files (chunks) to be created in the cache directory for a disk cache. This value overrides the default, which is the number of cache blocks divided by 8.

Each V file can accommodate a chunk of data. By default, each chunk can accommodate 64 kilobytes of data. To operate most efficiently, at least 90% of the cache must be in use. Use the **-files** option to increase the number of V files if this is not the case. Do not specify a value greater than 32,000.

Do not combine this option with the **-memcache** option, which is used for memory caching.

Note: The minimum number of V files you can specify with the **-files** option is 2. If you specify a value smaller than 2, the Cache Manager creates a cache with two V files.

-stat *number_of_status_cache_entries*
Specifies the number of entries in the machine's memory for recording status information about DFS files in the cache. The default is **300**.

-rootfileset *root_fileset*
Names the read/write fileset corresponding to the top-level (**root**) directory. This option is generally used for testing purposes only.

-cachedir *cache_directory*
Names the local disk directory to be used as the cache for disk caching. This value overrides the default, which must be specified in the second field of the **CacheInfo** file. The default is *dcelocal***/var/adm/dfs/cache**.

Do not combine this option with the **-memcache** option, which is used for memory caching. With memory caching, the **-cachedir** option, like the second field of the **CacheInfo** file, is ignored.

-mountdir *DFS_mount_directory*
Names the local disk directory where the DCE global namespace is to be mounted. This value overrides the default, which must be specified in the first field of the **CacheInfo** file. The default for a machine with a disk is the global namespace designation (**/...**); if **/...** is not used, symbolic links to the global namespace will not work.

-rootcell *root_cell*
Names the cell that contains the root fileset. This option is generally used for testing purposes only.

-settime Causes the local machine to select a random server machine in the local cell to use as the source of the correct time. If this option is specified, the local machine selects a server machine and checks the time on that machine every 10 minutes. If the time on the local machine differs by more than 2 seconds from the time on the selected server machine, the local machine adjusts its time to match that of the server machine.

For machines running the DCE Distributed Time Service (DTS) or the Network Time Protocol (NTP), it is recommended that the **-settime** option be omitted to prevent the machine from selecting and using two different time standards at once.

-mainprocs *number_of_background_daemons*
Specifies the number of background daemons to run on the machine. These daemons improve efficiency by performing prefetching and background writing of saved data. The default is two.

Increase the number of background daemons if the machine serves more than five users.

-tokenprocs *number_of_token_daemons*
Specifies the number of background daemons dedicated to servicing token revocation RPC requests from File Exporters. The default is two. (Token daemons run in addition to the background daemons associated with the **-mainprocs** option.)

Increase the number of token daemons if users on this machine interact with many File Server machines.

-ioprocs *number_of_I/O_background_daemons*
On a machine running the AIX operating system, specifies the number of background I/O daemons performing I/O operations. I/O daemons move data from disk to memory, and vice versa. The default is five.

On a machine running AIX, increase the number of I/O daemons if many users use the machine. *Use this option only on a machine running AIX*. Because no I/O daemons are used on a machine not running AIX, the option is ignored if it is used on a machine not running AIX.

-memcache Causes **dfsd** to initialize a memory cache rather than a disk cache. If this option is provided, space in memory is allocated for the cache; no disk space is used, even if it is available.

Do not combine this option with the **-files** option (which is used for machines that use disk caching). Also, do not combine this option with the **-cachedir** option; with memory caching, the **-cachedir** option, like the second field of the **CacheInfo** file, is ignored.

-dcache *number_of_entries*

Sets the number of dcache entries in memory; dcache entries store information about cache chunks.

For a disk cache, the *dcelocal* **/var/adm/dfs/cache/CacheItems** file contains one entry for each V file. By default, 100 entries from the **CacheItems** file are duplicated in machine memory; the **-dcache** option overrides the default.

For a memory cache, there is no **CacheItems** file; one dcache entry exists for each cache chunk. The Cache Manager determines the number of dcache entries (cache chunks) by dividing the cache size by the chunk size; the **-dcache** option sets the number of cache chunks. Do not combine this option with the **-blocks** option.

Use of this option with a disk cache is not necessary because it increases performance only marginally. It is not recommended with a memory cache because it requires the issuer to perform additional calculations.

-chunksize *chunk_exponent*

Sets the size of each cache chunk. Provide an integer between 13 and 18 to be used as an exponent of 2. This value overrides the default chunk size, which is 64 kilobytes (2^{16}) for a disk cache and 8 kilobytes (2^{13}) for a memory cache. A value less than 13 or greater than 18 sets the chunk size to the appropriate default for the type of cache in use. The unit of measure for chunk size is always bytes.

It is not recommended that you use this option with the **-dcache** option for a memory cache.

-namecachesize *number_of_name_cache_entries*

Sets the number of entries allocated for the Cache Manager's name lookup cache. Provide an integer greater than 0 (zero); the default number of name cache entries is **256**.

The name lookup cache stores the results obtained from remote directory lookup requests to DFS servers, which allows subsequent lookup requests for the same file or directory to be satisfied on the local DFS client rather than on the remote DFS server. Because name cache entries are recycled when the name lookup cache limit is reached, the ability to satisfy the request locally depends upon the size of the name lookup cache.

-initiallocalprotectlevel *level*
Specifies the initial DCE RPC authentication level for communications between the Cache Manager and File Servers within the same cell. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-minlocalprotectlevel *level*
Specifies the minimum acceptable DCE RPC authentication level for communications between the Cache Manager and File Servers within the same cell. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-initialremoteprotectlevel *level*
Specifies the initial DCE RPC authentication level for communications between the Cache Manager and File Servers within foreign cells. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-minremoteprotectlevel *level*
Specifies the minimum acceptable DCE RPC authentication level for communications between the Cache Manager and File Servers within foreign cells. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-verbose Directs **dfsd** to produce a more detailed trace of its activities than it does by default. The trace is displayed on standard output (**stdout**) unless it is directed elsewhere.

-debug Causes **dfsd** to produce a highly detailed trace of its activities, which can be useful for debugging purposes. The trace is displayed on standard output (**stdout**) unless it is directed elsewhere.

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

The **help** and **apropos** commands available with all command suites are also available with **dfsd**. See the **bos_help(8dfs)** and **bos_apropos(8dfs)** reference pages for examples using these commands.

Description

The **dfsd** process initializes the DFS Cache Manager on a client machine according to the information specified with the options described previously. It must be run on all DFS client machines. It is usually added to the proper start-up file (**/etc/rc** or its equivalent) rather than typed at the command shell prompt. (The **dfsbind** process must be run before the **dfsd** process in a start-up file.) The binary file for the **dfsd** process resides in *dcelocal* **/bin/dfsd**.

Specifically, the **dfsd** process does the following:

- Transfers information about cell membership to kernel memory. This information can be changed only by rebooting and running **dfsd**.
- Determines if the cache is on the local disk or in machine memory. A disk cache is used unless the **-memcache** option is provided. If the **-memcache** option is used, no disk space is used, even if it is available; the Cache Manager maintains all cached data and cache-related information in memory.
- Defines the name of the local disk directory devoted to a disk cache. The second field in the **CacheInfo** file specifies the default directory. If necessary, **dfsd** creates the directory, provided its parent directory exists. Any directory that formerly served as the disk cache is left on the disk.
- Sets the size of the cache. The third field in the **CacheInfo** file specifies the default cache size in kilobytes.

For a disk cache, the value in the **CacheInfo** file is an upper limit that can be increased only with the **-blocks** option; it cannot be increased with the other options available with the **dfsd** process. For a memory cache, the **-dcache** option alone or in combination with the **-chunksize** option overrides the cache size specified in the **CacheInfo** file; these combinations are not recommended.

After initialization, use the **cm setcachesize** command to change the size of a disk cache without rebooting. The value set with the **cm setcachesize** command is overridden the next time the machine is rebooted and **dfsd** is run. The **cm setcachesize** command does not work for memory caches; the machine must be rebooted. (The **cm getcachesize** command can be used to display the current size of the cache, the amount in use, and the type of cache—disk or memory.)

- Sets the size of each chunk of data in the cache and, by implication, the amount of data the Cache Manager requests at one time from the File Exporter. For a memory cache, if the total cache size divided by the chunk size leaves a remainder, **dfsd** rounds the number down, resulting in a slightly smaller cache.
- Sets the number of dcache entries allocated in machine memory for storing information about the cache chunks in a disk cache.
- Sets the number of empty V files created in the cache directory for a disk cache. (A memory cache cannot use V files because it does not use disk storage; the number of chunks is instead equal to the number of dcache entries.)
- Sets the number of **stat** entries in machine memory for caching status information about cached DFS files.
- Sets the number of entries in the name lookup cache for storing the results of remote directory lookups.
- Specifies the directory on the machine's local disk where DFS is mounted. The first field in the **CacheInfo** file specifies the default directory.
- Selects a random server machine in the local cell as the source of the correct time if the **-settime** option is provided.
- Sets the initial RPC authentication level and minimum RPC authentication bound for communications between the Cache Manager and File Servers.

In addition to setting cache configuration parameters, **dfsd** also starts the following types of daemons. On most system types, these daemons appear as nameless entries in the output of the **ps** command.

- One or more maintenance daemons, which perform routine periodic maintenance tasks such as the following:
  - — Performing garbage collection
  - — Synchronizing files
  - — Probing processes on File Server machines every few minutes
  - — Refreshing information about filesets referenced by the Cache Manager once per hour
  - — Keeping the machine's clock synchronized with the clock of the chosen server machine (if the **-settime** option is included with the **dfsd** command)
- One or more background daemons, which improve performance by performing delayed writing of updated data. The default number of background daemons is two, which is usually sufficient to handle up to five simultaneous users of a machine. Use the **-mainprocs** option to increase the number of background daemons if the machine serves more than five users.
- One or more token daemons, which handle token revocation RPC requests from the File Exporters on File Server machines (for example, by writing modified data back to the File Server machines). The default number of token daemons is two. Use the **-tokenprocs** option to increase this number if the machine interacts with many File Server machines from different cells.
- *On a machine running the AIX operating system*, one or more I/O daemons, which move data from disk to memory and from memory to disk. The default number of I/O daemons is five. Use the **-ioprocs** option to increase the number of I/O daemons performing I/O requests if the number of users working on the machine increases and the machine begins to experience performance problems.

  Use the **-ioprocs** *option only on a machine running AIX*. No I/O daemons are used on a machine not running AIX; the option is ignored if it is used on such a machine.

The default number of daemons is ten (one maintenance daemon, two background daemons, two token daemons, and five I/O daemons). You can alter only the number of background daemons, token daemons, and I/O daemons; **dfsd** initializes additional maintenance daemons as necessary.

RPC Security Settings

The **dfsd** command sets the DCE RPC security level for RPCs sent between a Cache Manager and DFS File Servers. The command sets two levels: an initial DCE RPC

security level used as a starting point in security level negotiations between the Cache Manager and a File Server, and the minimum DCE RPC security level that the Cache Manager will accept for such communications. Two sets of these levels are maintained: One set specifies the security levels for communications with File Servers within the local cell, and the other set specifies the security levels for communications with File Servers within foreign cells. Both sets of security levels can be adjusted through the **cm setprotectlevels** command.

In operation, the Cache Manager and File Server interact to arrive at a mutually acceptable authentication level for communications. The negotiation starts with an RPC that uses the initial authentication level sent from the Cache Manager to the File Server. If the initial authentication level is outside the minimum or maximum bounds set at the File Server, the File Server returns a response to the Cache Manager specifying that the authentication level is either too low or too high. The Cache Manager then decreases or increases its authentication level accordingly and retries the RPC. This process continues until the Cache Manager either adjusts its RPCs to an acceptable security level or the File Server requests a security level below the minimum set at the Cache Manager (causing the Cache Manager to refuse communications with the File Server). Once the Cache Manager and File Server have negotiated a security level, the Cache Manager stores this information so that it does not need to renegotiate this level for further communications with the File Server.

The Cache Manager and **fxd** default settings are such that communications occur at the packet integrity authentication level.

In addition to a general pair of upper and lower bounds for all communications between the File Server and Cache Manager, administrators can also set advisory bounds on a per-fileset basis. At present, these advisory levels serve only to bias the Cache Manager's selection of an initial authentication level (they may be enforced in a future version of DFS). Advisory bounds are set through the **fts setprotectlevels** command and are stored in the FLDB record for that fileset.

Note that the use of this command does not preclude communications with File Servers running earlier versions of DFS.

The various authentication levels are set by specifying either an integer value between 0 and 6, a complete string specifying the authentication level, or an abbreviation of that string as the *level* argument for the various command options. The following lists the various authentication levels:

- **rpc_protect_level_default** or **default** or **0**: Use the DCE default authentication level.

- **rpc_protect_level_none** or **none** or **1**: Perform no authentication.
- **rpc_protect_level_connect** or **connect** or **2**: Authenticate only when the Cache Manager establishes a connection with the File Server.
- **rpc_protect_level_call** or **call** or **3**: Authenticate only at the beginning of each RPC received.
- **rpc_protect_level_pkt** or **pkt** or **4**: Ensure that all data received is from the expected host.
- **rpc_protect_level_pkt_integrity** or **pkt_integrity** or **5**: Authenticate and verify that none of the data transferred has been modified.
- **rpc_protect_level_pkt_privacy** or **pkt_privacy** or **6**: Perform authentication as specified by all of the previous levels and also encrypt each RPC argument value.

Note that there is a trade-off between selecting higher security and performance. The higher levels of security require more overhead and increase the response time in file operations with File Servers.

Privileges Required

The issuer must be **root** on the local machine.

Examples

It is recommended that the **dfsd** process be included in the proper initialization file (/**etc/rc** or its equivalent) rather than typed at the command shell prompt. The **dfsbind** process must be run before the **dfsd** process in a start-up file. For most disk caches, the following form is appropriate in the initialization file:

dcelocal/**bin/dfsd**

The following line in an initialization file is appropriate when enabling a machine to serve more than five users:

dcelocal/**bin/dfsd -mainprocs 4**

The following line in an initialization file initializes a memory cache and sets the chunk size to 16 kilobytes (2^{14}):

dcelocal/**bin/dfsd -memcache -chunksize 14**

Related Information

Commands: **cm getcachesize(8dfs)**, **cm getprotectlevels(8dfs)**, **cm setcachesize(8dfs)**, **cm setprotectlevels(8dfs)**, **dfsbind(8dfs)**, **fts setprotectlevels**.

Files: **CacheInfo(4dfs)**, **CacheItems(4dfs)**, **FilesetItems(4dfs)**, **Vn(4dfs)**.

dfsexport

Purpose Exports DCE LFS aggregates and non-LFS partitions to the DCE namespace

Synopsis **dfsexport** [{**-all** | **-aggregate** *name*}] [**-type** *name*] [**-detach**][**-force**] [**-verbose**][**-help**]

Options

-all Specifies that all aggregates and partitions listed in the *dcelocal* **/var/dfs/dfstab** file are to be exported. Use the **-type** option with this option to export only DCE LFS aggregates or only non-LFS partitions. Use this option or use **-aggregate**; omit both options to list all aggregates and partitions currently exported from the local disk to the DCE namespace.

-aggregate *name*
Specifies the device name or aggregate name of the aggregate or partition to be exported. These names are specified in the first and second fields of the entry for the aggregate or partition in the **dfstab** file. Use this option or use **-all**; omit both options to list all aggregates and partitions currently exported from the local disk to the DCE namespace.

-type *name* Used with the **-all** option, specifies that only aggregates or partitions whose file system types match the type specified with this option are to be exported. The type can be specified as **lfs** to export only DCE LFS aggregates, or it can be specified as **ufs** to export only non-LFS partitions. The type of each aggregate or partition appears in the third field of the entry for the device in the **dfstab** file. The type must be specified in lowercase letters (as it appears in the **dfstab** file).

Use this option only with the **-all** option; it is ignored if it is used without the **-all** option. If it is omitted and **-all** is used, the command exports both **lfs** and **ufs** devices.

-detach Specifies that the aggregates or partitions indicated with the command's other options are to be detached (no longer exported), making them unavailable via the DCE namespace. Use **-all** or **-aggregate** with this

option to indicate the devices to be detached; use the **-type** option with **-all** to detach only one type of device.

Use the **-detach** option only when no users are accessing data on the aggregate or partition to be detached or when a serious emergency warrants its use. When the **-detach** option is used, the command revokes all tokens for data on a device before it detaches it. It does not detach a device unless it can revoke all necessary tokens. You can use the **-force** option to direct the command to detach a device even if it cannot revoke all necessary tokens.

To permanently detach an aggregate or partition, it must also be removed from the **dfstab** file. Otherwise, the **dfsexport** command exports the aggregate or partition the next time it is run (provided the aggregate or partition is included in the specification for the devices to be exported).

-force Used with the **-detach** option, directs the **dfsexport** command to detach an aggregate or partition even if it cannot revoke all tokens for data on the device. By default, the command does not detach a device unless it can revoke all necessary tokens. Use this option only when a serious emergency requires its use.

This option can be used only with the **-detach** option. The command fails if this option is used with any combination of options that does not include the **-detach** option.

-verbose Directs the command to report on its actions as it executes.

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

The **help** and **apropos** commands available with all command suites are also available with the **dfsexport** command. See the **bos help** and **bos apropos** reference pages for examples using these commands.

Description

The **dfsexport** command exports DCE LFS aggregates and non-LFS disk partitions from the local disk of a machine to the DCE namespace. File systems on exported aggregates and partitions are available to other users in the DCE namespace. The binary file for the **dfsexport** command resides in *dcelocal*/**bin/dfsexport**.

The command exports DCE LFS aggregates, non-LFS partitions, or both, based on the values provided with its options. If the **-all** option is provided, the command exports all aggregates and partitions listed in the *dcelocal*/**var**/**dfs**/**dfstab** file. If the **-aggregate** option is provided, it exports only the aggregate or partition whose device name or aggregate name is specified with the option. The specified name must be listed in the **dfstab** file.

The **-type** option can be used with the **-all** option to indicate that only DCE LFS aggregates or only non-LFS partitions are to be exported. If **lfs** is provided with the **-type** option, the command exports only DCE LFS aggregates; if **ufs** is specified with the **-type** option, it exports only non-LFS partitions. If the **-type** option is used, the **-all** option must also be included; otherwise, the **-type** option is ignored.

When **dfsexport** executes, it reads the **dfstab** file on the local disk of the machine to determine the aggregates and partitions available to be exported. An aggregate or partition must have an entry in the **dfstab** file if it is to be exported. Because this command reads the **dfstab** file, information supplied with its options must match exactly the information for an aggregate or partition specified in that file.

The **dfsexport** command reads a list of all currently exported aggregates and partitions that is maintained in the kernel of the local machine. The command will not export an aggregate or partition that is currently exported. The command also refuses to export a DCE LFS aggregate that needs to be recovered with the **salvage** command. If the **dfsexport** command fails with an exit status of **2**, use the **salvage** command to recover the aggregate that caused the failure and reissue the **dfsexport** command.

Issuing the **dfsexport** command with no options lists the aggregates and partitions currently exported from the local disk to the DCE namespace. The **fts lsaggr** command can also be used to display a current list of all aggregates and partitions exported from a machine.

The **dfsexport** command is generally included in a machine's initialization file (/**etc**/**rc** or its equivalent) rather than issued at the keyboard. Once included in the initialization file, the command automatically exports all indicated aggregates and partitions whenever the machine is rebooted. Typically, the command is included with its **-all** option to export all aggregates and partitions listed in the **dfstab** file.

Prior to using this command to export a non-LFS partition for the first time, perform the following steps:

1. Ensure that the partition is mounted locally; it can contain data or it can be empty.
2. Issue the **fts crfldbentry** command to register the non-LFS fileset that resides on the partition (each non-LFS partition contains a single fileset) in the Fileset

Location Database (FLDB). The Fileset Location Server (FL Server) can then track the fileset's location. The **fts crfldbentry** command also allocates a unique fileset ID number for the non-LFS fileset.

3. Create an entry for the non-LFS partition in the **dfstab** file on the machine on which the partition resides. Use the aggregate ID number specified with the **-aggrid** option of the **fts crfldbentry** command and the fileset ID number allocated by the command in the fourth and fifth fields of the entry for the partition. Also, use the name of the partition's local mount point as its aggregate name in the second field of its entry. (Once these steps are complete, use the **fts crmount** command to mount the non-LFS fileset that resides on the partition.)

Before exporting a non-LFS partition, also make sure that no users have files open on the partition. DFS cannot effectively synchronize file access between users who opened files from a non-LFS partition before the partition was exported and users who open files from the partition after the partition is exported because only the latter have tokens.

Before using this command to export a DCE LFS aggregate for the first time, complete the following steps:

1. Ensure that the disk partition on which the aggregate is to reside is initialized with the **newaggr** command; the partition cannot contain data when the **newaggr** command is executed. The **newaggr** command needs to be run on a partition only once. *Do not use the* **newaggr** *command to reinitialize a partition that contains data you want to preserve; the command destroys any data on the partition on which it is used.*

2. Create an entry for the DCE LFS aggregate in the **dfstab** file on the machine on which the aggregate is located. (Once the aggregate is exported, the **fts create** command can be used to create and register filesets on the aggregate, after which the **fts crmount** command can be used to mount the new filesets.)

The **dfsexport** command can also be used to detach an exported aggregate or partition from the DCE namespace. Detaching an aggregate or partition makes it unavailable in the namespace. To detach one or more aggregates or partitions, use the **-all** (and optionally the **-type**) option or the **-aggregate** option to specify the devices to be detached, and include the **-detach** option with the command.

Before it detaches a device, the command revokes all tokens for data on the device. When their tokens are revoked, clients flush data cached from the device, writing any modified data back to the device. If the command cannot revoke all necessary tokens,

it does not detach the device. (It instead displays a message reporting that the device is busy.)

The **-force** option can be used with the **-detach** option to direct the command to detach a device even if it cannot revoke all necessary tokens (that is, even if files from the device are still open). (You can also remove an aggregate or partition from the DCE namespace by removing its entry from the **dfstab** file and rebooting the machine.)

Privilege Required

If the command is issued with no options to list the aggregates and partitions exported from the local machine, no privileges are required. Otherwise, the issuer must be logged in as **root** on the local machine.

Cautions

Before detaching an aggregate or partition, attempt to ensure that no users are currently accessing data from filesets on the device. The command revokes all tokens for data on the device before it detaches it, which causes clients to flush data cached from the device (writing any modified data back to the File Server machine). However, a user who is accessing data from the device will no longer be able to save the data. Any attempt to perform an action that involves a detached aggregate or partition elicits a message reporting that the device is unknown. Exercise special caution before using both the **-detach** and **-force** options, which forces a device to be detached even if all tokens cannot be revoked (that is, even if files are still open).

Examples

The following command line is typically added to a machine's initialization file (**/etc/rc** or its equivalent). The line exports all of the aggregates and partitions that have entries in the machine's **dfstab** file.

```
dfsexport -all
```

The following command exports the aggregate whose device name (as it appears in the **dfstab** file) is **/dev/lv02**:

```
# dfsexport /dev/lv02
```

The command that follows exports all DCE LFS aggregates (all entries in the **dfstab** file with file system type **lfs**):

```
# dfsexport -all -type lfs
```

Exit Values

The **dfsexport** command can return the following exit values:

0	The command completed successfully.
1	The command failed for a reason other than that associated with an exit value of **2**.
2	The command failed because a DCE LFS aggregate to be exported needs to be recovered with the **salvage** command before it can be exported.

Related Information

Commands: **fts create(8dfs)**, **fts crfldbentry(8dfs)**, **fts crmount(8dfs)**, **fts lsaggr(8dfs)**, **newaggr(8dfs)**, **salvage(8dfs)**.

Files: **dfstab(4dfs)**.

dfsgw

Purpose

Introduction to the dfsgw command suite used with the DFS/NFS Secure Gateway

Options

The following options are used with many **dfsgw** commands. They are also described with the commands that use them.

-id *networkID:userID*
: Identifies an NFS client and the user whose DCE authentication from that client is to be manipulated. You can specify the network address or hostname of the NFS client; you must specify the UNIX user identification number (UID) of the user.

-dceid *login_name[:password]*
: Specifies the DCE principal name and password of the user for whom an entry in the authentication table (AT) is to be created.

-af *address_family*
: Specifies the style of network address to be used to identify hosts. By default, the command uses the only address family currently supported, **inet** (Internet).

-help
: Displays the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfsgw** command suite provides commands to manipulate entries in the local authentication table on a Gateway Server machine. The table contains an entry for each user who has DCE credentials on the Gateway Server machine. Each entry is a mapping that pairs the UID of the user and the network address of the NFS client for which the user has DCE credentials with the user's Process Activation Group (PAG).

The **dfsgw** command suite includes the following commands:

dfsgw add Obtains DCE credentials to provide a user with authenticated access to DFS from a specified NFS client. The command adds an entry to the authentication table to provide the user with authenticated access from the client. The command provides the same basic functionality from a Gateway Server machine that the **dfs_login** command provides from an NFS client.

dfsgw delete
Cancels a user's authenticated access to DFS from a specified NFS client. The command removes the user's entry for the client from the authentication table. The command provides the same basic functionality from a Gateway Server machine that the **dfs_logout** command provides from an NFS client.

dfsgw list Displays information about all users who are authenticated to DCE via the Gateway Server machine. The command lists all entries in the authentication table.

dfsgw query
Determines whether a specific user is authenticated to DCE via the Gateway Server machine. The command determines whether the user has an entry in the authentication table.

Commands in the **dfsgw** command suite provide a local administrative interface to the authentication table on a machine configured as a Gateway Server. Because each Gateway Server machine maintains its own authentication table, you must issue **dfsgw** commands on the Gateway Server machine whose authentication table you want to manipulate. The **dfs_login** and **dfs_logout** commands provide a remote mechanism for creating and deleting entries in the table.

Receiving Help

There are several different ways to receive help about DFS commands. The following examples summarize the syntax for the different help options:

$ **man dfsgw**
Displays the reference page for the command suite.

$ **man dfsgw_** *command*
Displays the reference page for an individual command. You must use an **_** (underscore) to connect the command suite to the command name. *Do not use the underscore when issuing the command.*

$ **dfsgw help**
Displays a list of commands in a command suite.

$ **dfsgw help** *command*
Displays the syntax for a single command.

$ **dfsgw***command* **-help**
Displays the syntax for a single command.

$ **dfsgw apropos -topic** *string*
Displays a short description of commands that match the specified *string*.

Consult the **dfs_intro(8dfs)** reference page for complete information about the DFS help facilities.

Privilege Required

To use the **add**, **delete**, or **query** command, the issuer must be logged into the Gateway Server machine either as the user whose credentials are to be manipulated or as local **root**. To use the **list** command, no privileges are required.

Exit Values

All **dfsgw** commands return an exit value of **0** (zero) upon successful completion. Otherwise, they return a nonzero exit value.

Related Information

Commands: **dfsgw_add(8dfs)**, **dfsgw_apropos(8dfs)**, **dfsgw_delete(8dfs)**, **dfsgw_help(8dfs)**, **dfsgw_list(8dfs)**, **dfsgw_query(8dfs)**, **dfs_intro(8dfs)**, **dfs_login(8dfs)**, **dfs_logout(8dfs)**.

dfsgw add

Purpose

Adds an entry to the authentication table on the Gateway Server machine

Synopsis

dfsgw add -id *networkID:userID* [**-dceid** *login_name* [: *password*]] [**-sysname** *sysname*] [**-remotehost** *name*] [**-af** *address_family*] [**-help**]

Options

-id *networkID:userID*

Identifies an NFS client and the user who is to be authenticated to DCE from that client. You can specify the network address or the host name of the NFS client; you must specify the UNIX user identification number (UID) of the user. The command creates an entry for the user in the local authentication table (AT) to provide the user with authenticated access to DFS from the specified NFS client.

-dceid *login_name*[**:** *password*]

Specifies the DCE principal name and, optionally, the password of the user for whom an entry is to be added to the authentication table. If you do not specify a principal name and password, the command prompts for them only if you do not already have a valid ticket-granting ticket (TGT) in the current login context. Similarly, if you specify your own principal name but omit your password, the command prompts for your password only if you do not already have a valid TGT in the current login context. The command always prompts for a password if you name a principal other than yourself. The command's interactive prompt provides for secure entry of the password.

-sysname *sysname*

Specifies the system name for *networkID*. This option defaults to the system name of the Gateway Server machine. The *sysname* argument is a unique name derived from **uname( )** that describes the machine architecture and OS type, such as **hp700_ux905** or **hp800_ux90**.

-remotehost *name*
Specifies the name of the remotehost. The default is the host name of *networkID*.

-af *address_family*
Specifies the style of network address to be used to identify hosts. By default, the command uses the only address family currently supported, **inet** (Internet).

-help Displays the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfsgw add** command authenticates a user to DCE. The command contacts the DCE Security Service to obtain a TGT for the user. To obtain a TGT, a user must have a valid account in the registry database of the DCE cell. The TGT is used to create a valid login context for the user. The login context includes a Process Activation Group (PAG), which DFS stores in the kernel of the Gateway Server machine to identify the user's TGT. The TGT serves as the user's DCE credentials to provide authenticated access to files and directories in the DFS filespace from the specified NFS client.

The **dfsgw add** command adds an entry for the user to the authentication table on the local Gateway Server machine. The entry is a mapping that pairs the user's UID and the network address of the NFS client for which the user has DCE credentials with the user's PAG. Because each Gateway Server machine maintains its own authentication table, you must issue the command on the Gateway Server machine on which an entry is to be added to the authentication table.

DCE credentials obtained with the command are valid for the default ticket lifetime in effect in the registry database of the DCE cell. Once they expire, the credentials can no longer be used for authenticated access to DFS. You can obtain new credentials by issuing the **dfsgw add** command on the Gateway Server machine or by issuing the **dfs_login** command on the NFS client from which you want authenticated access. The two commands provide essentially the same functionality, with the exception that the **dfs_login** command lets you request a specific ticket lifetime.

The **dfsgw add** command does not obtain a new TGT if you do not name a principal other than yourself on the command line and you already have a valid TGT in the current login context. If you do not already have an entry in the authentication table for the specified NFS client, the command uses your existing PAG to create a new entry for you. If you already have an entry in the authentication table for the NFS

client, the command has no effect. In either case, the command does not affect existing entries in the authentication table, and it does not affect the remaining ticket lifetime of your existing TGT.

Use the **dfsgw delete** command or the **dfs_logout** command to end an authenticated session by removing an entry from the authentication table.

Privileges Required

The issuer must be logged into the Gateway Server machine either as the user for whom credentials are to be created or as local **root**.

Output

The **dfsgw add** command displays the following prompts to request a DCE principal and password:

```
Enter Principal Name: principal
Enter Password: password
```

where *principal* is the name of the user to be authenticated to DCE, and *password* is the password of the named user; you supply both of these values. The command prompts for the *principal* name only if you do not specify a principal name with the **-dceid** option and you do not already have a valid TGT. The command prompts for the *password* only if you do not specify a password with the **-dceid** option and if either of the following is true:

- You name a user other than yourself with the **-dceid** option
- You do not name a user other than yourself with the **-dceid** option and you do not already have a valid TGT

If it succeeds in creating the entry in the authentication table, the command displays the following:

```
Mapping added successfully, PAG is PAG
```

where *PAG* identifies the PAG created with the command.

Examples

The following command creates an entry in the authentication table to grant authenticated access to DFS to the user named **ludwig**. The user, whose UID is **7439**, is requesting access from the NFS client that has network address **15.27.32.40**. The user provides the principal name with the **-dceid** option but omits the password; the command prompts for the user's password, which the user specifies as **beethoven** in the example.

```
dfsgw add -id 15.27.32.40:7439 -dceid ludwig
Enter Password: beethoven
Mapping added successfully, PAG is 41ffffe4
```

Exit Values

The **dfsgw add** command returns an exit value of **0** (zero) if it adds an entry for the user to the authentication table. Otherwise, it returns a nonzero exit value.

Related Information

Commands: **dfs_login(8dfs)**, **dfs_logout(8dfs)**, **dfsgw_delete(8dfs)**, **dfsgw_list(8dfs)**, **dfsgw_query(8dfs)**.

dfsgw apropos

Purpose

dfsgw apropos – Shows each help entry that contains a specified string

Synopsis

dfsgw apropos -topic *string* [**-help**]

Options

-topic *string* Specifies the keyword string for which to search. If it is more than a single word, surround the string with " " (double quotes) or other delimiters. Type all strings for **dfsgw** commands in lowercase letters.

-help Displays the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfsgw apropos** command displays the first line of the help entry for any **dfsgw** command that contains the string specified by the **-topic** option in its name or short description.

To display the syntax for a command, use the **dfsgw help** command.

Privilege Required

No privileges are required.

Output

The first line of an online help entry for a command lists the command and briefly describes its function. This command displays the first line for any **dfsgw** command where the string specified by the **-topic** option is part of the command name or the first line.

Examples

The following command lists all **dfsgw** commands that have the word **entry** in their names or short descriptions:

```
$ dfsgw apropos entry
```

```
add: add an entry to the AT
delete: delete an entry from the AT
```

Related Information

Commands: **dfsgw help(8dfs)**.

dfsgw delete

Purpose

Removes an entry from the authentication table on the Gateway Server machine

Synopsis

dfsgw delete -id *networkID:userID* [**-af** *address_family*] [**-help**]

Options

-id *networkID:userID*
: Identifies an NFS client and the user whose authentication to DCE from that client is to be canceled. You can specify the network address or the hostname of the NFS client; you must specify the UNIX user identification number (UID) of the user. The command removes the user's entry for the specified NFS client from the local authentication table (AT).

-af *address_family*
: Specifies the style of network address to be used to identify hosts. By default, the command uses the only address family currently supported, **inet** (Internet).

-help
: Displays the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfsgw delete** command cancels a user's authenticated access to DFS. The command removes the entry for the specified user and NFS client from the authentication table on the Gateway Server machine. Once the command removes the entry from the authentication table, the user for whom the entry existed no longer has authenticated access to DFS from the NFS client for which the entry existed.

Because each Gateway Server machine maintains its own authentication table, you must issue the command on the Gateway Server machine from which an entry is to be removed from the authentication table. The command has no effect on entries the

user may have in the authentication table for other NFS clients, and it has no effect on entries in the authentication tables on other Gateway Server machines.

You can also end an authenticated session by issuing the **dfs_logout** command on the NFS client from which authenticated access is no longer needed. To obtain DCE credentials and create an entry in the authentication table, use the **dfsgw add** command or the **dfs_login** command.

Privilege Required

The issuer must be logged into the Gateway Server machine either as the user whose entry is to be removed from the authentication table or as local **root**.

Examples

The following command deletes the entry from the authentication table that grants authenticated access to the user named **ludwig** from the NFS client that has network address **15.27.32.40**. The command is issued by the user **ludwig**, who has UID **7439**. Once the command is issued, **ludwig** no longer has authenticated access to DFS from the NFS client.

```
$ dfsgw del -id 15.27.32.40:7439
```

Exit Values

The **dfsgw delete** command returns an exit value of **0** (zero) if it removes the entry for the specified user from the authentication table. Otherwise, it returns a nonzero exit value.

Related Information

Commands: **dfsgw_add(8dfs)**, **dfsgw_list(8dfs)**, **dfsgw_query(8dfs)**, **dfs_login(8dfs)**, **dfs_logout(8dfs)**.

dfsgw help

Purpose

dfsgw help – Shows syntax of specified **dfsgw** commands or lists functional descriptions of all **dfsgw** commands

Synopsis

dfsgw help [**-topic** *string*]… [**-help**]

Options

-topic *string* Specifies each command whose syntax is to be displayed. Provide only the second part of the command name (for example, **list**, not **dfsgw list**). If this option is omitted, the output provides short descriptions of all **dfsgw** commands.

-help Displays the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfsgw help** command displays the first line (name and short description) of the online help entry for every **dfsgw** command if the **-topic** option is not provided. For each command name specified with the **-topic** option, the output lists the entire help entry.

Use the **dfsgw apropos** command to show each help entry that contains a specified string.

Privilege Required

No privileges are required.

Output

The online help entry for each **dfsgw** command consists of the following two lines:

- The first line names the command and briefly describes its function.
- The second line, which begins with **Usage:**, lists the command options in the prescribed order.

Examples

The following command displays the online help entry for the **dfsgw list** command:

```
$ dfsgw help list
```

```
dfsgw list: list all entries in the AT
Usage: dfsgw list [-help]
```

Related Information

Commands: **dfsgw apropos(8dfs)**.

dfsgw list

Purpose

Lists all entries in the authentication table on the Gateway Server machine

Synopsis

dfsgw list [**-help**]

Options

-help Displays help information for this command.

Description

The **dfsgw list** command lists all users who have DCE credentials for authenticated access to DFS from NFS clients. To provide this information, the command lists all entries from the local authentication table (AT). Because each Gateway Server machine maintains its own authentication table, you must issue the command on the Gateway Server machine that houses the authentication table from which entries are to be displayed.

Use the **dfsgw query** command to see the entry in the authentication table for a specific user. Note that the **dfsgw list** command provides some additional information not displayed by the **dfsgw query** command. For example, it displays the host name of the NFS client for which the DCE credentials are granted, the principal name of the user to whom the credentials are granted, the date and time at which the credentials expire, and the system name and remote host name used for the client.

Privileges Required

No privileges are required.

Output

The **dfsgw list** command displays the following output for each entry in the authentication table:

```
Mapping: hostname : principal =>PAG
Expires at date/time
@host=remotehost @sys=sysname
```

where

hostname Names the NFS client for which the entry grants authenticated access to DFS

principal Displays the principal name of the user to whom the entry grants authenticated access

PAG Identifies the Process Activation Group (PAG) that exists for the *hostname/principal* pair

date/time Specifies the date and time at which the DCE credentials identified by the PAG expire

remotehost Identifies the remote host name used for the *hostname /principal* pair

sysname Identifies the system name used for the *hostname /principal* pair

The **dfsgw list** command displays the following output if the authentication table contains no entries:

```
No mappings exist
```

Examples

The following command displays the current entries from the authentication table on the local Gateway Server machine. The first entry grants secure access to DFS to the user **ludwig** from the NFS client named **nfs1.abc.com**. The PAG associated with the user is **41ffffe4**; the user's DCE credentials expire at 6:33 a.m. on 23 July 1994.

```
dfsgw list
Mapping: nfs1.abc.com:ludwig => 41ffffe4
Expires at Sat Jul 23 06:33:18 1994
(@host=host1.xyz.com @sys=hp700ux_1001)
Mapping: nfs2.abc.com:frost => 41ffffa3
Expires at Sat Jul 23 08:36:23 1994
```

```
(@host=host2.xyz.com @sys=hp700ux_1001)
Mapping: nfs2.abc.com:wvh => 41ffffbe
Expires at Sun Jul 24 00:51:21 1994
(@host=host3.xyz.com @sys=hp700ux_1001)
.
.
.
```

Exit Values

The **dfsgw list** command returns an exit value of **0** (zero) if it succeeds in listing the entries from the authentication table. Otherwise, it returns a nonzero exit value.

Related Information

Commands: **dfs_login(8dfs)**, **dfs_logout(8dfs)**, **dfsgw_add(8dfs)**, **dfsgw_delete(8dfs)**, **dfsgw_query(8dfs)**.

dfsgw query

Purpose

Queries the authentication table on the Gateway Server machine

Synopsis

dfsgw query -id *networkID:userID* [**-af** *address_family*] [**-help**]

Options

-id *networkID:userID*
: Identifies an NFS client and the user whose authentication from the client is to be determined. You can specify the network address or the hostname of the NFS client; you must specify the UNIX user identification number (UID) of the user. The command searches the local authentication table (AT) to determine whether the user has an entry for the specified NFS client.

-af *address_family*
: Specifies the style of network address to be used to identify hosts. By default, the command uses the only address family currently supported, **inet** (Internet).

-help
: Displays the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfsgw query** command determines whether the specified user has DCE credentials for authenticated access to DFS from the specified NFS client. To provide this information, the command checks the local authentication table to determine whether the user has an entry for the NFS client. Because each Gateway Server machine maintains its own authentication table, you must issue the command on the Gateway Server machine that houses the authentication table to be queried. The command determines only whether the user has an entry for the specified client; the command does not report whether the user has entries for any other clients.

Use the **dfsgw list** command to see all entries in the authentication table. The **dfsgw list** command provides some additional information not displayed by the **dfsgw query** command. The **dfsgw query** command is useful for inclusion in scripts that determine only whether a user has authenticated access to DFS from an NFS client.

Privilege Required

The issuer must be logged into the Gateway Server machine either as the user whose entry in the authentication table is to be examined or as local **root**.

Output

The **dfsgw query** command displays the following line of output if the specified user has an entry for the specified NFS client in the authentication table:

```
Mapping found, PAG is PAG
```

where *PAG* identifies the Process Activation Group (PAG) that exists for the user. If the user does not have an entry for the NFS client in the authentication table, the **dfsgw query** command displays the following line of output instead:

```
No mapping found
```

Examples

The following command determines whether the authentication table on the local Gateway Server machine includes an entry that provides authenticated access to the user named **ludwig** from the NFS client that has network address **15.27.32.40**. The user **ludwig** has UID **7439**. The command reports that **ludwig** has an entry in the table; the PAG associated with the user is **41ffffe4**.

```
$ dfsgw query -id 15.27.32.40:7439
```

```
Mapping found, PAG is 41ffffe4
```

Exit Values

The **dfsgw add** command returns an exit value of **0** (zero) if it finds an entry for the specified user in the authentication table. Otherwise, it returns a nonzero exit value.

Related Information

Commands: **dfsgw_add(8dfs)**, **dfsgw_delete(8dfs)**, **dfsgw_list(8dfs)**, **dfs_login(8dfs)**, **dfs_logout(8dfs)**.

dfsgwd

Purpose

Initializes the Gateway Server process for the DFS/NFS Secure Gateway

Synopsis

dfsgwd [**-service** *service_number*] [**-sysname** *sysname*] [**-nodomains**][**-file** *log_file*] [**-verbose**][**-help**]

Options

-service *service_number*
: Specifies the port number to be used to communicate with the **dfsgwd** process on the Gateway Server machine. By default, the process uses port number **438**, the port number defined for the Gateway Server process in the **/etc/services** file or Network Information Services (NIS) services map file. (See the **services(4)** reference page in the *OSF/1 System and Network Administrator's Reference* for more information.)

-sysname *sysname*
: Specifies the system name for this Gateway Server. **dfsgwd** can handle NFS clients that do not recognize **@sys** and **@host**, using a system name of **unknown**. This name can be set by starting **dfsgwd** with the **-sysname** option. The *sysname* argument is a unique name derived from **uname( )** that describes the machine architecture and OS type, such as **hp700_ux905** or **hp800_ux90**.

-nodomains
: Uses the base host name (without the domain portion) for **@host**.

-file *log_file*
: Specifies the full pathname of the log file in which the **dfsgwd** process records information about the operations it performs. By default, the **dfsgwd** process writes output to the log file named *dcelocal*/**var/dfs/adm/DfsgwLog**.

-verbose
: Directs the process to write a message of the following form to the indicated log file each time an entry is added to the authentication table (AT):

```
INFO:  Adding ticket for "username"
```

where *username* is the name of the user for whom the entry is added.

-help Displays the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfsgwd** command initializes the Gateway Server process. The **dfsgwd** process runs on machines configured as DFS clients to enable remote authentication via the **dfs_login** command. The **dfsgwd** process works with the **dfs_login** command to obtain DCE credentials for users of NFS clients. The DCE credentials provide users with authenticated access to data in DFS.

The Gateway Server process manipulates mappings for authenticated users in the authentication table on the Gateway Server machine. Each mapping records the following information for an authenticated user:

- The user's UNIX user identification number (UID)
- The network address of the NFS client from which the user has authenticated access to DFS
- The PAG that stores the user's DCE ticket-granting ticket (TGT)

The **dfs_login** and **dfs_logout** commands provide a remote mechanism for creating and deleting entries in the authentication table on a Gateway Server machine. Commands in the **dfsgw** command suite provide a local administrative interface to the authentication table on a machine configured as a Gateway Server.

The Gateway Server process recognizes the **@sys** and **@host** variables on the NFS client system. This allows the Gateway Server to resolve pathnames to binaries and other system_dependent files correctly, based on the user's login system name and system type.

The binary file for the **dfsgwd** process resides in *dcelocal*/**bin**. The process is normally run on a DFS client that is exporting a mount point for **/...**, the root of the DCE namespace, via NFS. The process runs as the DCE principal **hosts**/*hostname*/**dfsgw-server**.

The **dfsgwd** process is usually started and controlled by the Basic OverSeer (BOS) Server (**bosserver**) process. The BOS Server restarts each process it monitors whenever the system is rebooted. If the **dfsgwd** process is not controlled by the

BOS Server, the **dfsgwd** process runs in the foreground by default. See Part 1 of this manual for information about configuring the **dfsgwd** process on a machine to be configured as a Gateway Server.

The **dfsgwd** process writes output about the operations it performs to a log file. By default, it writes output to the file named *dcelocal*/**var/dfs/adm/DfsgwLog**. You can use the **-file** option to name a different log file. If the **dfsgwd** process is controlled by the BOS Server, you can use the **bos getlog** command to read the log file.

Privileges Required

The issuer must be **root** on the local machine.

Files

dcelocal/**var/dfs/adm/DfsgwLog**
: The default log file for the **dfsgwd** process. You can use the **-file** option to specify a different pathname for the log file.

Related Information

Commands: **bos getlog(8dfs)**, **bosserver(8dfs)**, **dfs_login(8dfs)**, **dfs_logout(8dfs)**, **dfsgw(8dfs)**.

Files: **DfsgwLog(4dfs)**.

dfstrace

Purpose

dfstrace – Introduction to the **dfstrace** command suite

Options

The following options are used with many **dfstrace** commands. They are also listed with the commands that use them.

-set *set_name*

Specifies the name of an event set to be utilized by the command. An event set is a module designed to track specific events within the DFS kernel or within one or more server processes. Each event set generates trace messages relative to the type of events that it tracks and writes information on each event to, from one to eight trace logs. Tracing by event set allows users of the **dfstrace** commands to more quickly isolate and diagnose a problem.

Following are some of the DFS kernel event sets that you can see:

- **cm** – Cache Manager package
- **fshost** – File exporter host package
- **fx** – File exporter package
- **episode/anode** – LFS anode package
- **episode/logbuf** – LFS buffer/logging package
- **episode/vnops** – LFS vnode package
- **tkc** – Token cache package
- **tkm** – Token manager package
- **tpq** – Thread pool queue package
- **xops** – Vnode-to-fileset synchronization package

Following are some of the server process event sets that you can see:

- **bosserver** – **bosserver** package

- **dacl** – DFS ACL package
- **dfsauth** – DFS security package
- **flserver** – **flserver** package
- **ftserver** – **ftserver** package
- **ftutil** – Fileset utility package
- **ubikdisk** – Disk I/O subset of Ubik package
- **ubikvote** – Sync site election subset of Ubik package

-log *log_name*
Specifies the name of a server process or kernel trace log to be utilized by the command. All logs are stored in kernel or server process memory that is allocated on the initialization of the kernel or server process. The default size of a log, which is measured in 4-kilobyte units (kwords), is predefined; however, this size can be changed by a system administrator. Any number of event sets can write to a single log.

-cdsentry *server_entry_in_CDS*
Specifies the name of a server process to which to connect. This option is required when performing operations on server process logs and event sets; it must be omitted when performing operations on kernel logs and event sets. The full DCE pathname of a server process must be specified with **/.:/hosts/** *machine*/*process_name*.

-help Prints the online help for the command. All other valid options specified with this option are ignored. For complete details about receiving help, see the **dfs_intro(8dfs)** reference page.

Description

Commands in the **dfstrace** command suite are used by system administrators to diagnose problems within the DFS kernel or within the server processes that interface with the **dfstrace** command suite. This diagnosis is performed by reading the output of trace logs containing diagnostic messages written by event sets that track specific actions performed by the DFS kernel or a server process.

The commands in the **dfstrace** command suite allow you to perform the following functions:

- List information about event sets (using the **dfstrace lsset** command)

- Set an event set's state (using the **dfstrace setset** command)
- List information about trace logs (using the **dfstrace lslog** command)
- Change the size of trace logs (using the **dfstrace setlog** command)
- Dump the contents of trace logs (using the **dfstrace dump** command)
- Clear trace logs (using the **dfstrace clear** command)

Receiving Help

There are several different ways to receive help about DFS commands. The following examples summarize the syntax for the different help options:

$ **man dfstrace**
: Displays the reference page for the command suite.

$ **man dfstrace_** *command*
: Displays the reference page for an individual command. You must use an _ (underscore) to connect the command suite to the command name. *Do not use the underscore when issuing the command in DFS.*

$ **dfstrace help**
: Displays a list of commands in a command suite.

$ **dfstrace help***command*
: Displays the syntax for a single command.

$ **dfstrace apropos -topic** *string*
: Displays a short description of any commands that match the specified *string*.

Consult the **dfs_intro(8dfs)** reference page for complete information about the DFS help facilities.

Privilege Required

Except for the **dfstrace help** and **dfstrace apropos** commands, which require no privilege, executing the **dfstrace** commands require one of the following two types of privilege, depending on the operation being performed:

- If the **dfstrace** command is executed on a kernel log or event set, the issuer must be logged in as **root** on the local machine.
- If the **dfstrace** command is executed on a server process log or event set, the issuer must be listed in the admin list associated with that process on the machine

specified with the **-cdsentry** option (for example, **admin.fl** for the **flserver** process and **admin.ft** for the **ftserver** process).

Specific privilege information is listed with each command's description. In addition, if the BOS Server, on the same machine as a server process, is running with DFS authorization checking disabled, no privilege is required to issue **dfstrace** commands on the event sets and logs associated with that server process.

Related Information

Commands: **dfs_intro(8dfs)**, **dfstrace apropos(8dfs)**, **dfstrace clear(8dfs)**, **dfstrace dump(8dfs)**, **dfstrace help(8dfs)**, **dfstrace lslog(8dfs)**, **dfstrace lsset(8dfs)**, **dfstrace setlog(8dfs)**, **dfstrace setset(8dfs)**.

dfstrace apropos

Purpose

dfstrace apropos – Shows each help entry containing a specified string

Synopsis

dfstrace apropos -topic *string* [**-help**]

Options

-topic *string* Specifies the keyword string for which to search. If it is more than a single word, surround the string with " " (double quotes) or other delimiters. Type all strings for **dfstrace** commands in lowercase letters.

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

Description

The **dfstrace apropos** command displays the first line of the help entry for any **dfstrace** command containing the string specified by **-topic** in its name or short description.

To display the syntax for a command, use the **dfstrace help** command.

Privilege Required

No privileges are required.

Output

The first line of an online help entry for a command lists the command and briefly describes its function. This command displays the first line for any **dfstrace** command where the string specified by **-topic** is part of the command name or the first line.

Examples

The following command lists all **dfstrace** commands that have the word **list** in their names or short descriptions:

```
$ dfstrace apropos list
```

```
lslog: list available logs
lsset: list available event sets
```

Related Information

Commands: **dfstrace help(8dfs)**.

dfstrace clear

Purpose

dfstrace clear – Clears server process or kernel trace logs

Synopsis

dfstrace clear [{**-set** *set_name*… | **-log** *log_name*}] [**-cdsentry** *server_entry_in_CDS*] [**-help**]

Options

-set *set_name*
: Specifies the name of each event set whose logs you want to clear. Specify this option or the **-log** option; omit both to clear all nonpersistent kernel logs on the local machine or all nonpersistent server process logs for the server process specified with the **-cdsentry** option.

-log *log_name*
: Specifies the name of each log that you want to clear. Specify this option or the **-set** option; omit both to clear all nonpersistent kernel logs on the local machine or all nonpersistent server process logs for the server process specified with the **-cdsentry** option.

-cdsentry*server_entry_in_CDS*
: Specifies the full DCE pathname (**/.:/hosts/***machine /process_name*) of a server process whose logs you want to clear. Use the **-set** or **-log** option with this option to specify a distinct group of server process logs to clear; use this option alone to clear all nonpersistent logs associated with the server process. Omit this option to clear kernel logs.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfstrace clear** command clears the specified server process or kernel logs. If you want to clear a kernel log, it must reside on the local machine. If you want to clear a

server process log, it can reside on any machine; however, you must use the **-cdsentry** option to specify the appropriate server process.

To clear a specific log, identify the log with the **-set** or **-log** option. Use the **-cdsentry** option to clear a specific server process log; omit that option to clear a specific kernel log.

To clear all kernel logs on a machine, run the **dfstrace clear** command without any options. To clear all server process logs associated with a particular server, run the command with the **-cdsentry** option only. Note that you cannot clear persistent logs in this global manner. The persistent attribute prevents accidental clearing of important logs. The attribute is assigned to a log when the kernel or server process is compiled and cannot be changed.

Privilege Required

To clear a kernel log, the issuer must be logged in as **root** on the local machine. To clear a server process log, the issuer must be listed in the **admin** list associated with that process on the machine specified with the **-cdsentry** option (for example, **admin.fl** for the **flserver** process and **admin.ft** for the **ftserver** process).

Examples

The following command clears all logs used by the **fx** kernel event set on the local machine:

```
# dfstrace clear fx
```

The following command clears all logs used by the **ftserver** process on the machine **dewitt**:

```
$ dfstrace clear -cdsentry /.:/hosts/dewitt/ftserver
```

Related Information

Commands: **dfstrace lslog(8dfs)**, **dfstrace lsset(8dfs)**.

dfstrace dump

Purpose

dfstrace dump – Dumps server process or kernel trace logs

Synopsis

dfstrace dump [{**-set** *set_name*… | **-follow** *log_name*}] [**-file** *output_filename*] [**-sleep** *seconds_between_reads*] [**-cdsentry** *server_entry_in_CDS*] [**-help**]

OPTIONS

-set *set_name*

Specifies the name of each event set whose corresponding logs you want to dump. Specify this option or the **-follow** option; omit both to dump all kernel logs on the local machine or all server process logs for the server process specified with the **-cdsentry** option. If you specify multiple event sets that point to the same log, that log is dumped multiple times.

-follow *log_name*

Specifies the name of a kernel log to continuously dump. Process server logs cannot be continuously dumped. When a log is continuously dumped, it is also cleared. Specify this option or the **-set** option; omit both to dump all kernel logs on the local machine or all server process logs for the server process specified with the **-cdsentry** option.

-file *output_filename*

Indicates the name of a file to which to write the output of the command. If the log being dumped is a server process log, the *output_filename* cannot contain / (slashes); the file is automatically placed in the directory *dcelocal* **/var/dfs/adm**. Furthermore, if an *output_filename* is not provided, the output is placed in the file **icl.***server_process_name*. Server process logs cannot be directly dumped to standard output. (If the output file for a server process log already exists, the older version is moved to the file *output_filename***.old**.) If the log being dumped is a kernel log, the *output_filename* must specify the full or relative pathname of the output file.

-sleep *seconds_between_reads*
Defines the number of seconds that the command pauses between dumps when dumping a kernel log in continuous mode. This option can only be used with the **-follow** option. The default value is 10 seconds.

-cdsentry *server_entry_in_CDS*
Specifies the full DCE pathname (/**.:/hosts/***machine /process_name*) of a server process whose logs you want to dump. Use the **-set** option with this option to specify a distinct group of server process logs to dump; use this option alone to dump all logs associated with the specified server process. Omit this option to dump kernel logs.

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

DESCRIPTION

The **dfstrace dump** command dumps the specified kernel logs to standard output or the specified server process logs to the *output_filename* specified with the **-file** option. Server process logs cannot be directly dumped to standard output. If an *output_filename* is not provided for a server process log dump, the output is place in the file **icl.***server_entry_in_CDS*. The contents of a kernel log dump can be directed into a file, rather than to standard output, by using the **-file** option.

If you want to dump a kernel log, it must reside on the local machine. If you want to dump a server process log, it can reside on any machine; however, you must use the **-cdsentry** option to specify the appropriate server process.

To dump specific logs, identify the logs with the **-set** option. Use the **-cdsentry** option to dump specific server process logs; omit that option to dump specific kernel logs.

To continuously dump a single kernel log, issue the command with the **-follow** option. Server process logs cannot be continuously dumped.

To dump all kernel logs on a machine, run the **dfstrace dump** command without the **-set** or **-follow** option. To dump all server process logs associated with a particular server, run the command with the **-cdsentry** option, but without the **-set** option.

Privilege Required

To dump a kernel log, the issuer must be logged in as **root** on the local machine. To dump a server process log, the issuer must be listed in the **admin** list associated with

that process on the machine specified by the **-cdsentry** option (for example, **admin.fl** for the **flserver** process and **admin.ft** for the **ftserver** process).

OUTPUT

At the beginning of the output of each dump is the date and time at which the dump began. Unless the **-follow** option is specified, the number of logs being dumped is displayed. The content of each log is preceded by a message identifying the log.

Each log message contains the following three components:

- The time stamp associated with the message
- The process ID or thread ID associated with the message
- The message itself

Every 1024 seconds, a current time message is written to each log. This message has the following format:

```
time timestamp, pid 0: Current time: unix_time
```

Use the current time message to determine the actual time associated with each log message as follows:

1. Locate the log message whose actual time you want to determine.
2. Search backward through the dump record until you come to a current time message.
3. If the current time message's time stamp is smaller than the log message's time stamp, subtract the former from the latter. If the current time message's time stamp is larger than the log message's time stamp, add 1024 to the latter and subtract the former from the result.
4. Add the resulting number to the current time message's *unix_time* to determine the log message's actual time.

Since log data is stored in a finite, circular buffer, some of the data can be overwritten before being read. If this happens, the following message appears at the appropriate place in the dump:

```
Log wrapped; data missing.
```

Note: If this message appears in the middle of a dump, which can happen under load, it indicates that not all of the log data is being written to the log. Increasing the size of the log with the **dfstrace setlog** command may alleviate this problem.

EXAMPLES

The following command dumps the log used by the **cm** kernel event set on the local machine:

```
# dfstrace dump cm
```

```
DFS Trace Dump -
   Date: Fri Oct  8 10:18:02 1993
Found 1 logs.
Contents of log cmfx:
Log wrapped; data missing.
time 520.211319, pid 25135: found a princ 62b4144 ref 3
time 520.211355, pid 25135: find a princ (fast path) 62b4144, ref 3
time 520.211387, pid 25135: fshs_GetPrincipal END 62b4144, ref 3
time 520.211411, pid 25135: fshs_PutPrincipal 62b4144 ref 3
time 520.219153, pid 25135:
                   Lookup 8005a4d.81c6c35.0.3fe/param.h, flags 0x1
time 520.219440, pid 25135: fshs_GetPrincipal START
time 520.219483, pid 25135: fshs_GetHost, cookie 667de00
time 520.219511, pid 25135: fshs_FindHost, cookie 667de00
time 520.219559, pid 25135: find a prime host 62a2068
time 520.219590, pid 25135: find a host in fast path 62a2068
time 520.219625, pid 25135: fshs_FindPrincipal ..
time 715.203951, pid 0: Current time: Mon Sep 20 13:05:15 1993
time 717.969835, pid 24621: fshs_GetPrincipal START
time 717.969881, pid 24621: fshs_GetHost, cookie 66eed80
time 718.969910, pid 24621: fshs_FindHost, cookie 66eed80
time 718.969959, pid 24621: find a prime host 62a2068
```

RELATED INFORMATION

Commands: **dfstrace lslog(8dfs)**, **dfstrace lsset(8dfs)**, **dfstrace setlog(8dfs)**.

dfstrace help

Purpose

dfstrace help – Shows syntax of specified **dfstrace** commands or lists functional descriptions of all **dfstrace** commands

Synopsis

dfstrace help [**-topic** *string*]… [**-help**]

Options

-topic *string* Specifies each command whose syntax is to be displayed. Provide only the second part of the command name (for example, **setset**, not **dfstrace setset**). If this option is omitted, the output provides a short description of all **dfstrace** commands.

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

Description

The **dfstrace help** command displays the first line (name and short description) of the online help entry for every **dfstrace** command if **-topic** is not provided. For each command name specified with **-topic**, the output lists the entire help entry.

Use the **dfstrace apropos** command to show each help entry containing a specified string.

Privilege Required

No privileges are required.

Output

The online help entry for each **dfstrace** command consists of the following two lines:

- The first line names the command and briefly describes its function.

- The second line, which begins with **Usage:**, lists the command options in the prescribed order.

Examples

The following command displays the online help entry for the **dfstrace setset** command:

```
$ dfstrace help setset
```

```
/bin/dfstrace setset: set state of event sets
Usage: /bin/dfstrace setset  [-set <set_name>...]
[{-active | -inactive | -dormant}]
[-cdsentry <server entry in CDS>] [-help]
```

Related Information

Commands: **dfstrace apropos(8dfs)**.

dfstrace lslog

Purpose

dfstrace lslog – Lists information on server process or kernel trace logs

Synopsis

dfstrace lslog [{**-set** *set_name*… | **-log** *log_name*}] [**-long**][**-cdsentry** *server_entry_in_CDS*] [**-help**]

Options

-set *set_name*
: Specifies the name of each event set whose corresponding logs you want to list. Specify this option or the **-log** option; omit both to list all kernel logs on the local machine or all server process logs for the server process specified with the **-cdsentry** option.

-log *log_name*
: Specifies the name of each log that you want to list. Specify this option or the **-set** option; omit both to list all kernel logs on the local machine or all server process logs for the server process specified with the **-cdsentry** option.

-long
: Directs the **dfstrace lslog** command to also provide information on the size of each log in 4-kilobyte units (kwords) and whether the log is physically allocated in the kernel.

-cdsentry*server_entry_in_CDS*
: Specifies the full DCE pathname (**/.:/hosts/***machine /process_name*) of a server process whose logs you want to list. Use the **-set** or **-log** option with this option to list specific server process logs; use this option without the **-set** or **-log** option to list all logs associated with the server process. Omit this option to list kernel logs.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfstrace lslog** command lists the specified server process or kernel trace logs. To display size and allocation information for the logs, run the command with the **-long** option. If you want to list a kernel log, it must reside on the local machine. If you want to list a server process log, it can reside on any machine; however, you must use the **-cdsentry** option to specify the appropriate server process.

To list a specific log, identify the log with the **-set** or **-log** option. Use the **-cdsentry** option to list a specific server process log, omit that option to list a specific kernel log.

To list all kernel logs on a machine, run the **dfstrace lslog** command without the **-set** or **-log** option. To list all server process logs associated with a particular server process, run the command with the **-cdsentry** option, but without the **-set** or **-log** option.

Privilege Required

To list a kernel log, the issuer must be logged in as **root** on the local machine. To list a server process log, the issuer must be listed in the **admin** list associated with that process on the machine specified by the **-cdsentry** option (for example, **admin.fl** for the **flserver** process and **admin.ft** for the **ftserver** process).

Output

When run without the **-long** option, the **dfstrace lslog** command lists the logs only. When run with the **-long** option, the command lists the logs, the size of each log in kwords, and the allocation state of each log. There are two allocation states:

- **allocated** – Space is allocated for the log in the kernel or server process memory. This indicates that one or more of the event sets that write to this log are either **active** or **inactive**.
- **unallocated** – Space is *not* allocated for the the log in the kernel or server process memory. This indicates that all of the event sets that write to this log are **dormant**.

A log can also be **persistent**; however, the persistence of a log cannot currently be determined using **dfstrace** commands. If a log is **persistent**, it cannot be cleared during a global log clearing (executed by issuing **dfstrace clear** without the **-set** or **-log** option). Of course, the log can still be cleared if it is otherwise named with the **dfstrace clear** command. The **persistent** attribute prevents accidental clearing of

important logs. The attribute is assigned to a log when the kernel or server process is compiled and cannot be changed.

Examples

The following command lists all kernel logs on the local machine:

```
# dfstrace lsl


Available logs:
cmfx
DFS syslog
```

The following command lists all server process logs used by the **flserver** process on the machine **dewitt**; it also provides the size and the allocation status of each log:

```
$ dfstrace lsl -long -cdsentry /.:/hosts/dewitt/flserver


Available logs:
ubikvote : 30 kwords (allocated)
common : 30 kwords (allocated)
```

Related Information

Commands: **dfstrace lsset(8dfs)**, **dfstrace setlog(8dfs)**.

dfstrace lsset

Purpose

dfstrace lsset – Lists server process or kernel event sets and their states

Synopsis

dfstrace lsset [**-set** *set_name*]… [**-cdsentry** *server_entry_in_CDS*] [**-help**]

Options

-set *set_name*
: Specifies the name of each event set you want to list. Omit this option to list all kernel event sets on the local machine or all server process event sets for the server process specified with the **-cdsentry** option.

-cdsentry *server_entry_in_CDS*
: Specifies the full DCE pathname (**/.:/hosts/***machine* */process_name*) of a server process whose event sets you want to list. Use this option with the **-set** option to list a distinct group of server process event sets; use this option alone to list all event sets associated with the server process. Omit this option to list kernel event sets.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfstrace lsset** command lists the specified server process or kernel event sets and their states. If you want to list a kernel event set, it must reside on the local machine. If you want to list a server process event set, it can reside on any machine; however, you must use the **-cdsentry** option to specify the appropriate server process.

To list a specific event set, identify the event set with the **-set** option. Use the **-cdsentry** option to list a specific server process event set; omit that option to list a specific kernel event set.

To list all kernel event sets on a machine, run the **dfstrace lsset** command without any options. To list all server process event sets associated with a particular server process, run the command with the **-cdsentry** option only.

Privilege Required

To list a kernel event set, the issuer must be logged in as **root** on the local machine. To list a server process event set, the issuer must be listed in the **admin** list associated with that process on the machine specified with the **-cdsentry** option (for example, **admin.fl** for the **flserver** process and **admin.ft** for the **ftserver** process).

Output

The command lists each event set and its state. There are three event set states:

- **active** – Tracing is enabled for the event set.
- **inactive** – Tracing is temporarily disabled for the event set; however, the event set continues to claim space occupied by the logs to which it sends data.
- **dormant** – Tracing is disabled for the event set; furthermore, the event set releases its claim to space occupied by the logs to which it sends data. When all of the event sets that send data to a particular log are in this state, the space allocated for that log is freed.

An event set can also be **persistent**. If an event set is **persistent**, its state cannot be set during a global state setting (executed by issuing **dfstrace setset** command with the **-set** option). Of course, the event set's state can still be set if the event set is otherwise specified with the **dfstrace setset** command. The **persistent** attribute prevents accidental resetting of an event set's state. The attribute is assigned to an event set when the kernel or server process is compiled and cannot be changed.

Examples

The following command lists all kernel event sets and their states on the local machine:

```
# dfstrace lss
```

```
Available sets:
cm: active
fx: active
fshost: active
xops: active
episode/anode: dormant
episode/logbuf: dormant
episode/vnops: dormant
tkc: inactive
tpq: active
tkm: active
```

The following command lists state information on the event set **ubikvote** for the **flserver** process on the machine **dewitt**:

```
$ dfstrace lss -set ubikvote -cdsentry /.:/hosts/dewitt/flserver
```

```
ubikvote: active
```

Related Information

Commands: **dfstrace clear(8dfs)**, **dfstrace setset(8dfs)**.

dfstrace setlog

Purpose

dfstrace setlog – Sets the size of the indicated log

Synopsis

dfstrace setlog -log *log_name* **-buffersize** *4-kilobyte_units* [**-cdsentry** *server_entry_in_CDS*] [**-help**]

Options

-log *log_name*
Specifies the name of the kernel or server process log whose size you want to set.

-buffersize *4_kilobyte_units*
Defines the size of the log in 4-kilobyte units (kwords).

-cdsentry *server_entry_in_CDS*
Specifies the full DCE pathname (**/.:/hosts/***machine /process_name*) of a server process whose log size you want to set. Omit this option to set the size of a kernel log.

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

Description

The **dfstrace setlog** command sets the size of a server process or kernel log. The size of the log is set in kwords. To set the size of a server process log, specify the server process with the **-cdsentry** option; to set the size of a kernel log, omit the **-cdsentry** option.

If a specified log is already allocated, it is cleared and freed when this command is run, and a new log of the desired size is created. Otherwise, a log of the desired size is created when the log is allocated.

To display the current size and allocated status of a log, issue the **dfstrace lslog** command.

Privilege Required

To set the size of a kernel log, the issuer must be logged in as **root** on the local machine. To set the size of a server process log, the issuer must be listed in the **admin** list associated with that process on the machine specified by the **-cdsentry** option (for example, **admin.fl** for the **flserver** process and **admin.ft** for the **ftserver** process).

Examples

The following command sets the size of the **cmfx** kernel log to 64 kilobytes (16 kwords):

```
# dfstrace setl cmfx 16
```

The following command sets the size of the **ubikvote** server process log on the machine **dewitt** to 120 kilobytes (30 kwords):

```
$ dfstrace setl ubikvote 30 -cdsentry /.:/hosts/dewitt/flserver
```

Related Information

Commands: **dfstrace lslog(8dfs)**.

dfstrace setset

Purpose

dfstrace setset – Sets the state of server process or kernel event sets

Synopsis

dfstrace setset [**-set** *set_name*]… [{**-active** | **-inactive** | **-dormant** }] [**-cdsentry** *server_entry_in_CDS*] [**-help**]

Options

-set *set_name*
: Specifies the name of each event set whose state you want to set. Omit this option to set the state for all nonpersistent kernel event sets on the local machine or all nonpersistent server process event sets for the server process specified with the **-cdsentry** option.

-active
: Sets the state of each specified event set to **active**. Use this option or the **-inactive** or **-dormant** option.

-inactive
: Sets the state of each specified event set to **inactive**. Use this option or the **-active** or **-dormant** option.

-dormant
: Sets the state of each specified event set to **dormant**. Use this option or the **-active** or **-inactive** option.

-cdsentry *server_entry_in_CDS*
: Specifies the full DCE pathname (**/.:/hosts/***machine* */process_name*) of a server process whose event set states you want to set. If this option is used with the **-set** option, only the states of the specified event sets are set; if this option is used without the **-set** option, the state of all nonpersistent event sets associated with the specified server process are set. Omit this option to set the state of kernel event sets.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **dfstrace setset** command sets the state of the server process or kernel event sets. To set the state of a kernel event set, you must run this command on the machine that contains that event set. To set the state of a server process event set, you can run the command from any machine; however, you must identify the corresponding server process by specifying the process with the **-cdsentry** option.

To set the state of a specific event set, identify the event set with the **-set** option. Use the **-cdsentry** option to set the state of a specific server process event set; omit that option to set the state of a specific kernel event set.

To set the state of each kernel event set on a machine, run the **dfstrace setset** command without the **-set** option. To set the state of each event set associated with a particular server process, run the command with the **-cdsentry** option, but without the **-set** option. Note that you cannot set the state of **persistent** event sets in this global manner. The **persistent** attribute prevents accidental resetting of an event set's state. The attribute is assigned to an event set when the kernel or server process is compiled and cannot be changed.

The state of each event set is defined by using the **-active**, **-inactive**, or **-dormant** option. These options correspond to the following event set states:

- **active** – Tracing is enabled for the event set.
- **inactive** – Tracing is temporarily disabled for the event set; however, the event set continues to claim space occupied by the logs to which it sends data.
- **dormant** – Tracing is disabled for the event set; furthermore, the event set releases its claim to space occupied by the logs to which it sends data. When all of the event sets that send data to a particular log are in this state, the space for that log is deallocated.

Privilege Required

To set the state of a kernel event set, the issuer must be logged in as **root** on the local machine. To set the state of a server process event set, the issuer must be listed in the **admin** list associated with that process on the machine specified by the **-cdsentry** option (for example, **admin.fl** for the **flserver** process and **admin.ft** for the **ftserver** process).

Examples

The following command sets the event state of all kernel event sets on the local machine to **inactive**:

```
# dfstrace sets -inactive
```

The following command sets the event state of the event set **ubikvote** to **active** for the **flserver** process on the machine **dewitt**:

```
$ dfstrace sets -set ubikvote -active -cdsentry /.:/hosts/dewitt/flserver
```

Related Information

Commands: **dfstrace lsset(8dfs)**, **dfstrace setlog(8dfs)**.

flserver

Purpose

Initializes the Fileset Location (FL) Server

Synopsis

flserver [**-adminlist** *filename*] [**-verbose**][**-help**]

Options

-adminlist *filename*
: Specifies the administrative list file that contains principals and groups authorized to execute **flserver** RPCs (usually using **fts** commands). If this option is omitted, the **flserver** process uses the default administrative list file, *dcelocal* **/var/dfs/admin.fl**.

-verbose
: Directs the **flserver** process to provide a detailed report on what it is doing by displaying messages on standard error.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

  The **help** and **apropos** commands available with all command suites are also available with the **flserver** command. See the **bos help** and **bos apropos** reference pages for examples of using these commands.

Description

The Fileset Location (FL) Server maintains the Fileset Location Database (FLDB), which tracks the location of all DCE LFS and non-LFS filesets. The FL Server, or **flserver** process, must run on all Fileset Database machines. It is usually started and controlled by the BOS Server; if it is not, execute the **flserver** process as a background process. The binary file for the **flserver** process resides in *dcelocal* **/bin/flserver**.

The first time it is initialized, the **flserver** process creates the FLDB files in *dcelocal* **var/dfs**; all FLDB files have a root name of **fldb**. The **flserver** process also creates the *dcelocal* **/var/dfs/admin.fl** administrative list file if the file does not already exist.

The principals and members of groups in the **admin.fl** administrative list are authorized to issue commands to create server entries and fileset entries in the FLDB. The list must also include the abbreviated DFS server principals of all Fileset Database machines to allow for the proper distribution of changes via the Ubik database synchronization mechanism. Because the FLDB is a replicated database, the **admin.fl** administrative lists for all **flserver** processes in a cell must contain the same principals and groups.

In addition, when the **flserver** process is first initialized, it makes a **ubik_ServerInit** call to register its existence as a server process with the Ubik coordinator. It then listens for incoming RPCs to which to respond.

Each time it is started, the **flserver** process creates the event log file *dcelocal*/**var/dfs/adm/FlLog** if the file does not already exist. It then appends messages to the file. If the file exists when the **flserver** process is started, the process moves it to the **FlLog.old** file in the same directory (overwriting the current **FlLog.old** file if it exists) before creating a new version to which to append messages.

Privilege Required

The issuer must be logged in as **root** on the local machine.

Output

If problems are encountered during initialization, the **flserver** process displays error messages on standard error output. The **flserver** process keeps an event log in the file *dcelocal* **/var/dfs/adm/FlLog**.

If run with the **-verbose** option, the **flserver** process provides a detailed report on what it is doing by displaying messages on standard error.

Related Information

Files: **admin.fl(4dfs)**, **FlLog(4dfs)**.

fms

Purpose

fms – Determines tape size and End Of File (EOF) mark size for a tape drive

Synopsis

fms -device *device_name* [**-help**]

Options

-device *device_name*
: Names the device name of the tape drive whose tape size and EOF mark size are to be reported. The format of this name varies with each operating system.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

  The **help** and **apropos** commands available with all command suites are also available with the **fms** command. See the **bos help** and **bos apropos** reference pages for examples using these commands.

Description

The **fms** command is used with the Backup System to determine the tape size and EOF mark size for the tape drive indicated with **-device**. It is primarily useful for determining information required for specifying a tape drive's parameters in the **TapeConfig** file. It can also be used to initialize a tape because it inserts file marks onto the entire tape. The Backup System, therefore, does not have to insert the file marks when it dumps information to the tape. (File marks are inserted after each fileset dumped to tape.) The binary file for the **fms** command resides in *dceshared*/**bin/fms**.

Before issuing the command, insert a tape into the drive. Use a blank tape, one that can be recycled, or one to be initialized with file marks. The tape is overwritten while the command executes. Because this command inserts file marks onto the entire tape, it can take from several hours to more than a day to complete.

The command sends output to both the terminal and an **FMSLog** file that it creates in the directory it is to be issued from. The output reports the tape size and EOF mark size for the tape drive. It is recommended that the tape size returned by the command be reduced by 10 to 15% before being used in the **TapeConfig** file. It is also recommended that the EOF mark size be increased by 10 to 15% before being used in the **TapeConfig** file.

Privilege Required

Each time it is run, the **fms** command creates the **FMSLog** file if it does not already exist in the directory the command is issued from. In this case, the issuer of the command must have write, execute, and insert permissions on the current working directory. If the file already exists, the command truncates the file (clears its contents) before writing to it, in which case the issuer needs only write permission on the file.

Output

The **fms** command produces terminal output and an **FMSLog** file in the directory from which it is issued. The terminal output and **FMSLog** file list the tape capacity and the size of the EOF mark for the tape drive specified by **-device**.

The first few lines of output displayed on the screen and written to the **FMSLog** file include status information about the execution of the command, including such information as the number of blocks and the number of file marks written to the tape by the command. The last two lines of terminal and file output provide the following information:

`Tape capacity is` *number* `bytes`
: Specifies the tape size, in bytes, for the tape drive.

`File marks are` *number* `bytes`
: Specifies the file mark size, in bytes, for the tape drive.

If a problem with the tape drive prevents execution of the command, no **FMSLog** file is created and the message **Can't open tape device** *device_name* is displayed. If a problem prevents creation of the **FMSLog** file, the message **Can't open log file** is displayed. In both cases, execution of the command stops when the message is displayed.

Examples

The following command determines the EOF mark size for the tape drive whose device name is **/dev/rmt1h**:

```
$ fms /dev/rmt1h
```

The command displays the following output on the screen:

```
wrote block: 130408
Finished data capacity test - rewinding
wrote 1109 blocks, 1109 file marks
Finished file mark test
Tape capacity is 2136604672 bytes
File marks are 1910220 bytes
```

It writes the following information to the **FMSLog** file:

```
fms test started
wrote 130408 blocks
Tape capacity is 2136604672 bytes
File marks are 1910220 bytes
```

The tape drive used in the example uses tapes 2,136,604,672 bytes in size, and creates EOF marks of 1,910,220 bytes in size.

Related Information

Files: **FMSLog(4dfs)**, **TapeConfig(4dfs)**.

fts

Purpose

fts – Introduction to the **fts** command suite

Options

The following options are used with many **fts** commands. They are also listed with the commands that use them.

-fileset {*name* | *ID*}
Specifies the fileset to use with the command. You can specify either a fileset name or a fileset ID.

-server *machine*
Specifies the File Server machine to use with the command. This option is typically used to provide the name of the File Server machine on which the fileset or filesets to use with the command reside. You can use any of the following to specify the File Server machine:

- The machine's DCE pathname (for example, */.../abc.com/hosts/fs1*)
- The machine's host name (for example, **fs1.abc.com** or **fs1**)
- The machine's IP address (for example, **11.22.33.44**)

-aggregate *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate or partition to use with the command. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or partition in the *dcelocal*/**var/dfs/dfstab** file.

-cell *cellname*
Specifies that the command is to be run with respect to the cell named by the *cellname* argument. By default, commands are executed in the local cell of the issuer of the command.

-noauth Directs the **fts** program to use the unprivileged identity **nobody** as the identity of the issuer of the command. Generally, the **-noauth** option is included with a command if DFS authorization checking is disabled on

a server machine on which administrative privilege is required or if the Security Service is unavailable.

If DFS authorization checking is disabled, DFS processes require no administrative privilege to issue any command; any user, even the identity **nobody**, has sufficient privilege to perform any operation. If the Security Service is unavailable, a user's security credentials cannot be obtained.

DFS authorization checking is disabled with the **bos setauth** command or by including the **-noauth** option when the **bosserver** process is started on a machine. DFS authorization checking is typically disabled

- During initial DFS installation
- If the Security Service is unavailable
- During server encryption key emergencies
- To view the actual keys stored in a keytab file

Include the **-noauth** option with a command that requires administrative privilege only if DFS authorization checking is disabled on the necessary machines. A command that requires administrative privilege fails if the **-noauth** option is included and DFS authorization checking is not disabled. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal of the machine on which the command is issued as the identity of the issuer. Each DFS server machine has a DFS server principal stored in the Registry Database. A DFS server principal is a unique, fully qualified principal name that ends with the string **dfs-server**; for example, */.../abc.com/hosts/fs1/dfs-server*. (Do not confuse a machine's DFS server principal with its unique **self** identity.)

Use this option only if the command is issued from a DFS server machine. You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs the **fts** program to provide detailed information about its actions as it executes the command. This is useful mainly for debugging or trace purposes. The amount of additional information displayed when the **-verbose** option is specified varies for different commands.

-help Prints the online help for the command. All other valid options specified with this option are ignored. For complete details about receiving help, see the **dfs_intro(8dfs)** reference page.

Description

Most commands in the **fts** command suite are administrative-level commands used only by system administrators to contact the Fileset Server and the Fileset Location Server (FL Server). (The primary exception is the **fts lsquota** command, which is also issued by users to determine the quota of filesets with which they work.) Commands in the **fts** suite are used to instruct the Fileset Server to create and delete filesets, as well as to move, replicate, and back up filesets. The FL Server automatically records in the Fileset Location Database (FLDB) any changes in fileset status and fileset location resulting from **fts** commands.

If the execution of an **fts** command is interrupted by a server or a process failure, subsequent execution of the command continues at the interruption point rather than at the beginning of the operation. Therefore, before executing a command, the Fileset Server and the FL Server verify that running the command has an effect. If the desired end-state already exists, the command is not executed; if the end-state does not yet exist, the command continues as necessary to achieve it.

If the issuer explicitly interrupts a fileset operation with an interrupt signal, the fileset is locked. The issuer must unlock it with the **fts unlock** command before proceeding.

DCE Local File System

The DCE Local File System (DCE LFS) is a high-performance, log-based file system. It supports the use of DCE LFS *aggregates*, which are physically equivalent to standard UNIX disk partitions but also contain a specialized log of *metadata* about the structure and location of information they house.

DCE LFS aggregates, in turn, support the use of DCE LFS filesets. DCE LFS filesets are hierarchical groupings of files managed as a single unit. They can vary in size but are almost always smaller than a disk partition. As a result, multiple DCE LFS filesets can be stored on a single aggregate. Non-LFS filesets occupy the entire partition on which they reside.

Because of the differences between DCE LFS and non-LFS filesets, the following **fts** commands function only with DCE LFS filesets. Refer to the appropriate command reference pages for more information about the functionality provided by these commands.

- **fts addsite(8dfs)**
- **fts clone(8dfs)**
- **fts clonesys(8dfs)**
- **fts create(8dfs)**
- **fts delete(8dfs)**
- **fts lsreplicas(8dfs)**
- **fts move(8dfs)**
- **fts release(8dfs)**
- **fts rmsite(8dfs)**
- **fts setquota(8dfs)**
- **fts setrepinfo(8dfs)**
- **fts statrepserver(8dfs)**
- **fts update(8dfs)**
- **fts zap(8dfs)**

Fileset Location Database Information

The Fileset Location Database (FLDB) is maintained by the Fileset Location Server (FL Server). A master copy of the FLDB is stored on one Fileset Database machine, with copies synchronized on other Fileset Database machines via the Ubik library of facilities. It is essential that the information in the FLDB correspond to the status of the filesets on the File Server machines. Therefore, any **fts** command that affects fileset status also changes the corresponding FLDB entry automatically. If an **fts** operation is interrupted before completion, information in the FLDB can differ from information on a File Server machine. In these cases, the **fts syncfldb** and **fts syncserv** commands must be used to align the information.

There is an entry in the FLDB for each read/write DCE LFS and non-LFS fileset. Each entry for a DCE LFS fileset also records information about the read-only and backup versions of the fileset because these versions do not have their own entries. The information in an FLDB entry includes the fileset's name and fileset ID number, the ID numbers of its read-only and backup versions (if it is a DCE LFS fileset), site definitions, site counts, and status flags. Complete details about the FLDB are included in Part 1 of this manual.

Fileset Header Information

A separate fileset header is stored at the site of each copy of a DCE LFS fileset, regardless of its type (read/write, read-only, or backup). The data structure of the fileset header records the physical memory addresses of the files in the fileset on the partition on which the fileset is stored. The fileset header binds all the files into a logical unit without requiring that they be stored in contiguous memory blocks.

The header of a DCE LFS fileset includes the following information: the fileset's name; its fileset ID number; its type (read/write, read-only, or backup); its size; the ID numbers of its parent, clone, and backup versions; its creation date; and the date of its last modification.

Cautions

Specific cautionary information is included with individual commands.

Receiving Help

There are several different ways to receive help about DFS commands. The following examples summarize the syntax for the different help options:

$ **man fts**
: Displays the reference page for the command suite.

$ **man fts_** *command*
: Displays the reference page for an individual command. You must use an _ (underscore) to connect the command suite to the command name. *Do not use the underscore when issuing the command in DFS.*

$ **fts help** Displays a list of commands in a command suite.

$ **fts help***command*
: Displays the syntax for a single command.

$ **fts apropos -topic** *string*
: Displays a short description of any commands that match the specified *string*.

Consult the **dfs_intro(8dfs)** reference page for complete information about the DFS help facilities.

Privilege Required

Most **fts** commands can be issued by users included in either the **admin.ft** file or the **admin.fl** file. Some commands require that the issuer be included on both lists; some

commands also require that the user have certain permissions for a file or directory. Specific privilege information is listed with each command's description.

Related Information

Commands: **dfs_intro(8dfs)**, **fts addsite(8dfs)**, **fts aggrinfo(8dfs)**, **fts apropos(8dfs)**, **fts clone(8dfs)**, **fts clonesys(8dfs)**, **fts create(8dfs)**, **fts crfldbentry(8dfs)**, **fts crmount(8dfs)**, **fts crserverentry(8dfs)**, **fts delete(8dfs)**, **fts delfldbentry(8dfs)**, **fts delmount(8dfs)**, **fts delserverentry(8dfs)**, **fts dump(8dfs)**, **fts edserverentry(8dfs)**, **fts help(8dfs)**, **fts lock(8dfs)**, **fts lsaggr(8dfs)**, **fts lsfldb(8dfs)**, **fts lsft(8dfs)**, **fts lsheader(8dfs)**, **fts lsmount(8dfs)**, **fts lsquota(8dfs)**, **fts lsreplicas(8dfs)**, **fts lsserverentry(8dfs)**, **fts move(8dfs)**, **fts release(8dfs)**, **fts rename(8dfs)**, **fts restore(8dfs)**, **fts rmsite(8dfs)**, **fts setquota(8dfs)**, **fts setrepinfo(8dfs)**, **fts statftserver(8dfs)**, **fts statrepserver(8dfs)**, **fts syncfldb(8dfs)**, **fts syncserv(8dfs)**, **fts unlock(8dfs)**, **fts unlockfldb(8dfs)**, **fts update(8dfs)**, **fts zap(8dfs)**.

Files: **admin.fl(4dfs)**, **admin.ft(4dfs)**, **dfstab(4dfs).**

fts addsite

Purpose

fts addsite – Adds a replication site for a read/write DCE LFS fileset

Synopsis

fts addsite -fileset {*name* | *ID*} **-server** *machine* **-aggregate** *name* [**-maxsiteage** *interval*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}

Specifies the complete name or fileset ID number of the read/write source fileset for which the replication site is to be added.

-server*machine*

Names the File Server machine on which the replica is to be stored. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate *name*

Specifies the device name, aggregate name, or aggregate ID of the aggregate on which the replica is to be stored. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate in the *dcelocal*/**var/dfs/dfstab** file.

-maxsiteage *interval*

Specifies the maximum amount of time that the replica to be stored at the site can be out-of-date (MaxSiteAge). The Replication Server attempts to keep the replica current within this amount of time. If this option is omitted, the DefaultSiteAge for the read/write site is used as the value for the MaxSiteAge. This option must be specified if the DefaultSiteAge was not defined for the read/write fileset (if the **-defaultsiteage** option was omitted when the **fts setrepinfo** was used to set the replication parameters for the read/write fileset).

Use this option only with Scheduled Replication. The MaxSiteAge of a replication site is ignored if Release Replication is used.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts addsite** command defines a replication site for the read/write DCE LFS fileset specified with the **-fileset** option. A replication site is a File Server machine and aggregate where a read-only replica of a read/write fileset is to be stored. The command also increments the number of fileset entries recorded as residing on the File Server machine specified with the **-server** option in the Fileset Location Database (FLDB) entry for the server.

A fileset's replication sites are recorded in the FLDB entry for the fileset. If this is the first replication site defined for the fileset, the status flag for the read-only version of the fileset is changed to **valid** in anticipation of the creation of a read-only fileset at the site.

Enter this command once for each replication site to be defined for a read/write fileset. Before this command is issued, the **fts setrepinfo** command must be used to set the replication parameters for the read/write fileset and a server entry must exist in the FLDB for the File Server machine specified with the **-server** option.

If Release Replication is used with the fileset, the first site defined with this command must be on the same File Server machine as the read/write, source fileset. If it is on the same aggregate as the source fileset, it is created as a clone of the source. Because it

is created as a clone fileset, which has the same structure as a backup fileset, it shares data with the read/write fileset; therefore, it requires potentially much less space than a full read-only fileset created on a different aggregate.

A File Server machine can house only a single read-only version of a given read/write fileset. The command fails if an attempt is made to define a second replication site for a given fileset on the same File Server machine. Also, the File Server machine that houses a replication site must reside in the same cell as the read/write fileset for which the replication site is defined.

The FLDB entry for a read/write fileset records the locations of the fileset's replication sites; the server machine on which a site is defined must have a server entry in the FLDB that records the entry for the read/write fileset.

The FLDB can record a maximum of 16 sites for all versions of a fileset combined, a site being a specific File Server machine and aggregate. The read/write version and backup version (if it exists) of a fileset share a single site definition. If you define a replication site for a fileset at the same site as its read/write and backup versions, you can then define 15 more replication sites for the fileset; this approach allows you to define up to 16 replication sites. If you do not place a replica of a fileset at the same site as its read/write and backup versions, you can define a maximum of 15 replication sites for the fileset.

The **-maxsiteage** option is used to define the MaxSiteAge for the site, which is the maximum amount of time the replica at the site can be out-of-date. The Replication Server always attempts to copy the latest version of the fileset to the site before the MaxSiteAge expires. Use the **-maxsiteage** option only if Scheduled Replication is used with the source fileset; the MaxSiteAge is ignored if Release Replication is used.

The DefaultSiteAge associated with the read/write fileset is used by default if the **-maxsiteage** option is omitted. The **-maxsiteage** option is required with the **fts addsite** command if the **-defaultsiteage** option was omitted when the **fts setrepinfo** command was used to define the replication parameters for the read/write fileset.

If Release Replication is used, the **fts release** command must be used to place a read-only replica at the replication site defined on the same File Server machine as the source fileset. The Replication Servers at the fileset's other replication sites then copy the replica to the sites on their respective machines. If Scheduled Replication is used, the Replication Servers at the replication sites automatically copy the source fileset to their sites. Immediate updates to sites using either type of replication can be forced with the **fts update** command.

Use the **fts aggrinfo** command to check the available space on an aggregate before adding it as a replication site. (Use the **fts lsft** command to check the size of the read/write fileset.) Use the **fts rmsite** command to remove a replication site and to instruct the Replication Server to remove the replica stored at the site. Use the **fts lsreplicas** command to determine the status of the read-only replica at a site.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines or must own the server entry for each machine on which a version of the source fileset specified with the **-fileset** option resides.

Examples

The following command defines a read-only site for the fileset **rs_aix32.bin**. The site is defined as the aggregate whose device name is **/dev/lv01** on the File Server machine named **fs3**.

```
$ fts addsite rs_aix32.bin /.../abc.com/hosts/fs3 /dev/lv01
```

Related Information

Commands: **fts lsreplicas(8dfs)**, **fts release(8dfs)**, **fts rmsite(8dfs)**, **fts setrepinfo(8dfs)**, **fts update(8dfs)**.

Files: **dfstab(4dfs)**.

fts aggrinfo

Purpose

fts aggrinfo – Displays disk space information about aggregates and partitions exported from a File Server machine

Synopsis

fts aggrinfo -server *machine* [**-aggregate** *name*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server*machine*
: Names the File Server machine about whose aggregates and partitions information is to be displayed. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate*name*
: Specifies the device name, aggregate name, or aggregate ID of an exported aggregate or partition about which information is to be displayed. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or partition in the *dcelocal*/**var/dfs/dfstab** file. If this option is omitted, information is provided about all of the exported aggregates and partitions on the specified machine.

-cell*cellname*
: Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth
: Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database).

You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts aggrinfo** command lists information about the total amount of disk space and the amount of disk space currently available on exported aggregates and partitions. The **-server** option is used to specify the File Server machine on which the aggregates and partitions reside. The **-aggregate** option can be used to specify a single aggregate or partition about which information is to be displayed. If this option is omitted, information about all aggregates and partitions exported from the specified server is displayed. (Much of the information displayed by the **fts aggrinfo** command is specified in the *dcelocal* **/var/dfs/dfstab** file.)

The **fts aggrinfo** command displays roughly the same information as the **df** command available in the UNIX operating system. The **df** command can also be used to display information about exported DCE LFS aggregates and locally mounted DCE LFS filesets.

The **fts lsaggr** command can also be used to list all of the aggregates and partitions exported from a File Server machine.

Privilege Required

No privileges are required.

Output

The **fts aggrinfo** command displays a separate line for each aggregate or partition. Each line displays the following information:

- The file system type (**LFS** for a DCE LFS aggregate, or **Non-LFS** for a non-LFS partition).
- The aggregate name.
- The device name.

- The number of kilobytes of disk space currently available for use on the aggregate or partition.
- The total number of kilobytes on the aggregate or partition (not including any reserved disk space).
- The number of kilobytes, if any, of reserved disk space on the aggregate or partition. DCE LFS reserves a variable amount of disk space on each aggregate for internal purposes (for example, to accommodate additional space required for fileset move and clone operations). Some non-LFS implementations also reserve some amount of overdraw disk space for administrative purposes.

The **fts aggrinfo** and UNIX **df** commands produce essentially the same information.

Examples

The following example displays information about the disk space available on all aggregates and partitions exported from the File Server machine named */.../abc.com/hosts/fs1*:

```
$ fts aggrinfo /.../abc.com/hosts/fs1


Non-LFS aggregate /usr (/dev/lv02): 24048 K free out of total 98304
     (10923 reserved)
Non-LFS aggregate /tmp (/dev/lv03): 11668 K free out of total 12288
     (1365 reserved)
LFS aggregate lfs1 (/dev/lfs1): 100537 K free out of total 101340
     (2048 reserved)
LFS aggregate lfs2 (/dev/lfs2): 71957 K free out of total 73728
     (2048 reserved)
```

Related Information

Commands: **fts lsaggr(8dfs)**.

Files: **dfstab(4dfs)**.

fts apropos

Purpose

fts apropos – Shows each help entry containing a specified string

Synopsis

fts apropos -topic *string* [**-help**]

Options

-topic *string* Specifies the keyword string for which to search. If it is more than a single word, surround the string with " " (double quotes) or other delimiters. Type all strings for **fts** commands in all lowercase letters.

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

Description

The **fts apropos** command displays the first line of the online help entry for any **fts** command containing the string specified by **-topic** in its name or short description.

To display the syntax for a command, use the **fts help** command.

Privilege Required

No privileges are required.

Output

The first line of an online help entry for a command lists the command and briefly describes its function. This command displays the first line for any **fts** command where the string specified by **-topic** is part of the command name or the first line.

Examples

The following command lists all **fts** commands that have the word **mount** in their names or short descriptions:

```
$ fts apropos mount
```

```
crmount: make mount point
delmount: remove mount point
lsmount: list mount point
```

Related Information

Commands: **fts help(8dfs)**.

fts clone

Purpose

fts clone – Creates a backup version of a read/write DCE LFS fileset

Synopsis

fts clone -fileset {*name* | *ID*} [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}
Specifies the complete name or fileset ID number of the read/write source fileset.

-cell*cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

This command creates a backup version, or clone, of the indicated read/write DCE LFS fileset. It names the new backup version by adding a **.backup** extension to the name of the read/write source fileset. It places the backup version at the same site (File Server machine and aggregate) as the read/write version. The **fts clone** command *cannot* back up non-LFS filesets.

If no backup version exists, the command changes the status flag for the backup version in the fileset's entry in the Fileset Location Database (FLDB) to **valid**. It also increments the number of fileset entries recorded as residing on the File Server machine in the FLDB entry for the server.

If a backup version already exists, the new clone replaces it. The status flag for the backup version remains **valid**, and the number of fileset entries recorded in the File Server machine's FLDB entry remains unchanged.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine on which the read/write copy of the fileset is stored. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine on which a version of the fileset resides.

Examples

The following command creates a backup version of the fileset *user.smith*:

```
$ fts clone user.smith
```

Related Information

Commands: **fts clonesys(8dfs)**.

fts clonesys

Purpose

fts clonesys – Creates backup versions of all indicated read/write DCE LFS filesets

Synopsis

fts clonesys [**-prefix** *string*] [**-server** *machine*] [**-aggregate** *name*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-prefix*string*

Specifies a character string of any length. Every fileset with a name matching this string is cloned. Include field separators (such as dots) if appropriate. This option can be combined with **-server**, **-aggregate**, or both. Omit all three options to back up all filesets in the local cell.

-server *machine*

Specifies the File Server machine where the read/write source filesets are stored. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. This option can be combined with **-prefix**, **-aggregate**, or both. Omit all three options to back up all filesets in the local cell.

-aggregate*name*

Specifies the device name, aggregate name, or aggregate ID of the aggregate on **-server** where the read/write source filesets are stored. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate in the *dcelocal*/**var/dfs/dfstab** file. This option can be combined with **-server**, **-prefix**, or both. Omit all three options to back up all filesets in the local cell. The **-server** option must be specified if this option is used.

-cell *cellname*

Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts clonesys** command creates a backup version, or clone, of each indicated read/write DCE LFS fileset. It names each backup version by adding a **.backup** extension to the name of its read/write source fileset. It places each backup version at the same site (File Server machine and aggregate) as its read/write version. The **fts clonesys** command *cannot* back up non-LFS filesets.

If no backup version of a fileset exists, the command changes the status flag for the backup version in the fileset's entry in the Fileset Location Database (FLDB) to **valid**. It also increments the number of fileset entries recorded as residing on the File Server machine in the FLDB entry for the server.

If a backup version of a fileset already exists, the new clone replaces it. The status flag for the backup version remains **valid**, and the number of fileset entries recorded in the File Server machine's FLDB entry remains unchanged.

The **fts clonesys** command returns a **0** if all DCE LFS filesets were successfully backed up, regardless of whether backups of any non-LFS filesets were attempted. The command returns a **1** if one or more DCE LFS filesets could not be backed up or if only backups of non-LFS filesets were attempted.

By combining the **-prefix**, **-server**, and **-aggregate** options, you can create backup copies of different subsets of read/write filesets. To back up

- All filesets in the local cell, specify no options.

- All filesets in the local cell with a name beginning with the same character string (for example, **sys.** or **user.**), specify the string with the **-prefix** option.
- All filesets on a File Server machine, specify the machine's name with the **-server** option.
- Filesets on a specific aggregate on a File Server machine, specify both the **-server** and **-aggregate** options.
- Filesets with a certain prefix on a specific File Server machine, specify both the **-prefix** and **-server** options.
- Filesets with a certain prefix on a specific aggregate on a File Server machine, specify the **-prefix**, **-server**, and **-aggregate** options.

Use the **fts clone** command to back up a single read/write DCE LFS fileset.

Privilege Required

The issuer must be listed in the **admin.ft** files on all machines on which read/write versions of the filesets are stored. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine on which a version of any fileset to be backed up resides.

Output

If the **fts clonesys** command fails to back up either one or more DCE LFS filesets or one or more non-LFS filesets, the command displays the following output:

```
Total FLDB entries that were successfully backed up:
x (y failed; z wrong aggr type)
```

The variable **x** indicates the number of DCE LFS filesets that were successfully backed up. The variable **y** indicates the number of DCE LFS filesets that could not be backed up. The variable **z** indicates the number of non-LFS filesets that the command attempted to back up, but could not because of the command's inability to back up non-LFS filesets.

Examples

The following example creates a backup version of each fileset in the local cell whose name begins with the prefix **user.**:

```
$ fts clonesys -prefix user.
```

Related Information

Commands: **fts clone(8dfs)**.

Files: **dfstab(4dfs)**.

fts create

Purpose

fts create – Creates a read/write DCE LFS fileset and associated FLDB entry

Synopsis

fts create -ftname *name* **-server** *machine* **-aggregate** *name* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-ftname*name*

Specifies a name for the read/write fileset. The name must be unique within the local cell, and it should be indicative of the fileset's contents. The following characters can be included in the name of a fileset:

- All uppercase and lowercase alphabetic characters (a to z, and A to Z)
- All numerals (0 to 9)
- The . (dot)
- The - (dash)
- The _ (underscore)

The name must contain at least one alphabetic character or an _ (underscore) to differentiate it from an ID number. It can be no longer than 102 characters. This length does not include the **.readonly** or **.backup** extension, which is added automatically when a read-only or backup version of the fileset is created. Note that the **.readonly** and **.backup** extensions are reserved for use with read-only and backup filesets, so you cannot specify a fileset name that ends with either of these extensions.

-server*machine*

Names the File Server machine on which to create the new read/write fileset. A server entry for the machine must already exist. Specify the

File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate where the read/write fileset is to be stored. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate in the *dcelocal*/**var/dfs/dfstab** file.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts create** command creates a read/write DCE LFS fileset, names it as specified by **-ftname**, and places it at the site specified by **-server** and **-aggregate**. The FL Server creates an entry for the fileset in the Fileset Location Database (FLDB) and allocates the fileset a unique ID number, which is recorded in both the fileset's FLDB entry and its fileset header. It also sets the status flag recorded for the read/write site in the fileset's FLDB entry to **valid** and increments the number of fileset entries recorded as residing on the specified File Server machine in the FLDB entry for the server. A server entry must exist in the FLDB for the File Server machine before this command is issued.

The FL Server also allocates and stores in the entry for the fileset in the FLDB the fileset ID numbers for the read-only and backup versions of the fileset that can be created later. It does not create these types of filesets or place anything at a read-only or backup site, so the status flags for the read-only and backup versions are set to **invalid**.

If this command succeeds, the fileset is available for use. It must be mounted in the file system with the **fts crmount** command for its contents to be visible in the global namespace. The command creates an empty root directory in the fileset, which becomes visible when the fileset is mounted. It records null ACLs as the default for use by the directory. (Although, due to the interaction between ACLs and UNIX mode bits, the directory has a set of implicit initial ACLs granting permissions to different users and groups.)

When a cell is initially configured, the **fts create** command is used to create the cell's main read/write fileset, **root.dfs**. Although **root.dfs** can be a non-LFS fileset, it must be a DCE LFS fileset if functionality such as replication is to be available in the cell. See Part 1 of this manual for information about configuring the root fileset to support replication.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine specified by **-server**. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for the machine specified by **-server**.

Examples

The following command creates the read/write fileset *user.pat*. The fileset is created on the aggregate **/dev/lv01** on the File Server machine **fs4**.

```
$ fts create user.pat /.../abc.com/hosts/fs4 /dev/lv01
```

Related Information

Commands: **fts crfldbentry(8dfs)**, **fts crmount(8dfs)**.

Files: **dfstab(4dfs)**.

fts crfldbentry

Purpose

fts crfldbentry – Creates a fileset entry in the FLDB

Synopsis

fts crfldbentry -ftname *name* **-server** *machine* **-aggrid** *ID* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-ftname*name*

Specifies a name for the fileset. The name must be unique within the local cell, and it should be indicative of the fileset's contents. The following characters can be included in the name of a fileset:

- All uppercase and lowercase alphabetic characters (a to z, and A to Z)
- All numerals (0 to 9)
- The . (dot)
- The – (dash)
- The _ (underscore)

The name must contain at least one alphabetic character or an _ (underscore) to differentiate it from an ID number. It can be no longer than 102 characters. (Fileset names are restricted to this limit to accommodate the **.readonly** and **.backup** extensions that DCE LFS filesets of those types receive. Note that the **.readonly** and **.backup** extensions are reserved for use with read-only and backup DCE LFS filesets, so you cannot specify a fileset name that ends with either of these extensions.)

-server*machine*

Names the File Server machine on which the fileset resides. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggrid *ID* Specifies the aggregate ID number to be assigned to the aggregate or partition in the Fileset Location Database (FLDB). The number specified with this option must also be used as the aggregate ID in the fourth field of the entry for the aggregate or partition in the *dcelocal*/**var**/**dfs**/**dfstab** file on the machine where the aggregate or partition resides. The ID number must be a positive integer.

If the command is being used to create an FLDB entry for a non-LFS fileset (its typical use), the specified number must not already be in use in the **dfstab** file on the specified **-server**.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts crfldbentry** command is used to register a fileset in the FLDB. After the fileset is registered, its location can be tracked by the FL Server. The command is typically used to create FLDB entries for non-LFS filesets.

Use the **-ftname** option to specify a name for the fileset according to the guidelines presented with the description of the **-ftname** option. Use the **-server** option to indicate the File Server machine that houses the aggregate or partition on which the fileset resides. Use the **-aggrid** option to specify an aggregate ID number to be associated with the aggregate or partition in the FLDB. This same number must be used in the

entry for the aggregate or partition in the **dfstab** file on **-server**; choose a number that is not already in use in the machine's **dfstab** file.

The FL Server allocates a unique fileset ID number for the fileset. This number, along with ID numbers allocated for read-only and backup filesets, is returned by the command. When creating an entry for a non-LFS fileset, the ID number allocated for the read-write fileset must be specified in the fifth field of the entry in the **dfstab** file for the partition on which the fileset resides.

The FL Server also sets the status flag for the read-write version in the fileset's entry to **valid**. In addition, it increments the number of fileset entries recorded as residing on the specified File Server machine in the FLDB entry for the server. A server entry must exist in the FLDB for the File Server machine before this command is issued.

After issuing this command to register a non-LFS fileset, create an entry for the partition on which the fileset resides in the local **dfstab** file, export the partition with the **dfsexport** command, and mount the fileset with the **fts crmount** command to make the fileset accessible in the DCE namespace. The **fts crserverentry** command must be used before this command to create a server entry in the FLDB for the machine on which the fileset resides.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for the machine specified by **-server**.

Examples

The following example creates an entry in the FLDB for a non-LFS fileset named *user.jlw*. The fileset is located on the File Server machine named **fs3**. The aggregate ID of the partition the fileset resides on is **7**.

```
$ fts crfldbentry user.jlw /.../abc.com/hosts/fs3 7
```

Related Information

Commands: **dfsexport(8dfs)**, **fts create(8dfs)**, **fts crserverentry(8dfs)**, **fts crmount(8dfs)**.

Files: **dfstab(4dfs)**.

fts crmount

Purpose

fts crmount – Creates a mount point for a fileset

Synopsis

fts crmount -dir *directory_name* {**-fileset** {*name* | *ID*} | **-global** } [**-rw**][**-fast**] [**-help**]

Options

-dir *directory_name*
Names the location in the file tree at which the root directory of the fileset is to be mounted. The specified location becomes the fileset's mount point. The specified mount point must not already exist, but the parent directory of the mount point must exist in the DFS filespace.

-fileset {*name* | *ID*}
Specifies the complete name or fileset ID number of the fileset to be mounted. The mount point for the fileset is created at the location specified with the **-dir** option. The read/write, read-only, or backup version of the fileset can be named. Use this option or use the **-global** option.

-global Specifies that the mount point named with the **-dir** option is for the root of the DCE global namespace (a global root mount point). Use this option or use the **-fileset** option. Do not use the **-global** option under normal circumstances; it is maintained to provide backward compatibility with other file systems (for example, AFS).

-rw Specifies the type of the mount point as read/write. By default, a mount point is regular. If this option is used, the **-fileset** option must specify the read/write version of the fileset.

An important function of the **-rw** option is to mount a cell's main read/write fileset, **root.dfs**, below the top level of the cell's DFS filespace. The option must be used in this capacity at installation if replication is to be available in the cell. See the description section for more information

about the different types of mount points and about using the **-rw** option to create a read/write mount point for the fileset **root.dfs**.

-fast Directs **fts** not to verify the existence of the fileset indicated with the **-fileset** option. Use this option to create a mount point for a fileset that does not yet exist.

By default, **fts** contacts the Fileset Location (FL) Server for the cell that houses the parent directory of the name supplied with the **-dir** option to verify that the FLDB contains an entry for the fileset to be mounted. If no FLDB entry exists for the specified fileset, the command displays a warning. The specified mount point is always created, regardless of whether the **-fast** option is provided; the option simply suppresses the check and any possible warnings.

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

Description

The **fts crmount** command creates a mount point for the fileset specified with the **-fileset** option at the location in the file tree specified with the **-dir** option. The mount point makes the contents of the specified fileset visible and accessible to users. Once this command is used to mount a fileset, no further actions need to be taken to mount the fileset and the fileset never needs to be mounted again.

A mount point is actually a special symbolic link that acts as an association between a directory location and a fileset. Mount points look and function like standard directories. When the Cache Manager encounters a mount point in a pathname traversal, it determines which fileset is associated with the mount point. When it finds that fileset, it can access files or directories contained in the fileset's root directory. It traverses any paths leading through directories or other mount points in the fileset until it finds the directory or file indicated by the pathname.

To a large extent, the type of a mount point determines the version of a fileset through which the Cache Manager searches for a requested directory or file. By default, every mount point created by the **fts crmount** command is a regular mount point, which is the most common type of mount point. However, if you include the **-rw** option with the command, the command creates a read/write mount point.

When the Cache Manager encounters a regular mount point, it checks the version of the fileset that the mount point indicates. If the mount point indicates the read-only or

backup version of the fileset, the Cache Manager accesses that version. If the mount point indicates the read/write version of the fileset, the Cache Manager considers the fileset in which the mount point resides and acts accordingly, as follows:

- If a regular mount point that names a read/write fileset resides in a read-only fileset, the Cache Manager first determines whether the fileset is replicated. If the fileset is not replicated, the Cache Manager attempts to access the read/write version of the fileset; if the read/write version does not exist or is inaccessible, the Cache Manager cannot access the fileset. If the fileset is replicated, the Cache Manager attempts to access a read-only version of the fileset; if all of the read-only copies are unavailable, the Cache Manager cannot access the fileset (it does not attempt to access the read/write version of the fileset).
- If a regular mount point that names a read/write fileset resides in a read/write fileset, the Cache Manager attempts to access only the read/write version of the fileset. If the read/write version does not exist or is inaccessible, the Cache Manager cannot access the fileset.

When the Cache Manager encounters a read/write mount point, it attempts to access only the read/write version of the fileset, regardless of the type of fileset in which the mount point resides. If the read/write version of the fileset does not exist or is inaccessible, the Cache Manager cannot access the fileset.

Regular mount points promote greater fileset availability than read/write mount points because they allow the Cache Manager to access read-only filesets as often as possible. Because multiple copies of read-only filesets typically exist, regular mount points generally increase fileset availability. Conversely, because only a single version of a read/write fileset can exist, read/write mount points generally reduce fileset availability.

You typically mount filesets with regular mount points. A regular mount point is explicitly not a "read-only" mount point. Although the Cache Manager attempts to access a read-only version of a fileset when it encounters a regular mount point, it accesses the read/write version of the fileset if no read-only versions of the fileset exist or if it is traversing a read/write path (that is, if it accessed the mount point in a read/write fileset).

During a cell's configuration, an important function of the **fts crmount** command is to create a mount point for the cell's main read/write fileset, **root.dfs**. The command must be used with the **-rw** option to create an explicit read/write mount point for the fileset below the top level of the cell's DFS filespace. The mount point for the fileset must be created at **/...**/*cellname* **/fs/.rw**.

The **root.dfs** fileset is the first fileset mounted in a cell. The Cache Manager automatically mounts it at the top level of the cell's DFS filespace (**/...**/*cellname* / **fs** by default, but it can be defined as something else). It must be created as a DCE LFS fileset with the **fts create** command if functionality such as replication is to be available in the cell.

Once the **root.dfs** fileset is mounted with a read/write mount point, it can be replicated. Replication is then available for DCE LFS filesets in the cell. If **root.dfs** is replicated before its read/write mount point is created with the **fts crmount** command, the read/write version of **root.dfs** cannot be accessed in the cell. See Part 1 of this manual for information about configuring **root.dfs** to support replication.

Privilege Required

If the parent directory of the mount point (the directory in which the mount point is to be created) is in a DCE LFS fileset, the issuer must have write, execute, control, and insert permissions on the directory. If the parent directory is in a non-LFS fileset, the issuer must have write and execute permissions on the directory.

Cautions

Do not mount a fileset at more than one location in the file system. Creating multiple mount points can distort the hierarchical nature of the file system. Because the Cache Manager stores only a single pointer to the parent directory of the mount point for each fileset, it can become confused about which pathname to follow when searching for a file in a fileset with multiple mount points. This is true even if the full pathname of a file is specified.

Create multiple mount points for a fileset sparingly, only in a very limited number of troubleshooting and testing situations. Remove the extraneous mount points as soon as they are no longer necessary.

Do not create a symbolic link that begins with a **#** (number sign) or a **%** (percent sign) character. Because a mount point is a special type of symbolic link that uses these characters internally to identify its type, the Cache Manager becomes confused if it encounters a normal symbolic link that begins with one of these characters.

Examples

The following command creates a regular mount point (the default type of mount point) for the read/write fileset named *user.jlw* at the directory named */.../abc.com/fs/usr/jlw*:

```
$ fts crmount /.../abc.com/fs/usr/jlw user.jlw
```

The following command creates a read/write mount point for the fileset named **root.dfs** in the cell *abc.com*. The fileset is mounted at **.rw**, immediately below the top level of the cell's DFS filespace.

```
$ fts crmount /.../abc.com/fs/.rw root.dfs -rw
```

Related Information

Commands: **dfsd(8dfs)**, **fts create(8dfs)**, **fts delmount(8dfs)**, **fts lsmount(8dfs)**.

fts crserverentry

Purpose

fts crserverentry – Creates a server entry in the FLDB

Synopsis

fts crserverentry -server *machine* **-principal** *name* [**-quota** *entries*] [**-owner** *group*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server*machine*

Specifies the server machine for which an entry is to be created in the Fileset Location Database (FLDB). Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-principal *name*

Specifies the abbreviation for the DFS server principal to be registered in the FLDB for the machine (for example, **hosts**/*hostname*). The machine's principal name in the Registry Database must match this name.

-quota *entries*

Sets a limit on the number of fileset entries (read-write, read-only, and backup) in the FLDB that can be associated with the specified **-server**. If this option is omitted, a value of **0** (zero) is used, meaning an unlimited number of fileset entries can be associated with the specified File Server machine.

-owner *group*

Specifies the name of the group that is the owner of the server entry. A group can be specified by a full or abbreviated group name (for example, **/.../** *cellname/ group_name* or just *group_name*). Foreign groups cannot own a local server entry. If this option is omitted, no group owns the server entry. (The value **<nil>** is reported as the owner.)

-cell *cellname*
Specifies the cell in whose FLDB the server entry is to be created. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts crserverentry** command creates a server entry in the FLDB for the server machine specified with **-server**. You must issue this command for a server machine before issuing any other **fts** commands involving that machine (for example, before creating filesets on the machine with the **fts create** command, before adding the machine as a replication site with the **fts addsite** command, before moving filesets to the machine with the **fts move** command, and so on).

The **-quota** option is used to limit the number of filesets (read/write, read-only, and backup) that can reside on the specified File Server machine. When a fileset entry in the FLDB is defined to reference the File Server machine as the site of a fileset version, the FL Server increments the number of fileset entries recorded as residing on the server in its server entry. The FL Server creates no more than the specified **-quota** of fileset entries on the server machine.

The following commands update the number of fileset entries recorded for a File Server machine in its server entry. The **fts create**, **fts crfldbentry**, **fts addsite**, **fts restore**, **fts clone**, and **fts clonesys** commands increment the number of fileset entries recorded for the server; the **fts delete**, **fts delfldbentry**, and **fts rmsite** commands decrement the number of fileset entries recorded; the **fts move** command decrements

and increments the number of fileset entries recorded on the source and destination machines, respectively; and the **fts syncfldb** and **fts syncserv** commands can update the number of fileset entries recorded, as necessary.

The **-owner** option is used to specify a group of administrators who can administer entries in the FLDB for all of the filesets on the specified File Server machine. The same group can be given ownership of all server entries for the File Server machines in the domain where the specified machine resides. Members of the group can then manipulate the FLDB entries for all of the filesets in the domain where the File Server machine resides. This way, the administrators in the group need not be included on the **admin.fl** list for the entire cell, which would allow them to modify all of the fileset entries in the FLDB in that cell.

Use the **fts lsserverentry** command to display the current values from the FLDB for a server entry. Use the **fts edserverentry** command to change the current values in the FLDB for a server entry. Use the **fts delserverentry** command to remove a server entry from the FLDB.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines.

Examples

The following example adds a server entry to the FLDB for a server machine named **fs1**. The abbreviated DFS server principal of the machine is specified with the **-principal** option as **hosts/fs1**. Because they are omitted, the **-quota** and **-owner** options receive the default values of **0** (zero) and the empty group ID (displayed as <**nil**>), respectively.

```
$ fts crserverentry /.../abc.com/hosts/fs1 hosts/fs1
```

Related Information

Commands: **fts delserverentry(8dfs)**, **fts edserverentry(8dfs)**, **fts lsserverentry(8dfs)**.

fts delete

Purpose

fts delete – Removes a specified read/write or backup version of a DCE LFS fileset

Synopsis

fts delete -fileset {*name* | *ID*} **-server** *machine* **-aggregate** *name* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}
Specifies the complete name or fileset ID number of the read/write or backup fileset to be removed. Include the **.backup** extension if specifying the name of a backup fileset.

-server*machine*
Names the File Server machine from which to remove the fileset. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate from which to remove the fileset. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate in the *dcelocal* **/var/dfs/dfstab** file.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database).

You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts delete** command removes the read/write or backup DCE LFS fileset indicated by the **-fileset** option from the site specified by the **-server** and **-aggregate** options. Versions of the fileset are removed and the Fileset Location Database (FLDB) entry for the fileset updated to record the removals as follows:

- Removing a read/write fileset automatically removes its associated backup version (if the backup version exists). If read-only versions of the fileset exist, site information for the read/write and backup versions of the fileset is removed from the fileset's FLDB entry and the status flags for both versions are set to **invalid** (their fileset ID numbers are still recorded), but the read-only versions of the fileset are not affected. If no read-only versions of the fileset exist, the entire entry for the fileset is removed from the FLDB.
- Removing a backup fileset removes site information for the backup version from the fileset's FLDB entry and marks the backup version as **invalid** but does not erase its fileset ID number. Read/write and read-only versions of the fileset are not affected.

The number of fileset entries recorded in the server entry in the FLDB for the File Server machine from which a read/write or backup version of a fileset is removed is decremented once for each deleted version of the fileset. (Note that, if the indicated version of a fileset does not exist at the specified site but is referenced in the fileset's FLDB entry, the command removes site information about that version of the fileset from the fileset's entry and generally performs all other operations as indicated.)

Before you remove the read/write (and backup) version of a fileset, use the **fts rmsite** command to remove the fileset's replication sites and to instruct the Replication Server to remove the replicas stored at the sites. If Release Replication was used for the fileset, use the **fts rmsite** command to remove the replication site and replica stored at the read/write fileset's site as well.

After removing a fileset, use the **fts delmount** command to remove its mount point. Note that it might be better in some cases to remove a fileset's mount point before deleting the fileset; removing the mount point first ensures that no users are accessing the fileset when it is deleted.

If the DCE LFS fileset to be removed is also mounted locally (as a file system on its File Server machine), you must remove its local mount point before you delete it; the **fts delete** command cannot be used to delete a fileset that is mounted locally. In addition, because the backup version of a fileset is removed when its read/write version is removed, you cannot remove the read/write version of a fileset if its backup version is mounted locally. You must remove the backup version's local mount point before deleting the read/write version.

The **fts delfldbentry** command can be used to remove an FLDB entry for a fileset. Use the command only when you are certain that a fileset deletion was not recorded in the FLDB. The **fts zap** command can be used to remove a fileset when it is urgent that the fileset be removed but the FLDB is inaccessible. When the FLDB is again accessible, use the **fts delfldbentry** command to remove the fileset's FLDB entry or use the **fts syncfldb** and **fts syncserv** commands to synchronize the FLDB with the state of the filesets.

The **fts delfldbentry** command is also used to remove the FLDB entry for a non-LFS fileset. The **fts delmount** command is then used to remove its mount point, and the **dfsexport** command is used to detach the partition it resides on from the global namespace.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine specified by **-server**. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine on which a version of the fileset to be deleted resides.

Examples

The following command deletes the read/write fileset named *user.terry* and its backup version (if it exists) from the aggregate named **/dev/lv01** on the File Server machine named **fs3**:

```
$ fts delete user.terry /.../abc.com/hosts/fs3 /dev/lv01
```

Related Information

Commands: **dfsexport(8dfs)**, **fts delfldbentry(8dfs)**, **fts delmount(8dfs)**, **fts rmsite(8dfs)**, **fts syncfldb(8dfs)**, **fts syncserv(8dfs)**, **fts zap(8dfs)**.

Files: **dfstab(4dfs)**.

fts delfldbentry

Purpose

fts delfldbentry – Removes a specified entry from the FLDB

Synopsis

fts delfldbentry {**-fileset** {*name* | *ID*} | **-prefix** *string*} [**-server** *machine*] [**-aggregate** *name*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}

Specifies the complete name or fileset ID number of a fileset. The entire entry for the fileset is removed from the Fileset Location Database (FLDB), regardless of whether the read/write, read-only, or backup version of the fileset is specified. Provide this option or use the **-prefix** option.

-prefix *string*

Specifies a character string of any length. Every FLDB entry that lists a fileset whose name begins with this exact string is removed (unless more restrictive constraints are specified with the **-server** and optionally **-aggregate** options). Include field separators such as dots if appropriate. Provide this option (optionally with **-server** and **-aggregate**) or use the **-fileset** option.

-server *machine*

Names a File Server machine. If a fileset's name matches the specified **-prefix** and it is listed as residing on the specified File Server machine, its entry is removed from the FLDB. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. If this option is used, **-prefix** must be used; **-aggregate** can also be used.

-aggregate *name*

Specifies the device name, aggregate name, or aggregate ID of an aggregate or partition on **-server**. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or

partition in the *dcelocal*/**var/dfs/dfstab** file. If a fileset's name matches the specified **-prefix** and it resides on the specified aggregate on **-server**, its entry is removed from the FLDB. If this option is used, **-prefix** and **-server** must be used.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts delfldbentry** command removes the entries for all indicated filesets from the FLDB. Regardless of the version of a fileset (read/write, read-only, or backup) specified with the command, the fileset's entire entry is removed. The command has no effect on actual filesets on File Server machines, only on their FLDB entries.

The command also decrements the number of fileset entries recorded in server entries, as appropriate. For each version of a fileset whose entry is removed from the FLDB, the number of fileset entries recorded in the server entry for the File Server machine it resides on is reduced by one.

By using the **-fileset** option alone or combining the **-prefix**, **-server**, and **-aggregate** options in increasingly specific ways, FLDB entries can be removed for varying numbers of filesets. To remove the FLDB entry for

- A single fileset, specify **-fileset**.

- Every fileset whose name begins with a certain character string (for example, **sys.** or **user.**), regardless of site, specify **-prefix**.
- Every fileset whose name begins with a certain character string and that is stored on a specific File Server machine, specify **-prefix** and **-server**.
- Every fileset whose name begins with a certain character string and that is stored on a specific aggregate of a specific File Server machine, specify **-prefix**, **-server**, and **-aggregate**.

This command can be used if the issuer is certain that a fileset removal is not recorded in the FLDB and does not want to take the time to synchronize an entire File Server machine. It can also be used to remove the FLDB entry for a non-LFS fileset to be removed from the global namespace. (Use the **fts rmsite** command to remove an incorrect entry for a read-only site from the FLDB.)

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine that houses a version of any fileset whose FLDB entry is to be removed.

Cautions

Do not use this command as the standard way to remove a fileset entry from the FLDB. It can make the FLDB inconsistent with the filesets on server machines. Use the **fts delete** command to remove the fileset entry from the FLDB at the same time that the fileset is deleted. Also, because it can be used to remove multiple FLDB entries simultaneously, use this command carefully.

Examples

The following command removes the FLDB entry for the fileset **user.temp**:

```
$ fts delfldbentry user.temp
```

The following command removes all FLDB entries for filesets whose names begin with **test** and that are stored on the File Server machine named **fs3**:

```
$ fts delfldbentry -prefix test -server /.../abc.com/hosts/fs3
```

Related Information

Commands: **fts clone(8dfs)**, **fts delete(8dfs)**, **fts rmsite(8dfs)**, **fts syncfldb(8dfs)**, **fts syncserv(8dfs)**, **fts zap(8dfs)**.

Files: **dfstab(4dfs)**.

fts delmount

Purpose

fts delmount – Removes a mount point

Synopsis

fts delmount -dir *directory_name…* [**-help**]

Options

-dir *directory_name*
: Names the mount point to be deleted. Provide a complete pathname. The last element in the pathname must be an actual name, not . (dot) or .. (dot dot).

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **fts delmount** command removes the mount point specified by **-dir**. The associated fileset is not affected, but it is inaccessible if no other mount points exist for it. When the mount point for a fileset is removed, any fileset mounted only as a subdirectory of the fileset's root directory becomes inaccessible.

Privilege Required

If **-dir** resides in a directory in a DCE LFS fileset, the issuer must have write, execute, and delete permissions on the directory in which it resides. If **-dir** resides in a directory in a non-LFS fileset, the issuer must have write and execute permissions on the directory in which it resides.

Examples

The following command removes the mount point for the fileset *user.vijay*, which is mounted at */.../abc.com/fs/usr/vijay*:

```
$ fts delm /.../abc.com/fs/usr/vijay
```

Related Information

Commands: **fts crmount(8dfs)**, **fts lsmount(8dfs)**.

fts delserverentry

Purpose

fts delserverentry – Deletes a server entry from the FLDB

Synopsis

fts delserverentry -server *machine* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server *machine*
: Specifies the server machine whose entry is to be removed from the Fileset Location Database (FLDB). Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-cell*cellname*
: Specifies the cell from whose FLDB the server entry is to be removed. The default is the local cell of the issuer of the command.

-noauth
: Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose
: Directs **fts** to provide detailed information about its actions as it executes the command.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **fts delserverentry** command removes the existing server entry from the FLDB for the server machine specified with **-server**. When the command is issued, no fileset entries in the FLDB can reference the server entry to be removed as the site of a fileset. If a fileset entry in the FLDB references the server entry to be removed, the command fails.

Use the **fts crserverentry** command to create a server entry in the FLDB. Use the **fts lsserverentry** command to display the current values from the FLDB for a server entry. Use the **fts edserverentry** command to change the current values in the FLDB for a server entry.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines.

Examples

The following example deletes the server entry from the FLDB for the server machine named **fs1**. No filesets can reside on the machine when the command is issued.

```
$ fts delserverentry /.../abc.com/hosts/fs1
```

Related Information

Commands: **fts crserverentry(8dfs)**, **fts edserverentry(8dfs)**, **fts lsserverentry(8dfs)**.

fts dump

Purpose

fts dump – Converts a fileset to a bytestream format and places it in a file

Synopsis

fts dump -fileset {*name* | *ID*} {**-time** {*date* | *0*} | **-version** *number*} [**-server** *machine*] [**-file** *filename*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}

Specifies the complete name or fileset ID number of the fileset to be dumped. The read/write, read-only, or backup version of the fileset can be dumped. Append the **.readonly** or **.backup** extension to the name of the fileset to dump the read-only or backup version instead of the read/write version.

-time {*date* | *0*}

Specifies a full or incremental dump. Three values are legal:

0 (zero) A **0** (zero) value causes a full dump of the current version of the fileset.

mm/*dd*/*yy* A month/day/year value causes an incremental dump from 12:00 a.m. (00:00) on the indicated date (for example, **1/23/90** or **10/16/92**). Only files with modification time stamps equal to or later than the specified date and time are dumped.

mm/*dd*/*yy hh*:*mm*

An exact date and time value causes an incremental dump from the specified time on the indicated date. Only files with modification time stamps equal to or later than the specified date and time are dumped. The time must be in 24-hour format (for example, **20:30** for 8:30 p.m.). The date format is the same as for a date alone. Surround the entire argument with " " (double quotes) because it

contains a space (for example, **"1/23/90 22:30"** or **"10/16/92 3:45"**).

Use this option to perform a full dump or to perform an incremental dump of only those files in the fileset modified since a specific date or date and time. Use this option or use **-version**.

-version *number*

Specifies an incremental dump based on the indicated fileset version number. Each DCE LFS fileset has a version number. Each file in the fileset records the version number that was current when the file was last modified. If this option is specified, only those files with version numbers equal to or greater than the specified version number are dumped. (A DCE LFS fileset's version number is recorded in its fileset header; it has the same format as a fileset ID number. Use the **fts lsheader** or **fts lsft** command to display a fileset's current version number.)

Use this option or use **-time**. Use this option only with DCE LFS filesets.

-server*machine*

Names the File Server machine that houses the version of the fileset to be dumped. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

This option is useful for dumping a particular read-only replica of a DCE LFS fileset for which multiple replicas exist. If you include the **.readonly** extension with the name of a fileset specified with the **-fileset** option, or if you specify the ID number of the read-only version of a fileset with the **-fileset** option, you can use the **-server** option to indicate the machine that houses the specific replica to be dumped. If you omit the **-server** option in these cases, the command dumps the replica that resides at the fileset's oldest read-only site (the replica at the site that has been defined for the longest time).

This option is always unnecessary if the read/write or backup version of a fileset is to be dumped.

-file *filename*

Specifies the complete pathname of the file to which the dump is to be written. If a complete pathname is not specified, the file is written to the current working directory. If this option is omitted, the data is sent to standard output (**stdout**).

-cell*cellname*
: Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth
: Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose
: Directs **fts** to provide detailed information about its actions as it executes the command.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **fts dump** command converts the contents of the indicated fileset to a bytestream format. It puts the converted contents into the file specified with the **-file** option. If this option is omitted, the dumped data is sent to **stdout**. Both non-LFS and read/write, read-only, and backup DCE LFS filesets can be dumped.

The command's options can be used to perform the following types of dumps:

- A value of 0 (zero) specified with the **-time** option causes a full dump of the fileset.
- A date specified with the **-time** option causes an incremental dump of all files modified since 12:00 a.m. (00:00) on that date.
- A date and time specified with the **-time** option cause an incremental dump of all files modified since that date and time.
- A version number specified with **-version** causes an incremental dump of all files in the fileset with version numbers equal to or greater than the specified version number.

Dumping a fileset does not affect its status in the Fileset Location Database (FLDB) or at the site from which it is dumped. However, it does make the fileset inaccessible for the duration of the dump operation. For this reason, it is customary to dump the backup version of a fileset to prevent the read/write version from being inaccessible for an extended time.

If a read-only replica of a DCE LFS fileset is to be dumped and multiple replicas of the fileset exist, the **-server** option can be used to name the File Server machine that houses the specific replica to be dumped. Indicating a specific replica can be useful if network or hardware problems caused only some of a fileset's replicas to be updated. It can be especially useful for restoring the read/write version of a fileset that was lost before all of its replicas were updated, since you can dump and restore a specific replica that was updated before the read/write version was lost. (By default, all replicas of the same fileset are always identical; to determine whether all replicas of a fileset are the same, use the **fts lsft** command to display information about specific replicas.)

The **fts restore** command can be used to restore a fileset dumped with the **fts dump** command. You can use the **fts restore** command to restore a dump file to any type of fileset (DCE LFS or non-LFS), regardless of the type of fileset from which it was created. Thus, you can dump and restore data between DCE LFS and non-LFS filesets, as well as between different types of non-LFS filesets. (See the documentation for the **fts restore** command for more information about dumping and restoring filesets between different types of file systems.)

You cannot restore a fileset dumped in one cell to a site in another cell.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine on which the fileset is stored. In addition, the issuer must have the write, execute, and insert permissions on the directory in which the dump file is to reside.

Examples

The following command executes a full dump of the fileset *user.terry* into the file named */tmp/terry.dump*:

```
$ fts dump user.terry -time 0 /tmp/terry.dump
```

The following command executes an incremental dump of the fileset *user.smith* into the file named */tmp/smith.013191.dump*. Only those files in the fileset with modification time stamps equal to or later than 6:00 p.m. on 31 January 1991 are included in the dump.

```
$ fts dump user.smith -time "1/31/91 18:00" /tmp/smith.013191.dump
```

Related Information

Commands: **fts lsft(8dfs)**, **fts restore(8dfs)**.

fts edserverentry

Purpose

fts edserverentry – Edits a server entry in the FLDB

Synopsis

fts edserverentry -server *machine* [{**-rmaddr** | **-addaddr** *address* | **-changeaddr** *address*}] [**-principal** *name*] [**-quota** *entries*] [{**-owner** *group* | **-noowner** }] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server *machine*

Specifies the server machine whose entry in the Fileset Location Database (FLDB) is to be edited. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. If the **-rmaddr**, **-addaddr**, or **-changeaddr** option is used with the command, specify the network address.

-rmaddr Removes the address specified with **-server** from the server entry identified by **-server** in the FLDB. If the name of the machine rather than one of its addresses is specified with **-server**, the command can choose one of the machine's addresses at random to be removed from the FLDB. Because this can have unpredictable results, always specify an address with **-server** when using the **-rmaddr** option. In addition, the command fails if the address to be removed is the only address present for the machine in the FLDB.

If this option is specified, do not specify the **-addaddr** or **-changeaddr** option.

-addaddr *address*

Adds the additional address specified with this option to the server entry specified by **-server** in the FLDB. A machine can have from one to four addresses associated with it in the FLDB. The command fails if you attempt to add a fifth address for the machine to the FLDB.

If the name of the machine rather than one of its addresses is specified with **-server**, the command can choose one of the machine's addresses

in the FLDB at random to have the address added to it. Because this can have unpredictable results, always specify an address with **-server** when using the **-addaddr** option.

If this option is specified, do not specify the **-rmaddr** or **-changeaddr** option.

-changeaddr *address*
Substitutes the address specified with this option for the address specified by **-server** in the FLDB. If the name of the machine rather than one of its addresses is specified with **-server**, the command can choose one of the machine's addresses at random to be replaced with the address specified with this option. Because this can produce unpredictable results, always specify an address with **-server** when using the **-changeaddr** option.

If this option is specified, do not specify the **-rmaddr** or **-addaddr** option.

-principal*name*
Changes the abbreviation for the DFS server principal that is registered for the machine in the FLDB (for example, **hosts/***hostname*). The machine's principal name in the Registry Database must match this name. If this option is omitted, the abbreviated DFS server principal currently associated with the server entry remains unchanged.

-quota*entries*
Changes the limit on the number of fileset entries (read/write, read-only, and backup) in the FLDB that can be associated with the specified **-server**. A value of 0 (zero) allows an unlimited number of fileset entries to be associated with the server. If this option is omitted, the number of fileset entries currently allowed for the specified File Server machine remains unchanged.

-owner *group*
Changes the group that is the owner of the server entry. In the entry, the specified group replaces the current owning group, if any. A group can be specified by a full or abbreviated group name (for example, **/.../***cellname*/*group_name* or just *group_name*). Foreign groups cannot own a local server entry. If this option is omitted, no group owns the server entry. (The value **<nil>** is reported as the owner.) Use this option or use the **-noowner** option; omit both options to leave the current owning group unchanged.

-noowner Specifies that no group is to own the server entry. In the entry, the empty group ID, displayed as **<nil>**, replaces the group that currently owns the server entry; the entry is unchanged in this regard if no group presently owns the server entry. Use this option or use the **-owner** option; omit both options to leave the current owning group unchanged.

-cell *cellname*
Specifies the cell in whose FLDB the server entry is to be modified. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts edserverentry** command alters a server entry in the FLDB for the server machine specified with the **-server** option. Use the **-rmaddr** option to remove an address associated with a server from the FLDB. Use the **-addaddr** option to add a new address for a server to the FLDB, or use the **-changeaddr** option to change an address for a server in the FLDB.

The **-principal** option can be used to change the abbreviated DFS server principal associated with the server entry. The **-quota** option can be used to alter the number of fileset entries that can be associated with the File Server machine in the FLDB, and the **-owner** option can be used to assign a new group as the owner of the server entry (or the **-noowner** option can be used to indicate that no group owns the server entry).

Unless a value associated with a server entry is explicitly modified with this command, its current value in the FLDB remains unchanged. The values associated with a server entry are initially specified when the server entry is created with the **fts crserverentry** command. The values can then be modified at any time with the **fts edserverentry** command. Use the **fts lsserverentry** command to display the current values from the FLDB for a server entry. Use the **fts delserverentry** command to remove a server entry from the FLDB.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines.

Examples

The following command modifies the server entry in the FLDB for a server machine. The command changes the machine's network address from **191.54.206.36**, as specified with the **-server** option, to **191.54.206.46**, as indicated with the **-changeaddr** option. The command also allows the server to accommodate an unlimited number of fileset entries by providing a value of **0** (zero) with the **-quota** option.

```
$ fts edserverentry 191.54.206.36 -changeaddr 191.54.206.46 -quota 0
```

Related Information

Commands: **fts crserverentry(8dfs)**, **fts delserverentry(8dfs)**, **fts lsserverentry(8dfs)**.

fts help

Purpose

fts help – Shows syntax of specified **fts** commands or lists functional descriptions of all **fts** commands

Synopsis

fts help [**-topic** *string*]… [**-help**]

Options

-topic *string* Specifies each command whose syntax is to be displayed. Provide only the second part of the command name (for example, **lsft**, not **fts lsft**). If this option is omitted, the output provides a short description of all **fts** commands.

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

Description

The **fts help** command displays the first line (name and short description) of the online help entry for every **fts** command if **-topic** is not provided. For each command name specified with **-topic**, the output lists the entire help entry.

Use the **fts apropos** command to show each help entry containing a specified string.

Privilege Required

No privileges are required.

Output

The online help entry for each **fts** command consists of the following two lines:

- The first line names the command and briefly describes its function.

- The second line, which begins with **Usage:**, lists the command options in the prescribed order.

Examples

The following command displays the online help entry for the **fts delmount** command:

$ fts help delmount

```
fts delmount: remove mount point
Usage: fts delmount -dir <directory_name>...  [-help]
```

Related Information

Commands: **fts apropos(8dfs)**.

fts lock

Purpose

fts lock – Locks a fileset entry in the FLDB

Synopsis

fts lock -fileset {*name* | *ID*} [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}
: Specifies the complete name or fileset ID number of the fileset whose entry in the Fileset Location Database (FLDB) is to be locked. All versions of the fileset referenced in the entry are affected by the lock, regardless of whether the read/write, read-only, or backup version of the fileset is specified.

-cell *cellname*
: Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth
: Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose
: Directs **fts** to provide detailed information about its actions as it executes the command.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **fts lock** command locks the entry in the FLDB for the fileset indicated with the **-fileset** option. Locking a fileset's FLDB entry blocks operations on all versions of the fileset, regardless of whether the read/write, read-only, or backup version of the fileset is indicated with the **-fileset** option. Locking a fileset's entry prevents all versions of the fileset from being modified with **fts** commands.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine on which a version of the fileset to be locked resides.

Cautions

Do not use this command in normal circumstances. It is useful only if the system administrator wants to guarantee that no one else manipulates the fileset until the lock is released and if there is reason to believe that locking will not happen automatically. Locking a fileset entry inhibits only operations such as deleting and cloning of the fileset; it does not prevent the reading of data from the fileset.

Examples

The following command locks the FLDB entry for *user.terry*:

```
$ fts lock user.terry
```

Related Information

Commands: **fts unlock(8dfs)**, **fts unlockfldb(8dfs)**.

fts lsaggr

Purpose

fts lsaggr – Lists all exported aggregates and partitions on a File Server machine

Synopsis

fts lsaggr - server *machine* [**- cell** *cellname*] [{**-noauth** | **-localauth** }] [**- verbose**] [**-help**]

Options

-server *machine*
: Names the File Server machine whose exported aggregates and partitions are to be listed. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-cell *cellname*
: Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth
: Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **- localauth** option.

-localauth
: Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose
: Directs **fts** to provide detailed information about its actions as it executes the command.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **fts lsaggr** command displays information about all exported aggregates and partitions on the File Server machine specified by the **-server** option. The information about each aggregate and partition is specified in the *dcelocal* **/var/dfs/dfstab** file on the machine.

You can also issue the **dfsexport** command with no options to list all aggregates and partitions currently exported from the local disk to the DCE namespace. You can use the **fts aggrinfo** command to display information about the amount of disk space available on a specific aggregate or partition or on all aggregates and partitions on a File Server machine.

Privilege Required

No privileges are required.

Output

This command displays a separate line for each aggregate or partition. Each line displays the following information:

- The aggregate name, specified in the second field of the **dfstab** file
- The device name, specified in the first field of the **dfstab** file
- The aggregate ID, specified in the fourth field of the **dfstab** file
- The file system type, specified in the third field of the **dfstab** file

Examples

The following example shows that two non-LFS partitions and two DCE LFS aggregates are exported from the File Server machine named */.../abc.com/hosts/fs1*:

```
$ fts lsaggr /.../abc.com/hosts/fs1
```

```
There are 4 aggregates on the server /.../abc.com/hosts/fs1(fs1.abc.com):
             /usr (/dev/lv02): id=3     (non-LFS)
             /tmp (/dev/lv03): id=4     (non-LFS)
```

```
lfs1 (/dev/lfs1): id=10     (LFS)
lfs2 (/dev/lfs2): id=11     (LFS)
```

Related Information

Commands: **dfsexport(8dfs)**, **fts aggrinfo(8dfs)**.

Files: **dfstab(4dfs)**.

fts lsfldb

Purpose

fts lsfldb – Shows information from fileset entries in the FLDB

Synopsis

fts lsfldb [**-fileset** {*name* | *ID*}] [**-server** *machine*] [**-aggregate** *name*] [**-locked**][**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}

Specifies the complete name or fileset ID number of a fileset about which information from the Fileset Location Database (FLDB) is to be displayed. Use this option or use **-server** (and optionally **-aggregate**), **-locked**, or both. Omit this option and the **-server**, **-aggregate**, and **-locked** options to display information about all fileset entries in the FLDB.

-server *machine*

Names a File Server machine about whose filesets information from the FLDB is to be displayed. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. This option can be combined with **-aggregate** to display information about the filesets on a single aggregate on **-server**, or it can be combined with **-locked** to display information about the filesets with locked FLDB entries on the server machine. Use this option alone or with **-aggregate**, **-locked**, or both, or use **-fileset**. Omit this option and the **-fileset**, **-aggregate**, and **-locked** options to display information about all fileset entries in the FLDB.

-aggregate*name*

Specifies the device name, aggregate name, or aggregate ID of the aggregate or partition on **-server** about whose filesets information from the FLDB is to be displayed. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or partition in the *dcelocal*/**var/dfs/dfstab** file. The **-server** option must be provided with this option. The **-locked** option can be supplied with this option to

display information about the filesets with locked FLDB entries on the aggregate.

-locked Specifies that the output show information only for filesets with locked FLDB entries. Use this option alone to see information for all filesets with locked FLDB entries. Use this option with **-server** (and optionally **-aggregate**) to see all filesets on a specific server machine (and optionally aggregate) with locked FLDB entries. Use this option alone or with **-server** (and optionally **-aggregate**) or use **-fileset**. Omit this option and the **-fileset**, **-server**, and **-aggregate** options to display information about all fileset entries in the FLDB.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts lsfldb** command formats and displays information about fileset entries from the FLDB. Its options can be combined to display information about a variety of different filesets. To display FLDB information for

- Every fileset entry, specify no options.
- Every fileset entry that mentions a specific File Server machine as the site of any version of a fileset, specify the name of the machine with **-server**.

- Every fileset entry that mentions a specific aggregate on a specific File Server machine as the site of any version of a fileset, specify both **-server** and **-aggregate**.
- The FLDB entries for filesets with locked entries, specify the **-locked** option alone or with **-server** (and optionally **-aggregate**).
- The fileset entry for a single fileset, specify the fileset name or ID number with **-fileset**.

Use the **fts lsheader** command to display information from fileset headers. To display more information about a single fileset, use the **fts lsft** command to display all of the information displayed by the **fts lsheader** command when the **-long** option is used and all of the information displayed by this command.

Privilege Required

No privileges are required.

Output

The **fts lsfldb** command displays the following information from the FLDB for each DCE LFS fileset specified with **-fileset** or **-server** (and optionally **-aggregate**). Because functionality such as replication is not supported for non-LFS filesets, this command displays less information for non-LFS filesets.

- The fileset's name.
- The fileset IDs of the read/write, read-only, and backup versions of the fileset.
- For each version, a status flag of **valid** indicates the version actually exists at a site; a status flag of **invalid** indicates the version does not exist at any site. (For the read-only version, the status flag indicates whether a replication site is defined.)
- The number of sites at which a version of the fileset exists.
- The maximum and minimum advisory RPC authentication bounds for use in communications with Cache Managers. There are two sets of bounds: One set governs communications with Cache Managers in the local cell, while the other set governs communications with Cache Managers in foreign cells. Currently, these bounds are not enforced but serve to bias the Cache Manager's initial authentication level.
- An indicator if the FLDB entry is locked. (The indicator is omitted if the entry is not locked.)

- The replication parameters associated with the fileset.
- Information identifying the File Server machines and aggregates (sites) where read/write (RW), read-only (RO), or backup (BK) versions of the fileset reside.
- For a read-only version, the MaxSiteAge replication parameter defined for that site; for a read/write version, **0:00:00**.
- The abbreviated DCE principal name of each File Server machine on which a version of the fileset resides, and the name of the group that owns the server entry for the machine (or **<nil>** if no group owns the server entry).

If the output includes more than one FLDB entry, information about the filesets is displayed in alphabetical order by fileset name. The last line of the output displays the total number of entries successfully reported and the total number of entries not reported (the number of entries that **failed**).

Examples

The following command shows an example of the output from the **fts lsfldb** command for a fileset named *user.terry*:

```
$ fts lsfldb user.terry
```

```
user.terry
        readWriteID  0,,196953  valid
        readOnlyID   0,,196594  invalid
        backupID     0,,196595  valid
Minimum local protection level: rpc_c_protect_level_none
Maximum local protection level: rpc_c_protect_level_pkt_privacy
Minimum remote protection level: rpc_c_protect_level_none
Maximum remote protection level: rpc_c_protect_level_pkt_privacy
number of sites: 1
  Sched repl: maxAge=2:00:00; failAge=1d0:00:00;
  reclaimWait=18:00:00; minRepDelay=0:05:00; defaultSiteAge=0:30:00
   server        flags   aggr   siteAge principal  owner
fs3.abc.com      RW,BK   lfs1   0:00:00 hosts/fs3  <nil>
```

Related Information

Commands: **fts lock(8dfs)**, **fts lsfldb(8dfs)**, **fts lsft(8dfs)**, **fts unlock(8dfs)**, **fts unlockfldb(8dfs)**.

Files: **dfstab(4dfs)**.

fts lsft

Purpose

fts lsft – Lists fileset information from both the fileset header and the FLDB entry

Synopsis

fts lsft [{**-path** {*filename* | *directory_name*} | **-fileset** {*name* | *ID*}}] [**-server** *machine*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-path {*filename* | *directory_name*}

Names a file or directory on the fileset whose fileset header and FLDB information is to be displayed. Use this option or use **-fileset**; omit both options to display information about the fileset that contains the current working directory.

-fileset {*name* | *ID*}

Specifies the complete name or fileset ID number of the fileset to be examined. The read/write, read-only, or backup version of the fileset can be specified. Append the **.backup** or **.readonly** extension to the name of the fileset to list information about the backup or read-only version instead of the read/write version; if the read/write version no longer exists, the command fails if the **.backup** or **.readonly** extension is not used with the name of the fileset.

Use this option or use **-path**; omit both options to display information about the fileset that contains the current working directory.

-server *machine*

Names the File Server machine that houses the fileset about which information is to be displayed. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

This option is useful for displaying information about a particular read-only replica of a DCE LFS fileset for which multiple replicas exist. If you include the **.readonly** extension with the name of a fileset specified with the **-fileset** option, specify the ID number of the read-only version

of a fileset with the **-fileset** option, or specify the path to a file or directory in a read-only fileset with the **-path** option, you can use the **-server** option to indicate the machine that houses the specific replica about which information is to be displayed. If you omit the **-server** option in these cases, the command displays information about the replica at the fileset's oldest read-only site (the replica at the site that has been defined for the longest time).

This option is always unnecessary if information is to be displayed about the read/write or backup version of a fileset.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts lsft** command displays information from both the fileset header and the Fileset Location Database (FLDB) entry for the specified fileset. It displays the same output as the **fts lsheader** command with the **-long** option and the **fts lsfldb** command for a single fileset. It can be used to learn the fileset ID number of a fileset or to examine locked FLDB entries.

The fileset whose information is to be displayed can be specified by indicating the name of a file or directory on the fileset with the **-path** option, or it can be specified by indicating its name or ID number with the **-fileset** option. Omit both the **-path**

and **-fileset** options to display information about the fileset that contains the current working directory. If the name of the fileset is specified with the **-fileset** option, the **.backup** or **.readonly** extension can be appended to the name to display information about one of those fileset versions rather than the read/write version.

If information about a read-only replica of a DCE LFS fileset is to be displayed and multiple replicas of the fileset exist, the **-server** option can be used to name the File Server machine that houses the specific replica about which information is to be displayed. Indicating a specific replica can be useful if network or hardware problems caused only some of a fileset's replicas to be updated. (By default, all replicas of the same fileset should always contain the same information.)

Use the **fts lsheader** command to display information from fileset headers. Use the **fts lsfldb** command to display information from fileset entries in the FLDB.

Privilege Required

No privileges are required.

Output

The **fts lsft** command displays the following information from the fileset header and the FLDB entry for a specified DCE LFS fileset. Because non-LFS filesets do not have DCE LFS fileset headers, and because functionality such as replication is not supported for non-LFS filesets, this command displays less information for a non-LFS fileset.

The command displays the following information from the fileset's header:

- The fileset's name (with a **.readonly** or **.backup** extension, if appropriate)
- Its fileset ID number
- Its type (**RW** for read/write, **RO** for read-only, or **BK** for backup)
- Its type (**LFS** or **non-LFS**)
- Information about the state of the fileset
- Its status (**On-line**, **Off-line**, or an error indicator)
- The File Server machine, aggregate name, and aggregate ID number on which it resides
- The ID numbers of the parent, clone, and backup filesets related to the fileset

- The ID numbers of the low-level backing and low-level forward filesets related to the fileset
- Its version number
- Its allocation and allocation usage, in kilobytes
- Its quota and quota usage, in kilobytes
- The day, date, and time when the fileset was created (replicated or backed up for a read-only or backup fileset)
- The day, date, and time when the contents of the fileset were last updated (same as the creation time for a read-only or backup fileset)

It then displays the following information from the fileset's entry in the FLDB:

- The fileset's name.
- The fileset IDs of the read/write, read-only, and backup versions of the fileset.
- For each version, a status flag of **valid** indicates the version actually exists at a site; a status flag of **invalid** indicates the version does not exist at any site. (For the read-only version, the status flag indicates whether a replication site is defined.)
- The maximum and minimum advisory RPC authentication bounds for use in communications with Cache Managers. There are two sets of bounds: One set governs communications with Cache Managers in the local cell while, the other set governs communications with Cache Managers in foreign cells. Currently, these bounds are not enforced but serve to bias the Cache Manager's initial authentication level.
- The number of sites at which a version of the fileset exists.
- An indicator if the FLDB entry is locked. (The indicator is omitted if the entry is not locked.)
- The replication parameters associated with the fileset.
- Information identifying the File Server machines and aggregates (sites) on which read/write (**RW**), read-only (**RO**), or backup (**BK**) versions of the fileset reside.
- For a read-only version, the MaxSiteAge replication parameter defined for that site; for a read/write version, **0:00:00**.
- The abbreviated DCE principal name of each File Server machine on which a version of the fileset resides, and the name of the group that owns the server entry for the machine (or **<nil>** if no group owns the server entry).

Examples

The following example displays information from the fileset header and FLDB entry for a DCE LFS fileset named *user.terry*:

$ fts lsft -fileset user.terry

```
user.terry 0,,196953 RW LFS     states 0x10010005 On-line
    fs3.abc.com, aggregate lfs1 (ID 10)
    Parent 0,,196953   Clone 0,,0   Backup 0,,196955
    llBack 0,,0        llFwd 0,,0   Version 0,,25963
    429496729 K alloc limit;      1252 K alloc usage
        15000 K quota limit;      9340 K quota usage
    Creation Fri Oct 15 16:45:16 1993
    Last Update Mon Nov 22 11:36:00 1993


user.terry
        readWriteID  0,,196953  valid
        readOnlyID   0,,196594  invalid
        backupID     0,,196595  valid
Minimum local protection level: rpc_c_protect_level_none
Maximum local protection level: rpc_c_protect_level_pkt_privacy
Minimum remote protection level: rpc_c_protect_level_none
Maximum remote protection level: rpc_c_protect_level_pkt_privacy
number of sites: 2
  Sched repl: maxAge=2:00:00; failAge=1d0:00:00;
  reclaimWait=18:00:00; minRepDelay=0:05:00;
defaultSiteAge=0:30:00
   server        flags   aggr   siteAge principal  owner
fs3.abc.com      RW,BK   lfs1   0:00:00 hosts/fs3  <nil>
```

Related Information

Commands: **fts lsfldb(8dfs)**, **fts lsheader(8dfs)**.

fts lsheader

Purpose

fts lsheader – Shows information from fileset headers

Synopsis

fts lsheader - server *machine* [**- aggregate** *name*] [{**-fast** | **- long** }] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**- help**]

Options

-server *machine*
Names a File Server machine about whose filesets header information is to be displayed. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. This option can be combined with the **-aggregate** option to name a specific aggregate on **-server**.

-aggregate *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate or partition on **-server** from whose filesets header information is to be displayed. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or partition in the *dcelocal* **/var/dfs/dfstab** file. The **-server** option must be provided with this option.

-fast Directs the output to display only the fileset ID numbers of all filesets on the indicated server (and optionally the aggregate). If you use this option, do not use the **-long** option.

-long Directs the output to display more detailed information about all filesets on the indicated server (and optionally the aggregate). If you use this option, do not use the **- fast** option.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **- localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts lsheader** command formats and displays information from the fileset headers of filesets on the specified server (and optionally the aggregate or partition). To display information from the headers of all filesets on a specific File Server machine, specify the name of the server machine with the **-server** option. To specify information from the headers of all filesets on a specific aggregate on a File Server machine, specify the name of the server machine with the **-server** option and the name of the aggregate or partition with the **-aggregate** option.

Include the **-fast** option with the command to display only the ID numbers of the filesets at the specified location. Include the **- long** option with the command to display more detailed information from the headers of the filesets at the specified location.

Use the **fts lsfldb** command to display information from fileset entries in the Fileset Location Database (FLDB). To display more information about a single fileset, use the **fts lsft** command to display all of the information displayed by this command when the **- long** option is used and all of the information displayed by the **fts lsfldb** command.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine specified by **-server**.

Output

The **fts lsheader** command displays different output about the filesets at the specified location depending on whether the **- fast** or **-long** option is included. Information about the filesets is displayed in numeric order by fileset ID number if the **- fast** option is used; otherwise, it is displayed in alphabetical order by fileset name.

The information described in this section is displayed for DCE LFS filesets. Because non-LFS filesets do not have DCE LFS fileset headers, the **fts lsheader** command displays much less information for non-LFS filesets, and the **-fast** and **-long** options have less of an impact on the amount of output displayed.

If the **-fast** option is used, the command lists the ID number of each fileset. If the **-aggregate** option is omitted, the command also displays the total number of filesets on the specified server.

If both the **-fast** and **-long** options are omitted, the command displays the following information:

- The File Server machine, aggregate name, and aggregate ID number where the filesets reside.
- The total number of filesets on the aggregate.
- Each fileset's name (with a **.readonly** or **.backup** extension, if appropriate).
- Each fileset's fileset ID number.
- Each fileset's type (**RW** for read/write, **RO** for read-only, or **BK** for backup).
- Each fileset's allocation usage and quota usage, in kilobytes.
- Each fileset's status (**On-line**, **Off-line**, or an error indicator).
- The total number of filesets online, the total number of filesets offline, and the total number of filesets busy. A busy fileset is one upon which a fileset-related operation is currently in progress (for example, the fileset is being moved or cloned, or the Replication Server is currently forwarding changes from the fileset to read-only replicas).

If the **-long** option is used, the command displays the following additional information for each fileset:

- Whether it is a DCE LFS (**LFS**) or **non-LFS** fileset
- Information about the state of the fileset
- The ID numbers of the parent, clone, and backup filesets related to the fileset

- The ID numbers of the low-level backing and low-level forward filesets related to the fileset
- The version number of the fileset
- The allocation and allocation usage, in kilobytes, of the fileset
- The quota and quota usage, in kilobytes, of the fileset
- The day, date, and time when the fileset was created (replicated or backed up for a read-only or backup fileset)
- The day, date, and time when the contents of the fileset were last updated (same as the creation time for a read-only or backup fileset)

Examples

The following examples show output from the **fts lsheader** command when it is executed with the **-fast** option, with neither the **-fast** option nor the **-long** option, and with the **-long** option. All three examples display output primarily for the same fileset, *user.terry* (ID number **0,,196953**).

$ fts lsheader /.../abc.com/hosts/fs3 /dev/lfs1 -fast

```
0,,196953
0,,196956
    .
    .
0,,199845
0,,199846
```

$ fts lsheader /.../abc.com/hosts/fs3 /dev/lfs1

```
Total filesets on server fs3 aggregate lfs1 (ID 10): 16
user.terry           0,,196953 RW   5071 K alloc  8421 K quota On-line
user.wvh             0,,196956 RW   4955 K alloc  9371 K quota On-line
```

```
    .
    .
Total filesets on-line 15; total off-line 1; total busy 0
```

$ fts lsheader /.../abc.com/hosts/fs3 /dev/lfs1 -long

```
Total filesets on server fs3 aggregate lfs1 (ID 10): 16
user.terry 0,,196953 RW LFS     states 0x10010005   On-line
    fs3.abc.com, aggregate lfs1 (ID 10)
    Parent 0,,196953   Clone 0,,0   Backup 0,,196955
    llBack 0,,0        llFwd 0,,0   Version 0,,25963
    429496729 K alloc limit;      1252 K alloc usage
        15000 K quota limit;      9340 K quota usage
    Creation Tue Oct 15 16:45:16 1991
    Last Update Fri Nov 22 11:36:00 1991
  user.wvh 0,,196956 RW LFS     states 0x10010005  On-line
    .
    .
Total filesets on-line 15; total off-line 1; total busy 0
```

Related Information

Commands: **fts lsfldb(8dfs)**, **fts lsft(8dfs)**.

Files: **dfstab(4dfs)**.

fts lsmount

Purpose

fts lsmount – Lists the filesets associated with mount points

Synopsis

fts lsmount -dir *directory_name*… [**-help**]

Options

-dir *directory_name*
: Names each directory that serves as a mount point for a fileset. The last element in the specified pathname must be an actual name, not . (dot) or .. (dot dot).

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **fts lsmount** command displays the name of the fileset for which each directory specified with the **-dir** option is the mount point. The association between a mount point and a fileset is created with the **fts crmount** command; it is removed with the **fts delmount** command.

Privilege Required

The issuer must have read permission on each directory indicated with the **-dir** option, regardless of whether each indicated directory resides in a directory in a DCE LFS or non-LFS fileset.

Output

The **fts lsmount** command displays the following message for each directory that is a mount point:

```
'directory_name' is a mount point for fileset 'fileset_name'
```

where *directory_name* is the name of a directory specified with the **-dir** option, and *fileset_name* is the name of the fileset for which *directory_name* serves as a mount point. The command also provides the following information about the directory and fileset:

(number sign)
: Precedes *fileset_name* if *directory_name* is a regular mount point.

% (percent sign)
: Precedes *fileset_name* if *directory_name* is a read/write mount point.

! (exclamation point)
: Replaces *fileset_name* if the directory is a global root mount point (a mount point for the root of the DCE global namespace).

The **fts lsmount** command displays the following message for each directory that is not a mount point:

```
'directory_name' is not a mount point.
```

Examples

The following example lists the mount point *vijay*, which is a regular mount point for the fileset named *user.vijay*:

```
$ fts lsmount vijay
```

```
'vijay' is a mount point for fileset '#user.vijay'
```

Related Information

Commands: **fts crmount(8dfs)**, **fts delmount(8dfs)**.

fts lsquota

Purpose

fts lsquota – Shows quota and quota usage information for filesets and disk size and usage information for aggregates or partitions

Synopsis

fts lsquota [{**-path** {*filename* | *directory_name*}… | **-fileset** {*name* | *ID*}}] [**-cell** *cellname*] [{**-noauth** | **- localauth** }] [**-verbose**][**-help**]

Options

-path*filename* or *directory_name*
: Names a file or directory from each fileset about which quota, size, and usage information is to be displayed. Include filenames or directory names from different filesets if desired. It is not necessary to name more than one file or directory from the same fileset. Use this option or use **-fileset**; omit both options to display information about the fileset containing the current working directory.

-fileset *name* or *ID*
: Specifies the complete name or fileset ID number of each fileset about which quota, size, and usage information is to be displayed. Use this option or use **-path**; omit both options to display information about the fileset that contains the current working directory.

-cell *cellname*
: Specifies the cell with respect to which the command is to be run. The default is the local cell of the issuer of the command.

-noauth
: Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. Generally, the **-noauth** option is included if DFS authorization checking is disabled on a server machine on which administrative privilege is required or if the Security Service is unavailable. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **bos** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a

machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions during command execution.

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

Description

The **fts lsquota** command displays quota and quota usage information about filesets and disk size and usage information about the partitions or aggregates on which the filesets reside. Use the **- path** option to specify a file or directory on a fileset to see information about that fileset; use the **-fileset** option to specify the name or ID number of a fileset to see information about that fileset; omit both options to see information about the fileset that contains the current working directory.

For DCE LFS filesets, the **fts lsquota** command displays the quota and quota use (in kilobytes) and the percentage of the quota in use. For both DCE LFS and non-LFS filesets, this command displays the name of the fileset, information about the number of available kilobytes on the aggregate or partition on which the fileset resides, the number of kilobytes in use on the aggregate or partition, and the percentage of the aggregate or partition in use. It also reports whether the device is a DCE LFS aggregate or a non-LFS partition.

The size of a non-LFS fileset is equal to the size of the partition on which it resides. Therefore, the size and usage information displayed for the partition (non-LFS aggregate) in the output of the **fts lsquota** command equals the quota and quota usage information of the fileset on the partition. Using this command with a non-LFS fileset is analogous to using the UNIX **df** command with the partition on which the fileset resides. (Note that the **df** command can be used to display the size of exported DCE LFS aggregates and locally mounted DCE LFS filesets, but it cannot be used to display the size of a DCE LFS fileset that is not mounted locally.)

The **fts lsheader** and **fts lsft** commands can be used to display the quota of a DCE LFS fileset. The **fts aggrinfo** command can be used to display the total disk space on an aggregate and the amount currently available.

By default, every newly created DCE LFS fileset has a quota of 5000 kilobytes. The **fts setquota** command can be used to increase or decrease the quota of a DCE LFS

fileset. Because the quota of a DCE LFS fileset does not represent the amount of physical data stored on the fileset, it can be larger than the size of the aggregate on which the fileset resides. Similarly, the combined quotas of all filesets on an aggregate can be larger than the size of the aggregate.

The quota of a non-LFS fileset cannot be changed via DFS. (The **fts setquota** command works only with DCE LFS filesets.)

Privilege Required

No privileges are required.

Output

This command displays the following information about each specified fileset:

- The name of the fileset
- The quota, in kilobytes, of the fileset (DCE LFS only)
- The number of kilobytes of the quota currently in use on the fileset (DCE LFS only)
- The percentage of the quota currently in use on the fileset (DCE LFS only)
- The percentage of available disk space currently in use on the aggregate or partition on which the fileset resides
- The number of kilobytes of disk space in use and available on the aggregate or partition on which the fileset resides
- The file system type of the aggregate (**LFS** or **non-LFS**)

If the fileset quota usage rises above 90% or the aggregate or partition usage rises above 97%, the appropriate percentage is indicated with < < and the message <<**WARNING** is displayed at the end of the output line.

Note: Because each non-LFS partition contains a single fileset, the information displayed for a non-LFS partition applies to the single non-LFS fileset it houses. Ignore the quota, quota usage, and quota usage percentage values displayed for a non-LFS fileset; they are always 0 (zeros). Consult the disk size, usage, and percentage values displayed for the partition on which the non-LFS fileset resides to determine the corresponding values for the fileset.

Examples

The following command lists quota and quota usage information for the fileset that contains the directory named */.../abc.com/fs/usr/terry* , and it displays size and usage information for the aggregate that contains this fileset. The command also displays size and usage information for the partition that contains the directory named */.../abc.com/fs/usr/jlw* . The first directory resides on the DCE LFS fileset named *user.terry*; the quota of the DCE LFS fileset is less than the size of the aggregate on which it is located. The second directory resides on the non-LFS fileset named *user.jlw*; the quota of the non-LFS fileset is the same as the size of the partition on which it is located.

```
$ fts lsq /.../abc.com/fs/usr/terry /.../abc.com/fs/usr/jlw
```

```
Fileset Name   Quota   Used   % Used   Aggregate
user.terry     15000   5071      34%     86% = 84538/98300 (LFS)
user.jlw           0      0       0%     84% = 8448/10000 (non-LFS)
```

The following command lists quota and usage information for the DCE LFS fileset named *user.jean*, and size and usage information for the aggregate on which the fileset resides. The **<<WARNING** message directs the issuer's attention to the fact that the percentage of the quota in use on the indicated fileset is well above the warning level of 90%.

```
$ fts lsq -f user.jean
```

```
Fileset Name   Quota   Used   % Used   Aggregate
user.jean       5000   4955     99%<<  9
2% = 87436/98300 (LFS) <<WARNING
```

Related Information

Commands: **fts aggrinfo(8dfs)**, **fts lsft(8dfs)**, **fts lsheader(8dfs)**, **fts setquota(8dfs)**.

fts lsreplicas

Purpose

fts lsreplicas – Displays the statuses of DCE LFS fileset replicas

Synopsis

fts lsreplicas -fileset {*name* | *ID*} {**-all** | **-server** *machine*} [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset{*name* | *ID*}
Specifies the complete name or fileset ID number of the fileset whose replicas are to be checked.

-all
Specifies that all replicas of **-fileset** are to be checked. Use this option or use **-server**.

-server*machine*
Names a specific File Server machine on which replicas of **-fileset** are to be checked. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. Use this option or use **-all**.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts lsreplicas** command shows the replication statuses of read-only replicas of the read/write DCE LFS fileset specified with the **-fileset** option. Use the command's options to check replicas of **-fileset** as follows:

- To check the status of the replica stored on a specific File Server machine, specify the name of the machine with the **-server** option.
- To check the status of all replicas, specify the **-all** option.

If Release Replication is used for a read/write fileset, use the **fts release** command to place replicas of the fileset at replication sites. (If Scheduled Replication is used, the Replication Server automatically places replicas at replication sites according to specified parameters.) Use the **fts update** command to request that the Replication Server make an immediate update of the replicas of any read/write fileset.

Use the **fts statrepserver** command to check the status of the Replication Server process on a specific File Server machine. Use the **fts addsite** command to add a replication site; use the **fts rmsite** command to remove a replication site.

Privilege Required

No privileges are required.

Examples

The following command displays the status of each replica of the read/write fileset named **rs_aix32.bin**:

```
$ fts lsr rs_aix32.bin -a
```

Related Information

Commands: **fts addsite(8dfs)**, **fts release(8dfs)**, **fts rmsite(8dfs)**, **fts statrepserver(8dfs)**, **fts update(8dfs)**.

fts lsserverentry

Purpose

fts lsserverentry – Lists a server entry from the FLDB

Synopsis

fts lsserverentry {**-server** *machine* | **-all** } [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server*machine*
: Specifies the name of the server machine whose entry in the Fileset Location Database (FLDB) is to be displayed. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. Use this option or use the **-all** option.

-all
: Specifies that the entries for all server machines in the FLDB are to be displayed. Use this option or use the **-server** option.

-cell *cellname*
: Specifies the cell from whose FLDB the specified server entries are to be listed. The default is the local cell of the issuer of the command.

-noauth
: Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose
: Directs **fts** to provide detailed information about its actions as it executes the command.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

Description

The **fts lsserverentry** command displays server entry information from the FLDB. If the **-server** option is specified, entry information from the FLDB for only the indicated server machine is displayed. If the **-all** option is specified, entry information from the FLDB for all server machines is displayed.

Use the **fts crserverentry** command to create a server entry in the FLDB. Use the **fts edserverentry** command to modify a server entry in the FLDB. Use the **fts delserverentry** command to remove a server entry from the FLDB.

Privilege Required

No privileges are required.

Examples

The following command displays the server entry from the FLDB for a server machine named **fs1**:

```
$ fts lsserverentry /.../abc.com/hosts/fs1
```

Related Information

Commands: **fts crserverentry(8dfs)**, **fts delserverentry(8dfs)**, **fts edserverentry(8dfs)**.

fts move

Purpose

fts move – Moves a read/write DCE LFS fileset to another site

Synopsis

fts move -fileset {*name* | *ID*} **-fromserver** *source_machine* **-fromaggregate** *source_name* **-toserver** *dest_machine* **-toaggregate** *dest_name* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}
Specifies the complete name or the fileset ID number of the read/write fileset to be moved.

-fromserver *source_machine*
Names the File Server machine on which the fileset currently resides. Specify the File Server machine by its DCE pathname, its host name, or its IP address.

-fromaggregate*source_name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate on which the fileset currently resides. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate in the *dcelocal*/**var/dfs/dfstab** file.

-toserver *dest_machine*
Names the File Server machine to which the fileset is to be moved. Specify the File Server machine by its DCE pathname, its host name, or its IP address.

-toaggregate *dest_name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate to which the fileset is to be moved. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate in the **dfstab** file.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts move** command moves the indicated read/write DCE LFS fileset from its current site (specified with the **-fromserver** and **-fromaggregate** options) to the destination site (specified with the **-toserver** and **-toaggregate** options). The command decrements the number of fileset entries recorded as residing on the machine indicated with the **-fromserver** option in the Fileset Location Database (FLDB) entry for the machine, and it increments the number of fileset entries recorded as residing on the machine specified with the **-toserver** option in the FLDB entry for that machine. It also automatically removes the backup version of the fileset, if it exists, from the current site. To create a new backup version at the destination site, use the **fts clone** command.

It is not possible to move a read-only or backup fileset. For read-only filesets, the corresponding action is to create a new replication site with the **fts addsite** command and remove an existing one with the **fts rmsite** command. Because the backup version of a read/write fileset is automatically deleted when its read/write source is moved, a backup fileset can be moved only by moving its read/write source fileset and issuing the **fts clone** command to create a new backup version.

Furthermore, it is not possible to move a fileset from a site in one cell to a site in another cell. Filesets can be moved only between two sites in the same cell. The filesets are assumed to reside in the local cell of the issuer unless the name of a foreign cell is specified with the **-cell** option.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be moved to a different File Server machine. It can be moved only to a different aggregate on the same File Server machine. If the command is used to move a DCE LFS fileset that is locally mounted, its **-fromserver** and **-toserver** options must name the same File Server machine; otherwise, the command fails. (To move a locally mounted fileset to a different machine, remove its local mount point before issuing this command.)

In addition, because the backup version of a fileset is removed when its read/write version is moved, you cannot move a fileset (not even to another aggregate on the same File Server machine) if its backup version is mounted locally. You must remove the backup version's local mount point before moving the fileset.

Privilege Required

The issuer must be listed in the **admin.ft** files on both the source and destination machines. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entries for the source machine, the destination machine, and any machines on which replicas of the fileset reside. In addition, the source machine (**-fromserver**) must be listed in the **admin.ft** file on the destination machine (**-toserver**).

Examples

The following command moves the fileset *user.smith* from **/dev/lv01** on **fs3** to **/dev/lv02** on **fs7**:

```
$ fts move user.smith /.../abc.com/hosts/fs3 /dev/lv01 /.../abc.com/hosts/fs7 /dev/lv02
```

Related Information

Commands: **fts addsite(8dfs)**, **fts clone(8dfs)**, **fts delete(8dfs)**, **fts release(8dfs)**.

Files: **dfstab(4dfs)**.

fts release

Purpose Initiates Release Replication by placing read-only version of a read/write DCE LFS fileset at the local site

Synopsis **fts release -fileset** {*name* | *ID*} [**-wait**][**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}
Specifies the complete name or fileset ID number of the read/write fileset to be replicated locally (cloned if the local replication site is defined on the same aggregate as the read/write fileset). Once the fileset is replicated locally, the Replication Servers at the fileset's replication sites copy the replica to their sites.

-cell*cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-wait Directs the command to not terminate (return a prompt) until all replicas are up to date.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts release** command is used to initiate the replication process for a fileset that uses Release Replication. The command "releases" a new read-only copy of the DCE LFS fileset specified with the **-fileset** option. It places the new read-only copy at the local replication site defined on the same File Server machine as the read/write fileset. The Replication Servers at each of the fileset's replication sites (specified File Server machines and aggregates) then update the copies of the read-only replica at the sites on their respective machines.

Note that, as with updating a new version of a fileset that uses Scheduled Replication, releasing a fileset that uses Release Replication does not ensure immediate access to data in the new version of the replica. A Cache Manager continues to provide data cached from the old version of the replica until the MaxAge for the fileset expires or until the Cache Manager needs to access data from the replica that it has not already cached.

To gain immediate access to data in the new version of the replica, issue the **cm flush** or **cm flushfileset** command to flush the old data from the cache. This forces the Cache Manager to replace data it has cached from the replica. Replication Servers begin replication in parallel; however, until all replicas have been updated, you cannot directly force the Cache Manager to access data from the new version of the replica.

Before the **fts release** command can be used, the **fts setrepinfo** command must be used to define the replication parameters for the read/write fileset. If Release Replication is to be used, the **-release** option must be specified with the **fts setrepinfo** command. The **fts addsite** command must also be used to define the replication sites for the read/write fileset. For Release Replication, the replication site on the same File Server machine as the read/write fileset must be defined first. The read/write fileset must have at least one replication site defined before the **fts release** command can be issued. The replication parameters and sites for a read/write fileset are recorded in the fileset's entry in the Fileset Location Database (FLDB).

The **fts release** command does not alter the replication type and parameters defined for the specified fileset. The command can be used only with a fileset that uses Release Replication; it returns an error if the specified fileset uses Scheduled Replication. The **fts update** command can be used to request an immediate update of the replicas of a fileset that uses Scheduled Replication.

Use the **fts lsreplicas** command to check the status of replicas. Use the **fts statrepserver** command to check the status of the Replication Server on a File Server machine.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine on which the source read/write fileset is stored. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entries for the machine on which the source fileset resides and all machines on which the read-only replicas are to reside.

Examples

The following command releases (initiates Release Replication for) the fileset named **pmax_osf1.bin**:

```
$ fts release pmax_osf1.bin
```

Related Information

Commands: **cm flush(8dfs)**, **cm flushfileset(8dfs)**, **fts addsite(8dfs)**, **fts lsreplicas(8dfs)**, **fts setrepinfo(8dfs)**, **fts statrepserver(8dfs)**, **fts update(8dfs)**.

fts rename

Purpose

fts rename – Renames a fileset

Synopsis

fts rename -oldname *oldname* **-newname** *newname* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-oldname*oldname*

Specifies the current name of the read/write fileset.

-newname*newname*

Specifies the new name for the read/write fileset. The name must be unique within the local cell, and it should be indicative of the fileset's contents. The following characters can be included in the name of a fileset:

- All uppercase and lowercase alphabetic characters (a to z, and A to Z)
- All numerals (0 to 9)
- The . (dot)
- The – (dash)
- The _ (underscore)

The name must contain at least one alphabetic character or an _ (underscore) to differentiate it from an ID number. It can be no longer than 102 characters. This length does not include the **.readonly** or **.backup** extension that is added automatically when a read-only or backup version of a DCE LFS fileset is created. Note that the **.readonly** and **.backup** extensions are reserved for use with read-only and backup DCE LFS filesets, so you cannot specify a fileset name that ends with either of these extensions.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts rename** command changes the name of the read/write fileset specified with **-oldname** to the name specified with **-newname**. The names of the read/write fileset's read-only copies and backup copy, if any, automatically change to match.

After issuing this command, the issuer must correct any mount points that refer to the old fileset name. This is done by removing each old mount point with the **fts delmount** command and creating a replacement for each with the **fts crmount** command. (These commands require that the issuer have the necessary file system permissions for the operations.)

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine on which the read/write fileset resides. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine on which a version of the fileset to be renamed resides.

Examples

The following command changes the incorrect fileset name **osf1.bin** to the correct fileset name **pmax_osf1.bin**:

```
$ fts rename osf1.bin pmax_osf1.bin
```

Related Information

Commands: **fts crmount(8dfs)**, **fts delmount(8dfs)**.

fts restore

Purpose

fts restore – Converts a dump file from bytestream format to fileset format and places it in the file system

Synopsis

fts restore -ftname *name* **-server** *machine* **-aggregate** *name* [**-file** *filename*] [**-ftid** *ID*] [**-overwrite**][**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-ftname*name*

Specifies the name of the fileset to which the file is to be restored. If the file is to be restored as a new fileset, the name must be unique within the local cell, and it should be indicative of the fileset's contents. The following characters can be included in the name of a fileset:

- All uppercase and lowercase alphabetic characters (a to z, and A to Z)
- All numerals (0 to 9)
- The . (dot)
- The – (dash)
- The _ (underscore)

The name must contain at least one alphabetic character or an _ (underscore) to differentiate it from an ID number. It can be no longer than 102 characters. This length does not include the **.readonly** or **.backup** extension that is added automatically when a read-only or backup version of a DCE LFS fileset is created. Note that the **.readonly** and **.backup** extensions are reserved for use with read-only and backup DCE LFS filesets, so you cannot specify a fileset name that ends with either of these extensions.

-server *machine*
Specifies the File Server machine to which the file is to be restored. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate or partition on **-server** to which the file is to be restored. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or partition in the *dcelocal* **/var/dfs/dfstab** file.

-file *filename*
Specifies the complete pathname of the file to be restored. If a complete pathname is not provided, the file is assumed to reside in the current working directory. If this option is omitted, the data is read from standard input (**stdin**).

-ftid *ID* Specifies the fileset ID number to assign to the restored fileset. If this option is omitted and an existing fileset is to be overwritten, the fileset ID number of the existing fileset is used. If it is omitted and a new fileset is to be created, the FL Server allocates a new fileset ID number for the fileset. Use this option only when restoring a dump file as a DCE LFS fileset; use it sparingly and with great care. Omit this option when restoring a dump file as a non-LFS fileset.

-overwrite Specifies that the file to be restored can overwrite an existing fileset. If this option is omitted, the command exits without overwriting an existing fileset. You must use this option to overwrite a previously restored version of a fileset with an incremental dump of the same fileset; more information about conditions that must be met if a fileset is to be overwritten by an incremental dump is provided later in this reference page. You must also use this option to restore a dump file as a non-LFS fileset.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts restore** command translates a dump file created previously with the **fts dump** command from a bytestream format to a fileset format appropriate for the machine specified with the **-server** option. The dump file to be restored is indicated with the **-file** option. If this option is omitted, the data to be restored is read from **stdin**.

The fileset contained in the dump file can be restored as a new read/write DCE LFS fileset by specifying a name and site for the new fileset. The command assigns the fileset the name indicated with the **-ftname** option. It restores it to the site specified with the **-server** and **-aggregate** options. The dump file must contain the full dump of a fileset if it is to be restored as a new fileset.

Alternatively, the fileset contained in the dump file can be restored over an existing read/write version of the same fileset by specifying the name and site of the existing fileset. The command resets the creation time stored in the fileset's header to match the restore time. The **-overwrite** option must be used to specify that the dump file is to overwrite the existing fileset. If this option is omitted, the command displays an error message and exits instead of overwriting the existing fileset.

When restoring a dump file as a non-LFS fileset, the fileset must already exist for the non-LFS partition on which it resides to be exported to the DCE namespace. In this case, you must use the **-overwrite** option to overwrite the existing non-LFS fileset (even if the fileset to be overwritten contains no data).

If you are overwriting an existing fileset with an incremental dump, the fileset to be overwritten should initially have been restored as a new read/write fileset from a full dump. Also, both the dump file to be restored and the full dump that initially produced the read/write fileset to be overwritten must be dumps of the same fileset. (A full dump

of a fileset can be restored to overwrite an existing fileset, but the restored dump file overwrites all data in the existing fileset. An incremental dump of a fileset cannot be restored to overwrite an existing fileset that was not created from the restoration of a full dump.)

Multiple incremental dumps of a fileset can be restored to overwrite the same existing fileset provided the following conditions are true:

- The fileset to be overwritten must not have been modified (that is, no files added, removed, or saved, and no ACLs changed) since its most recent restoration from a full or incremental dump.
- The dump file to be restored must have been created *from* a date and time (as specified with the **-date** or **-version** option of the **fts dump** command) *no later* than the date and time at which the most recently restored dump of the fileset to be overwritten was dumped.
- The dump file to be restored must have been created *at* a date and time *later* than the date and time at which the most recently restored dump of the fileset to be overwritten was dumped.

The last two conditions indicate that the span of time recorded in the incremental dump to be restored must overlap and extend the span of time recorded in the fileset to be overwritten. For example, suppose the following dumps were made of a fileset: a full dump was made on 1 January 1992, an incremental dump from 31 December 1991 was made on 7 January 1992, and an incremental dump from 6 January 1992 was made on 14 January 1992. The following sequence of operations represents the only possible way to restore the fileset from all three of these dumps:

1. The full dump made on 1 January is restored as a new read/write fileset.
2. The incremental dump made on 7 January is restored to overwrite the read/write version of the fileset made from the full dump.
3. The incremental dump made on 14 January is restored to overwrite the read/write version of the fileset that includes data from the full and first incremental dumps.

No other sequence of restore operations involving all three dumps is possible. Any other sequence of steps will undoubtedly result in some or all of the data in the fileset being inaccessible or inconsistent.

When restoring a dump file as a DCE LFS fileset, a fileset ID number can be assigned to the restored fileset with the **-ftid** option. This is generally not recommended unless there is good reason to believe that an available fileset ID number can be specified. If the **-ftid** option is omitted, an overwritten DCE LFS fileset retains its current ID

number, or the FL Server allocates a new ID number for a new DCE LFS fileset restored from a dump file. If a new fileset ID number is assigned or allocated, the FL Server increments the number of fileset entries recorded as residing on the specified File Server machine in the Fileset Location Database (FLDB) entry for the server.

When restoring a dump file as a non-LFS fileset, do not use the **-ftid** option. Omit the option to continue to use the fileset ID number specified for the non-LFS fileset in the entry for its partition in the **dfstab** file. (Note that the restored dump file overwrites all data on the non-LFS partition.)

If a new fileset is created, use the **fts crmount** command to create a mount point for the fileset, making it visible in the DCE namespace. If an existing DCE LFS fileset is overwritten with this command, use the **fts update** command to release new read-only replicas based on the new version of the fileset, and use the **fts clone** command to create a new backup version of the fileset, as necessary.

You can use the **fts restore** command to restore a dump file to any type of fileset (DCE LFS or non-LFS), regardless of the type of fileset from which it was created. For example, a dump file of a DCE LFS fileset can be restored to a DCE LFS fileset or to any type of non-LFS fileset. Similarly, a dump file of a non-LFS fileset can be restored to a DCE LFS fileset or to a different type of non-LFS fileset. In any case, the contents of the dump file are translated into the appropriate format for the file system to which they are restored. (Refer to your vendor's documentation to verify the level of support for dump and restore operations between different types of file systems.)

Note that incompatible information may be lost when a fileset is dumped and restored between different types of file systems. For example, ACLs on objects in a DCE LFS fileset may be lost if the fileset is restored to a file system that does not support ACLs.

You cannot restore a fileset dumped in one cell to a site in another cell.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine specified by **-server** and must have the read permission on the dump file. The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine on which a version of the fileset is recorded as residing in the FLDB (generally only **-server** unless an existing fileset is to be overwritten).

Cautions

Ensure that all of the conditions discussed in the description section are met before restoring an incremental dump of a fileset over an existing fileset. Violation of any of

the conditions is very likely to result in inaccessibility or inconsistency of some or all of the data in the fileset.

Examples

The following example restores a file, */tmp/smith.013191.dump*, that contains an incremental dump of a fileset over an existing read/write version of the same fileset, *user.smith*. The incremental dump was created using a start date and time no later than the date and time when the most recently restored version of the fileset to be overwritten was dumped, and it was dumped at a date and time later than the date and time when the most recently restored version of the fileset to be overwritten was dumped. Also, the fileset to be overwritten has not been modified since it was last restored. The **-ftid** option is omitted, so the fileset retains its current fileset ID number.

```
$ fts restore user.smith /.../abc.com/hosts/fs1 lfs1 /tmp/smith.013191.dump -overwrite
```

The following command takes input directly from an **fts dump** command to create a new read/write fileset, *user.terry*, from an existing fileset, *user.smith*. The **-file** option is omitted from the **fts dump** command to send the output to **stdout**, and it is omitted from the **fts restore** command to read the input from **stdin**. (The information is "piped" from one command to the next.) The **-ftid** option is again omitted from the **fts restore** command; this time the FL Server allocates a new ID number for the fileset.

```
$ fts dump user.smith -time 0 | fts restore user.terry /.../abc.com/hosts/fs1 lfs1
```

Related Information

Commands: **fts clone(8dfs)**, **fts crmount(8dfs)**, **fts dump(8dfs)**, **fts update(8dfs)**.

Files: **dfstab(4dfs)**.

fts rmsite

Purpose

Removes a replication site and read-only DCE LFS fileset

Synopsis

fts rmsite -fileset {*name* | *ID*} **-server** *machine* **-aggregate** *name* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}
Specifies the complete name or fileset ID number of the read/write fileset for which a replication site and the read-only fileset stored at that site are to be removed.

-server *machine*
Specifies the File Server machine to be removed as a replication site. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate to be removed as a replication site. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate in the *dcelocal* **/var/dfs/dfstab** file. If the aggregate is not currently exported or has been detached, you must specify the aggregate ID.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this

option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts rmsite** command removes a replication site currently defined for the read/write DCE LFS fileset specified with the **-fileset** option. The **-server** and **-aggregate** options are used to specify the replication site to be removed. The command performs the following actions:

- It removes the definition of the replication site from the Fileset Location Database (FLDB) entry for the fileset.
- It decrements the number of fileset entries recorded as residing on the File Server machine specified with **-server** in the FLDB entry for the server.
- If the indicated fileset uses Release Replication and the specified site is on the same File Server machine as the read/write fileset, the command removes the replica (if it exists); see the **Cautions** section for more information. For any other replica, the command instructs the Replication Server at the site to remove the replica.

Other replication sites of the read/write fileset are not affected. If the command is used to remove a fileset's last replication site, the status flag for the read-only version in the fileset's FLDB entry is set to **invalid**. If it is used to remove the last existing version of a fileset, the fileset's entire FLDB entry is removed.

Before you use the **fts delete** command to remove the read/write (and backup) version of a fileset, use the **fts rmsite** command to remove the fileset's replication sites. If Release Replication was used for the fileset, use the **fts rmsite** command to remove the replication site (and replica) stored on the same File Server machine as the read/write fileset as well.

If the aggregate on which the replication site is defined is not currently exported or has been detached with the **dfsexport** command, you must specify the aggregate ID

of the aggregate; otherwise, the **fts rmsite** command cannot remove the replication site. If the aggregate is not exported or has been detached, the Replication Server on the File Server machine on which the aggregate resides stops trying to maintain the replica at the site once the **fts rmsite** command is issued, and it removes the replica from the site once the aggregate is again exported.

Replication sites are added with the **fts addsite** command. The replication type for a read/write fileset is set or changed with the **fts setrepinfo** command.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine that houses a version of the fileset for which the replication site and replica are to be removed. The issuer must also be listed in the **admin.ft** file on the machine specified by **-server** if the following are true:

- Release Replication is used for the fileset.
- The replication site on the same File Server machine as the read/write fileset is to be removed (in which case **-server** names the File Server machine on which the read/write fileset resides).
- A replica actually exists at the specified replication site.

Cautions

If you use Release Replication and you remove the read-only fileset that is on the same File Server machine as the read/write source, all other read-only filesets become unavailable upon the expiration of the fileset's FailAge parameter. The FailAge parameter is set using the **fts setrepinfo** command.

Examples

The following command removes the replication site on the aggregate **/dev/lv01** of the File Server machine **fs5** from the FLDB entry for the fileset named **rs_aix32.bin**. A replica of **rs_aix32.bin** that resides at the site is also removed.

```
$ fts rmsite rs_aix32.bin /.../abc.com/hosts/fs5 /dev/lv01
```

Related Information

Commands: **fts addsite(8dfs)**, **fts delete(8dfs)**, **fts setrepinfo(8dfs)**.

Files: **dfstab(4dfs)**.

fts setprotectlevels

Purpose

fts setprotectlevels – Sets advisory DCE remote procedure call (RPC) authentication levels for a specified fileset.

Synopsis

fts setprotectlevels -fileset {*name* | *ID*} [**-minlocalprotectlevel** *level*] [**-maxlocalprotectlevel** *level*] [**-minremoteprotectlevel** *level*] [**-maxremoteprotectlevel** *level*] [**-cell** *cellname*] [**-verbose**][**-noauth** | **-localauth**][**-help**]

Options

-fileset {*name*|*ID*}
Specifies a fileset either by its name or volume ID.

-minlocalprotectlevel *level*
Specifies the advisory lower bound DCE RPC authentication level for the specified fileset (used by DFS client Cache Managers within the same cell). The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-maxlocalprotectlevel *level*
Specifies the advisory upper bound DCE RPC authentication level for the specified fileset (used by DFS client Cache Managers within the same cell). The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-minremoteprotectlevel *level*
Specifies the advisory lower bound DCE RPC authentication level for the specified fileset (used by DFS client Cache Managers within foreign cells). The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of

that string. For a description of the various DCE RPC levels, see the Description section.

-maxremoteprotectlevel *level*
Specifies the advisory upper bound DCE RPC authentication level for the specified fileset (used by DFS client Cache Managers within foreign cells). The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Description section.

-cell *cellname*
Specifies the cell as *cellname* within which the specified fileset resides.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts setprotectlevels** command adjusts the minimum and maximum advisory DCE RPC authentication level bounds for a specified fileset. These bounds are used to bias a Cache Manager to a higher or lower security level when accessing the specified fileset. However, the bounds are simply advisory in that if the Cache Manager's security level settings are outside of the advisory bounds, the Cache Manager can cross the advisory and continue negotiating with a File Server. In this case, the Cache Manager's minimum security level (set with the **dfsd** or **cm setprotectlevels** command) and the File Server's maximum security bound (set with the **fxd** command) become the "hard"

limits. Note that if the **fts setprotectlevels** bounds fall outside of File Server bounds, the File Server bounds take precedence.

In practice, when a Cache Manager must access a given fileset it first consults a Fileset Location (FL) Server for the location of that fileset (or any replicas if it is replicated read-only fileset). Along with the location, the Cache Manager also receives the applicable minimum and maximum advisory bounds for that fileset. The Cache Manager then checks its initial authentication level and compares that to the range defined by the bounds. The Cache Manager then adjusts its initial authentication level as follows:

- If the Cache Manager's initial authentication level is within the range defined by the advisory bounds, the initial level is used without adjustment.
- If the Cache Manager's initial authentication level is above the maximum advisory bound, the Cache Manager adjusts the initial level to match the advisory upper bound. However, the Cache Manager will not adjust its authentication level below its own minimum setting.
- If the Cache Manager's initial authentication level is below the minimum advisory bound, the Cache Manager adjusts the initial level to match the advisory lower bound.

The negotiation process to set an RPC authentication level now occurs as usual between the Cache Manager and File Server. The Cache Manager sends an RPC using the initial authentication level (which may have been adjusted because of the advisory bounds) to the File Server. If the initial authentication level is outside the minimum or maximum bounds set at the File Server, the File Server returns a response to the Cache Manager specifying that the authentication level is either too low or too high. The Cache Manager then decreases or increases its authentication level accordingly and retries the RPC. This process continues until the Cache Manager either adjusts its RPCs to an acceptable security level or the File Server requests a security level below the minimum set at the Cache Manager (causing the Cache Manager to refuse communications with the File Server). Once the Cache Manager and File Server have negotiated a security level, the Cache Manager stores this information so that it does not need to renegotiate this level for further communications with the File Server.

Note that the use of this command does not preclude communication with Cache Managers running earlier versions of DCE.

The various authentication levels are set by specifying either an integer value between 0 and 6, a complete string specifying the authentication level, or an abbreviation of

that string as the *level* argument for the various command options. The following lists the various authentication levels:

- **0** or **default** or **rpc_protect_level_default**: Use the DCE default authentication level.
- **1** or **none** or **rpc_protect_level_none**: Perform no authentication.
- **2** or **connect** or **rpc_protect_level_connect**: Authenticate only when the Cache Manager establishes a connection with the File Server.
- **3** or **call** or **rpc_protect_level_call**: Authenticate only at the beginning of each RPC received.
- **4** or **pkt** or **rpc_protect_level_pkt**: Ensure that all data received is from the expected host.
- **5** or **pkt_integrity** or **rpc_protect_level_pkt_integrity**: Authenticate and verify that none of the data transferred has been modified.
- **6** or **pkt_privacy** or **rpc_protect_level_pkt_privacy**: Perform authentication as specified by all of the previous levels and also encrypt each RPC argument value.

Note that there is a trade-off between selecting higher security and performance. The higher levels of security require more overhead and increase the response time in file operations with File Servers.

Privilege Required

The issuer must have FLDB administration privileges or must be in the owner group for the File Server.

Examples

The following command sets the following authentication values:

- The maximum advisory authentication level for communication with Cache Managers in the local cell is set to packet integrity.
- The minimum advisory authentication level for communication with Cache Managers in the local cell is set to packet.
- The maximum advisory authentication level for communication with Cache Managers in foreign cells is set to packet security.

- The minimum advisory authentication level for communication with Cache Managers in foreign cells is set to packet security.

$ fts setprotectlevels -fileset richland.12 -maxlocalprotectlevel 5 -minlocalprotectlevel 4 -maxremoteprotectlevel 6 -minremoteprotectlevel 6

Related Information

Commands: **fts getprotectlevels(8dfs)**, **fxd(8dfs)**, **dfsd(8dfs)**, **fts setprotectlevels(8dfs)**

fts setquota

Purpose

fts setquota – Sets the maximum quota for a read/write DCE LFS fileset

Synopsis

fts setquota {**-path** {*filename* | *directory_name*} | **-fileset** {*name* | *ID*}} **-size** *kbytes* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-path {*filename* | *directory_name*}
Names a directory or file located on the read/write fileset whose quota is to be set. Use this option or use **-fileset**.

-fileset {*name* | *ID*}
Specifies the complete name or fileset ID number of the read/write fileset whose quota is to be set. Use this option or use **-path**.

-size *kbytes* Specifies the maximum amount of disk space that all of the files and directories in the read/write fileset can occupy. This includes files and directories in the read/write version of the fileset that are actually pointers to disk blocks in the backup or read-only version of the fileset. Specify the value in 1-kilobyte blocks. (A value of 1024 kilobytes is 1 megabyte.) By default, every newly created fileset has a quota of 5000 kilobytes.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database).

You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts setquota** command sets the quota limit for a read/write DCE LFS fileset. (It cannot be used to set the quota for a non-LFS fileset or for a read-only or backup DCE LFS fileset.) The fileset whose quota is to be set can be indicated by specifying the name of a file or directory in the fileset with the **-path** option or by indicating the fileset directly with the **-fileset** option.

Quota refers to the amount of disk space occupied by all of the files and directories in the read/write version of the fileset. This includes files and directories in the read/write version of the fileset that are actually pointers to disk blocks in the backup or read-only version of the fileset. Do not confuse quota with allocation; the latter identifies the amount of disk space occupied by the data that a fileset actually houses, excluding those files and directories that are pointers to disk blocks in another version of the fileset.

By default, every newly created fileset has a quota of 5000 kilobytes. This command increases or decreases a fileset's quota to the number of kilobytes specified with the **-size** option. Because it does not represent the amount of physical data the fileset contains, a fileset's quota can be larger than the size of the aggregate it resides on. Similarly, the sum of the quotas of all filesets on an aggregate can exceed the size of the aggregate.

The **fts lsft**, **fts lsheader**, and **fts lsquota** commands display, among other things, the current quota for a fileset.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine on which the fileset is stored.

Examples

The following command sets the quota for the fileset that contains the directory named */usr/terry* to 15,000 kilobytes:

```
$ fts setq /usr/terry 15000
```

Related Information

Commands: **fts lsft(8dfs)**, **fts lsheader(8dfs)**, **fts lsquota(8dfs)**.

fts setrepinfo

Purpose

fts setrepinfo – Sets or changes replication type and parameters for a read/write DCE LFS fileset

Synopsis

fts setrepinfo -fileset {*name* | *ID*} [{**-release** | **-scheduled** }] [**-change**][**-maxage** *interval*] [**-failage** *interval*] [**-reclaimwait** *interval*] [**-minrepdelay** *interval*] [**-defaultsiteage** *interval*] [**-clear**][**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}

Specifies the complete name or fileset ID number of the read/write source fileset for which the replication type and parameters are to be set or changed. This command is used to set parameters for either Release or Scheduled Replication.

-release Specifies that Release Replication is to be used with the fileset indicated with the **-fileset** option. When initially defining a fileset's replication parameters, use this option or use the **-scheduled** option. Afterward, omit both options when modifying the fileset's replication parameters without changing its replication type.

To change a fileset's replication type (from Release to Scheduled, or from Scheduled to Release), include both the **-change** option and either the **-release** or **-scheduled** option to indicate the new type of replication to be used with the fileset.

-scheduled Specifies that Scheduled Replication is to be used with the fileset indicated with the **-fileset** option. When initially defining a fileset's replication parameters, use this option or use the **-release** option. Afterward, omit both options when modifying the fileset's replication parameters without changing its replication type.

To change a fileset's replication type (from Release to Scheduled, or from Scheduled to Release), include both the **-change** option and either

the **-release** or **-scheduled** option to indicate the new type of replication to be used with the fileset.

-change Specifies that the type of replication currently used with the fileset indicated with the **-fileset** option is to be changed. Include the **-release** option to change the fileset's replication type from Scheduled to Release; include the **-scheduled** option to change the fileset's replication type from Release to Scheduled.

Omit this option when specifying the **-release** or **-scheduled** option to initially set a fileset's replication type. Also omit this option when changing a fileset's replication parameters without changing its replication type.

-maxage *interval*
Specifies the amount of time the Cache Manager distributes data cached from a read-only replica without attempting to verify that the data is current. The Replication Server maintains information about the currentness of a read-only replica, which it communicates to the Cache Manager via the File Exporter. For Scheduled Replication, a replica must remain current with respect to the read/write source fileset; for Release Replication, a replica must remain current with respect to the read-only fileset that resides on the same File Server machine as the read/write source fileset. The default is 2 hours. An effective value must be greater than or equal to 2 minutes. *Applicable to Release and Scheduled Replication.*

-failage *interval*
Specifies the amount of time the Cache Manager distributes data cached from a read-only replica if that data cannot be verified as current. The difference between FailAge and MaxAge is the amount of time the Cache Manager continues to distribute data cached from a read-only replica after that data cannot be verified as current. The default is 1 day or twice the MaxAge, whichever is larger. An effective value must be greater than or equal to the MaxAge. *Applicable to Release and Scheduled Replication.*

-reclaimwait*interval*
Specifies the amount of time the File Exporter waits before it reclaims storage space from deleted files—those not referred to by a directory (ReclaimWait). It also determines the frequency of the Cache Manager's keep-alive messages to the Replication Server.

The Cache Manager sends keep-alive messages to indicate that it is still using files on a read-only replica. A file being accessed from a replica remains available as long as the Cache Manager continues to notify the Replication Server that the file is still in use and the Replication Server continues to forward these notifications to the File Exporter. This is true even if the file has been removed from all directories on the read/write fileset in the interim. To prevent the File Exporter from reclaiming storage space occupied by deleted files, the Cache Manager sends keep-alive messages more frequently than the ReclaimWait interval. The default is 18 hours. An effective value must be greater than 2 hours; do not specify a value less than 90 minutes. *Applicable to Release and Scheduled Replication.*

-minrepdelay *interval*
Specifies how long the Replication Server waits after a read/write source fileset changes before it attempts to get a new copy of the fileset (MinRepDelay). The Replication Server tracks the currentness of replicas by maintaining a whole-fileset token for each fileset. If a Cache Manager changes the read/write fileset, the Replication Server relinquishes its whole-fileset token and waits for at least the time specified by MinRepDelay before requesting a new whole-fileset token. The default is 5 minutes or one quarter of the DefaultSiteAge, whichever is smaller. This value must be less than the MaxSiteAge specified for each replication site with the **-maxsiteage** option of the **fts addsite** command. *Applicable to Scheduled Replication only.*

-defaultsiteage *interval*
Specifies the default value to be used as the MaxSiteAge for a replication site (DefaultSiteAge). The DefaultSiteAge is used if the **-maxsiteage** option is omitted when the **fts addsite** command is used to add a replication site. The default is one quarter of the MaxAge. *Applicable to Scheduled Replication only.*

-clear Removes all replication parameters previously defined for the fileset. The options associated with the type of replication in use for the fileset can then be used to define new replication parameters, or they can all be omitted to allow the system to calculate new replication parameters for the fileset.

-cell*cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts setrepinfo** command is used to set or change the replication type and parameters for a read/write DCE LFS fileset. It affects the parameters for both Release and Scheduled Replication. It must be issued before replication sites can be defined for the fileset with the **fts addsite** command and before the **fts release** or **fts update** command can be used to copy replicas to the replication sites. The replication type and parameters for a fileset are stored in the fileset's entry in the Fileset Location Database (FLDB).

Use the following guidelines when deciding which type of replication (Release or Scheduled) to use with a read/write fileset:

- Use Release Replication if the fileset seldom changes or if the distribution of replicas must be tracked closely.
- Use Scheduled Replication if having the system release replicas of the fileset at regular intervals is preferred and the distribution of replicas does not need to be tracked.

When initially defining a fileset's replication type, include either the **-release** or **-scheduled** option. These options are then omitted from the command unless the replication type for the fileset is being changed (from Release to Scheduled, or from Scheduled to Release). To change the replication type, use the appropriate option (**-release** or **-scheduled**) to specify the new type, and include the **-change** option to indicate that the type is to be changed.

Note that, because Release Replication does not require a replication site to have a MaxSiteAge, it is likely that one or more Release Replication sites will have a MaxSiteAge of **0** (zero), which is the default value recorded for a site if no MaxSiteAge or DefaultSiteAge is specified. When changing from Release Replication to Scheduled Replication, the **-defaultsiteage** option *must* be used to set a DefaultSiteAge if any replication site does not have a MaxSiteAge and no DefaultSiteAge exists for the source fileset; otherwise, the **fts setrepinfo** command fails. If the command fails for this reason, reissue it, specifying a DefaultSiteAge with the **-defaultsiteage** option.

The **-maxage**, **-failage**, **-reclaimwait**, **-minrepdelay**, and **-defaultsiteage** options are used to set the corresponding replication parameters for a read/write fileset. (See the section on options for information on the replication parameter each option sets.) The following table lists each option's default value and describes the dependencies between the different options when they are used to set the replication parameters for either Release or Scheduled Replication.

Parameter	**Default**	**Release Replication**	**Scheduled Replication**
-maxage	2 hours	*Required only if* **-failage** is specified.	*Required only if* one of the following is specified: **-failage**, **-minrepdelay**, or **-defaultsiteage**.
-failage	The larger of 1 day or twice **-maxage**	*Optional*. If it is specified, the following are required: **-maxage** and **-reclaimwait**.	*Required only if* one of the following is specified: **-minrepdelay** or **-defaultsiteage**.

-reclaimwait	18 hours	*Required only if* **-failage** is specified.	*Required only if* one of the following is specified: **-failage**, **-minrepdelay**, or **-defaultsiteage**.
-minrepdelay	The smaller of 5 minutes or one quarter of **-defaultsiteage**	*Not applicable.*	*Required only if* one of the following is specified: **-failage** or **-defaultsiteage**.
-defaultsiteage	One-quarter of **-maxage**	*Not applicable.*	*Optional*. But if the other options are specified and **-defaultsiteage** is not, the **-maxsiteage** option of the **fts addsite** command is required when defining replication sites for the fileset.

The **fts** program calculates default values for each of the parameters *unless*

- The **-failage** option is specified for Release Replication.
- The **-failage**, **-minrepdelay**, or **-defaultsiteage** option is specified for Scheduled Replication.

Once one of these options is specified, the **fts** program no longer performs any default calculations; *interval* must be provided for all applicable options. (The exception is the **-defaultsiteage** option for Scheduled Replication, which is always optional.)

Also, because the **-minrepdelay** and **-defaultsiteage** options do not apply to Release Replication, they are recorded if specified but they are ignored.

Enter *interval* values as integers, using the following abbreviations to indicate units: **d** for days, **h** for hours, **m** for minutes, and **s** for seconds. The syntax for an *interval* is

[*integer* **d**] [*integer* **h**] [*integer*
m] [*integer*
s]

At least one of the four values (days, hours, minutes, or seconds) must be provided, and a unit abbreviation (**d**, **h**, **m**, or **s**) must be used with any integer. The unit abbreviations can be uppercase or lowercase, and they can be specified in any order. Examples of valid *interval* values are

3d2H
3M2h
1d6h30m45s

To change the replication parameters defined for a fileset, use the options for the parameters you want to change. To change *all* replication parameters associated with a fileset, use the **-clear** option to remove all replication parameters previously defined for the fileset, and use the options for the parameters you want to change to indicate the new parameters. To have the system calculate default values for all replication parameters, use only the **-clear** option.

Use the **fts lsfldb** or **fts lsft** command to display the replication parameters associated with a read/write fileset. Use the **fts lsreplicas** command to display the statuses of replicas at replication sites. Use the **fts statrepserver** command to display the status of the Replication Server on a File Server machine.

Note that replication is available in a cell only if the following conditions have been met: **root.dfs**, the cell's main read/write fileset, is a DCE LFS fileset; **root.dfs** was mounted with an explicit read/write mount point as a subdirectory of itself (the **root.dfs** fileset) when the cell was configured; and **root.dfs** is replicated. See Part 1 of this manual for information about configuring **root.dfs** to support replication.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine on which a version of the fileset resides. The issuer must also be listed in the **admin.ft** file on the machine on which the read/write

fileset resides if the following are true: The fileset's replication type is being changed from Release Replication to Scheduled Replication, and a replica actually resides at the replication site on the same File Server machine as the read/write fileset. (The first replication site defined for a fileset that uses Release Replication must be on the same File Server machine as the read/write fileset.)

Cautions

When using the **fts setrepinfo** command to set replication parameters, it is recommended that the default parameters (with the exception of MaxAge) be used for both types of replication. The dependencies between the parameters are complicated and should be defined by the issuer only when absolutely necessary.

Examples

The following command sets the initial Release Replication type and parameters for the read/write fileset named **rs_aix32.bin**. The default replication parameters are used for the fileset.

```
$ fts setrepinfo -fileset rs_aix32.bin -release
```

The following command changes the replication type for the **rs_aix32.bin** fileset from Release to Scheduled. It also clears the current replication parameters for the fileset and allows the system to calculate default values for all of the parameters.

```
$ fts setr -fileset rs_aix32.bin -scheduled -change -clear
```

The following command clears the current replication parameters used for the **rs_aix32.bin** fileset, replacing them with parameters specified by the issuer of the command:

```
$ fts setr rs_aix32.bin -maxage 6h -failage 12h -reclaimwait 1d \
-minrepdelay 15m -clear
```

The previous command changes the default Scheduled Replication parameters as follows:

- It increases the MaxAge from the default of 2 hours to 6 hours.
- It decreases the FailAge from the default of the larger of 1 day or twice the MaxAge to 12 hours (twice the MaxAge).
- It increases the MinRepDelay from the default of 5 minutes or one quarter of the DefaultSiteAge to 15 minutes.
- It increases the ReclaimWait from the default of 18 hours to 1 day.

Because the **-defaultsiteage** option is omitted from the command, the **-maxsiteage** option must be used when defining replication sites for the fileset with the **fts addsite** command.

Related Information

Commands: **fts addsite(8dfs)**, **fts lsfldb(8dfs)**, **fts lsft(8dfs)**, **fts lsreplicas(8dfs)**, **fts release(8dfs)**, **fts rmsite(8dfs)**, **fts statrepserver(8dfs)**, **fts update(8dfs)**.

fts statftserver

Purpose

fts statftserver – Reports on the activity of a Fileset Server

Synopsis

fts statftserver -server *machine* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server *machine*
Names the File Server machine about whose Fileset Server information is to be reported. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts statftserver** command reports on the actions of the Fileset Server (**ftserver** process) on the File Server machine specified with the **-server** option. The command returns information about the actions of the Fileset Server at the moment it is issued. This command is useful mainly if there is concern that a Fileset Server is not performing requested actions.

If no transactions are active on the specified machine, the command displays a message to that effect. This indicates that the Fileset Server is functioning properly. If transactions are active on the machine, the command displays information about the action currently being performed by the Fileset Server. Depending on the information displayed, the Fileset Server may or may not be functioning properly.

Output

If the Fileset Server is not currently performing any actions, the command displays the following message, indicating that the Fileset Server is functioning normally:

```
No active transactions on machine_name
```

If the Fileset Server is currently performing an action, the command displays information about the actions of the Fileset Server. The output includes fields containing ID numbers and flags that the Fileset Server sets for internal use. The details of the information returned by the command are more useful to programmers than to system administrators. A full understanding of the output requires familiarity with the code for the Fileset Server.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine specified by **-server**.

Related Information

Commands: **ftserver(8dfs)**.

fts statrepserver

Purpose

fts statrepserver – Displays the status of a Replication Server

Synopsis

fts statrepserver -server *machine* [**-long**][**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server*machine*
Names the File Server machine about whose Replication Server status information is to be displayed. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-long Specifies that more detailed information about the Replication Server is to be displayed. The additional output includes information about each replica managed by the Replication Server on the specified machine.

-cell*cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

-help Prints the online help for this command. All other options specified with this option are ignored.

Description

The **fts statrepserver** command displays information about the status of the Replication Server (**repserver** process) on the File Server machine specified with the **-server** option. Include the **-long** option to specify more detailed information about the Replication Server on the specified machine, as well as information about each replica managed by the Replication Server. Use the **fts lsreplicas** command to check the status of each replica of a fileset.

Privilege Required

No privileges are required.

Related Information

Commands: **fts lsreplicas(8dfs)**, **repserver(8dfs)**.

fts syncfldb

Purpose

fts syncfldb – Synchronizes FLDB entries to match their fileset headers

Synopsis

fts syncfldb -server *machine* [**-aggregate** *name*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server*machine*
Names the File Server machine from which to compare filesets to entries in the Fileset Location Database (FLDB). Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate*name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate or partition on **-server** for which to compare filesets to FLDB entries. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or partition in the *dcelocal* **/var/dfs/dfstab** file. Do not use this option under normal circumstances; omitting it allows synchronization of all filesets on **-server**. Use it only when just a single aggregate needs to be synchronized.

-cell*cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database).

You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts syncfldb** command inspects the fileset header of each online fileset that resides on a specified File Server machine (or, optionally, a specified aggregate or partition on that File Server machine). The command then checks that each FLDB entry is consistent with its fileset header. If the command encounters an inconsistency between a fileset header and its FLDB entry, the FLDB entry is changed to reflect the information in the fileset header. If the command encounters an FLDB entry without a corresponding fileset header, it deletes the FLDB entry; if the command encounters a fileset header without a corresponding FLDB entry, it creates an FLDB entry for that fileset.

The **fts syncfldb** command also performs the following additional functions:

- If it finds a backup fileset whose read/write source no longer exists at the same site, it displays a warning message.
- If it finds a fileset ID number that is larger than the value of the counter used by the FL Server when allocating fileset ID numbers, it records this ID number as the new value of the counter. The next fileset to be created receives a fileset ID number one greater than this number.
- If necessary, it increments or decrements the number of fileset entries recorded as residing on a File Server machine in the FLDB entry for the server.

The **fts syncfldb** command checks either all of the fileset headers on the File Server machine specified with the **-server** option or only the filesets on the optional partition or aggregate specified with the **-aggregate** option. The command checks a fileset header only if the fileset is marked as being **On-line**. If the command encounters a busy fileset on an aggregate, it exits without checking any other filesets. (A busy fileset is one upon which a fileset-related operation such as a move, clone, or release is currently being performed.)

It is recommended that the **fts syncfldb** command be run on all File Server machines in a cell *before* the **fts syncserv** command is run on the File Server machines in the cell. However, nothing prohibits the commands from being executed in the reverse order or independently of each other.

Note that the **fts syncfldb** and **fts syncserv** commands cannot restore replication information lost when the entry for a DCE LFS fileset is removed from the FLDB. Replication information must be reconstructed with the **fts setrepinfo** and **fts addsite** commands.

Because non-LFS filesets do not have fileset headers, the **fts syncfldb** and **fts syncserv** commands have limited effectiveness on non-LFS filesets. For example, because non-LFS filesets do not have fileset headers, the **fts syncfldb** command cannot determine the name of a non-LFS fileset that has no FLDB entry. If the command determines that it needs to create an FLDB entry for a non-LFS fileset, it generates a name of the form **SYNCFLDB-ADDED-***number*, where *number* is a unique number appended to the name to differentiate it from other names of the same type. The **fts rename** command then needs to be used to rename the fileset to its original name.

Privilege Required

The issuer must be listed in the **admin.ft** file on each machine that houses a version of any fileset stored at the specified site (**-server** and optionally **-aggregate**). The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine that houses a version of any fileset stored at the specified site.

Cautions

The physical disk on which a fileset resides cannot be moved from a machine in one cell to a machine in another cell with the expectation of simply running the **fts syncfldb** command to create an FLDB entry for the fileset in the new cell. Any attempt to introduce a fileset from one cell into another cell risks a fileset ID conflict between the newly introduced fileset and a fileset within the new cell that has the same fileset ID. This conflict causes one of the two conflicting filesets to be inaccessible.

Related Information

Commands: **fts addsite(8dfs)**, **fts rename**, **fts setrepinfo(8dfs)**, **fts syncserv(8dfs)**.

Files: **dfstab(4dfs)**.

fts syncserv

Purpose

fts syncserv – Synchronizes fileset headers to match their FLDB entries

Synopsis

fts syncserv -server *machine* [**-aggregate** *name*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server*machine*

Names the File Server machine for which to check entries in the Fileset Location Database (FLDB). Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate*name*

Specifies the device name, aggregate name, or aggregate ID of the aggregate or partition on **-server** for which to check FLDB entries. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or partition in the *dcelocal* **/var/dfs/dfstab** file. Do not use this option under normal circumstances; omitting it allows synchronization of all filesets on **-server**. Use it only when just a single aggregate needs to be synchronized.

-cell*cellname*

Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database).

You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts syncserv** command inspects the FLDB entry of each fileset on a specified File Server machine (or, optionally, a specified aggregate or partition on that File Server machine). The command then checks that each fileset header is consistent with its FLDB entry. If the command finds an inconsistency between the fileset name found in the fileset header and the name found in the FLDB entry, the fileset header is renamed to reflect the name in the FLDB entry. If the command encounters a fileset marked as **Off-line**, but the fileset's FLDB entry lists it as being **valid**, the command places the fileset online.

The **fts syncserv** command checks either all of the filesets on the File Server machine specified with the **-server** option or only the filesets on the optionally specified partition or aggregate specified with the **-aggregate** option. The command also checks the reported sites of all copies of an inspected fileset (even though that requires checking filesets on server machines other than -**server**).

It is recommended that the **fts syncfldb** command be run on all File Server machines in a cell *before* the **fts syncserv** command is run on the File Server machines in the cell. However, nothing prohibits the commands from being executed in the reverse order or independently of each other.

Note that the **fts syncserv** and **fts syncfldb** commands cannot restore replication information lost when the entry for a DCE LFS fileset is removed from the FLDB. Replication information must be reconstructed with the **fts setrepinfo** and **fts addsite** commands.

Because non-LFS filesets do not have fileset headers, the **fts syncserv** and **fts syncfldb** commands have limited effectiveness on non-LFS filesets. For example, because the **fts syncserv** command cannot destroy a disk partition, it cannot delete a non-LFS fileset, even if it determines that the fileset needs to be deleted. Instead, the **fts** program displays a warning message reporting the non-LFS fileset that needs to be deleted to

restore file system consistency. The proper commands then need to be used to delete the fileset.

Privilege Required

The issuer must be listed in the **admin.ft** file on each machine that houses a version of any fileset stored at the specified site (**-server** and optionally **-aggregate**). The issuer must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine that houses a version of any fileset stored at the specified site.

Examples

The following command synchronizes the FLDB entries of filesets whose site definitions mention **fs3**, including any copies of the filesets not located on **fs3**:

```
$ fts syncserv /.../abc.com/hosts/fs3
```

Related Information

Commands: **fts addsite(8dfs)**, **fts setrepinfo(8dfs)**, **fts syncfldb(8dfs)**.

Files: **dfstab(4dfs)**.

fts unlock

Purpose

fts unlock – Unlocks an entry in the FLDB

Synopsis

fts unlock -fileset {*name* | *ID*} [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}
Specifies the complete name or fileset ID number of the fileset whose entry in the Fileset Location Database (FLDB) is to be unlocked.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts unlock** command releases the lock on the FLDB entry for the fileset indicated by **-fileset**. Use the **fts unlockfldb** command to unlock multiple filesets at one time.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine on which a version of the fileset to be unlocked resides.

Cautions

Do not issue this command under normal circumstances. It is useful only if the FLDB entry for a fileset is locked but there is no reason to suspect inconsistency within the fileset or between it and the FLDB. Note that it is possible to list information from locked FLDB entries, even though they cannot be manipulated in other ways.

Examples

The following command unlocks the FLDB entry for the fileset named *user.terry*:

```
$ fts unlock user.terry
```

Related Information

Commands: **fts lock(8dfs)**, **fts unlockfldb(8dfs)**.

fts unlockfldb

Purpose

fts unlockfldb – Unlocks all specified locked entries in the FLDB

Synopsis

fts unlockfldb [**-server** *machine*] [**-aggregate** *name*] [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-server*machine*
: Names the File Server machine whose filesets are to have their Fileset Location Database (FLDB) entries unlocked. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address. Use this option with **-aggregate** to unlock the entries for the filesets on a specific aggregate on **-server**. Omit both this option and **-aggregate** to unlock all of the entries in the FLDB.

-aggregate *name*
: Specifies the device name, aggregate name, or aggregate ID of an aggregate or partition on **-server** on which the filesets whose FLDB entries are to be unlocked reside. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate or partition in the *dcelocal*/**var/dfs/dfstab** file. The **-server** option must be specified with this option. Omit both this option and **-server** to unlock all of the entries in the FLDB.

-cell *cellname*
: Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth
: Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth
: Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a

machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts unlockfldb** command releases the locks on the FLDB entries indicated by the combination of options specified. To unlock

- All entries in the FLDB, specify no options.
- All entries that mention a File Server machine in a site definition, specify the name of the File Server machine with **-server**.
- All entries that mention a specific site, specify both **-server** and **-aggregate**.
- A single fileset, use the **fts unlock** command instead.

Privilege Required

The issuer must be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine that houses a version of any fileset to be unlocked.

Cautions

Do not issue this command under normal circumstances. It is useful only if FLDB entries for filesets at a certain site are locked, but there is no reason to suspect inconsistency within the filesets or between the filesets and the FLDB. Note that it is possible to list information from locked FLDB entries, even though they cannot be manipulated in other ways.

Examples

The following command unlocks all locked entries in the FLDB:

```
$ fts unlockfldb
```

Related Information

Commands: **fts lock(8dfs)**, **fts unlock(8dfs)**.

Files: **dfstab(4dfs)**.

fts update

Purpose

fts update – Requests an immediate update of replicas of a read/write DCE LFS fileset that uses Scheduled Replication

Synopsis

fts update -fileset {*name* | *ID*} {**-all** | **-server** *machine*} [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**-verbose**][**-help**]

Options

-fileset {*name* | *ID*}
Specifies the complete name or fileset ID number of the read/write fileset whose replicas are to be updated immediately. For a fileset that uses Scheduled Replication, the command updates the indicated replicas to match the read/write version of the fileset. For a fileset that uses Release Replication, the command updates the replicas to match the read-only version that resides at the same site as the read/write version of the fileset.

-all Specifies that all replicas of the fileset indicated with the **-fileset** option are to be updated. Use this option or use the **-server** option.

-server *machine*
Names a specific File Server machine on which the replica of the fileset indicated with the **-fileset** option is to be updated. Only the replica on the specified File Server machine is updated. Specify the machine's DCE pathname, its host name, or its IP address. Use this option or use the **-all** option.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **-localauth** option.

-localauth Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts update** command asks the Replication Server to make an immediate update of replicas of the read/write DCE LFS fileset specified with the **-fileset** option. The effect of the command depends on whether the fileset to be updated uses Scheduled or Release Replication, as follows:

- *For a fileset that uses Scheduled Replication*, the command directs the Replication Servers on the indicated File Server machines to perform an immediate update based on the read/write version of the fileset. The Replication Servers ignore the value of the MinRepDelay parameter associated with the fileset; they immediately begin updating the replicas to match the version of the read/write fileset that exists at the time the command is issued.
- *For a fileset that uses Release Replication*, the command directs the Replication Servers on the indicated File Server machines to perform an immediate update based on the read-only replica that is stored on the same File Server machine as the read/write fileset (the replica that was created when the **fts release** command was last used for the fileset). The command does *not* first update the read-only replica on the read/write fileset's File Server machine. Because the MinRepDelay parameter does not apply to a fileset that uses Release Replication, the replicas should already be updated to match the replica on the read/write fileset's machine; the command should have no effect.

To indicate the replicas of the specified fileset that are to be updated, use the command's **-all** or **-server** option as follows:

- To update all replicas of the specified fileset, use the **-all** option.

- To update the replica stored on a specific File Server machine, identity the machine with the **-server** option.

Note that, as with releasing a new version of a fileset that uses Release Replication, forcing an update of a fileset that uses Scheduled Replication does not ensure immediate access to data in the new version of the replica. A Cache Manager continues to provide data cached from the old version of the replica until the MaxAge for the fileset expires or until the Cache Manager needs to access data from the replica that it has not already cached.

To gain immediate access to data in the new version of the replica, issue the **cm flush** or **cm flushfileset** command to flush the old data from the cache. This forces the Cache Manager to replace data it has cached from the replica. Replication Servers begin replication in parallel; however, until all replicas have been updated, you cannot directly force the Cache Manager to access data from the new version of the replica.

The **fts update** command does not change the replication type and parameters defined for the specified fileset. Before the **fts update** command can be used, the **fts setrepinfo** command must be used to define the replication parameters for the read/write fileset. The **fts addsite** command must also be used to define at least one replication site for the read/write fileset.

Use the **fts lsreplicas** command to check the status of replicas of the fileset. Use the **fts statrepserver** command to check the status of the Replication Server on a File Server machine.

Privilege Required

No privileges are required.

Examples

The following command requests an immediate update of the replica of the read/write fileset named **pmax_osf1.bin** at the replication site defined on the File Server machine named **fs3**:

```
$ fts update pmax_osf1.bin /.../abc.com/hosts/fs3
```

Related Information

Commands: **cm flush(8dfs)**, **cm flushfileset(8dfs)**, **fts addsite(8dfs)**, **fts lsreplicas(8dfs)**, **fts release(8dfs)**, **fts setrepinfo(8dfs)**, **fts statrepserver(8dfs).**

fts zap

Purpose

fts zap – Removes a DCE LFS fileset from its site without updating the FLDB

Synopsis

fts zap -ftid *ID* **-server** *machine* **-aggregate** *name* [**-cell** *cellname*] [{**-noauth** | **-localauth** }] [**- verbose**][**-help**]

Options

-ftid *ID*
Specifies the fileset ID number of the fileset to remove; a fileset name is not a valid argument.

-server*machine*
Names the File Server machine from which to remove the fileset. Specify the File Server machine using the machine's DCE pathname, the machine's host name, or the machine's IP address.

-aggregate *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate on **-server** from which to remove the fileset. These identifiers are specified in the first, second, and fourth fields of the entry for the aggregate in the *dcelocal* **/var/dfs/dfstab** file.

-cell *cellname*
Specifies the cell where the command is to be run. The default is the local cell of the issuer of the command.

-noauth
Directs **fts** to use the unprivileged identity **nobody** as the identity of the issuer of the command. If you use this option, do not use the **- localauth** option.

-localauth
Directs **fts** to use the DFS server principal name of the machine on which the command is issued as the identity of the issuer. Use this option only if the command is issued from a DFS server machine (a machine that has a DFS server principal in the local Registry Database). You must be logged into the server machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

-verbose Directs **fts** to provide detailed information about its actions as it executes the command.

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

Description

The **fts zap** command removes the DCE LFS fileset with the fileset ID number specified by **-ftid** from the site defined by **- server** and **-aggregate**. It neither changes the corresponding Fileset Location Database (FLDB) entry for the fileset nor decrements the number of fileset entries recorded in the server entry in the FLDB for the specified File Server machine.

This command is useful in situations in which it is important to delete a fileset but, for some reason, the FLDB is unreachable (for example, the FL Server is unavailable). The issuer can remove information about the deleted fileset from the FLDB by running the **fts syncserv** and **fts syncfldb** commands. The issuer can also reconcile the FLDB with the **fts rmsite** command (which removes site information about a read-only version from a fileset's FLDB entry), the **fts delete** command (which removes site information about the read/write or backup version from a fileset's FLDB entry), or the **fts delfldbentry** command (which removes the entire entry for a fileset from the FLDB). (The **fts zap** command can also be used to remove normally temporary "clone" filesets that can sometimes be left after an interrupted **fts move** operation.)

If the DCE LFS fileset to be removed is also mounted locally (as a file system on its File Server machine), you must remove its local mount point before you delete it. The **fts zap** command cannot be used to delete a fileset that is mounted locally.

Privilege Required

The issuer must be listed in the **admin.ft** file on the machine specified by **-server**.

Cautions

Do not use this command as the standard way to remove a fileset. It can make the FLDB inconsistent with the filesets on File Server machines. Use the **fts delete** command to remove the fileset entry from the FLDB at the same time that the fileset is removed.

Examples

The following command removes the fileset with fileset ID **0,,36870988** from **/dev/lv01** on **fs6**, without recording the change in the FLDB:

```
$ fts zap 0,,36870988 /.../abc.com/hosts/fs6 /dev/lv01
```

Related Information

Commands: **fts delete(8dfs)**, **fts delfldbentry(8dfs)**, **fts rmsite(8dfs)**, **fts syncfldb(8dfs)**, **fts syncserv(8dfs)**.

Files: **dfstab(4dfs)**.

ftserver

Purpose

Initializes the Fileset Server

Synopsis

ftserver [**-adminlist** *filename*] [**-verbose**][**-help**]

Options

-adminlist *filename*
: Specifies the administrative list file that contains principals and groups authorized to execute **ftserver** RPCs (usually using **fts** commands). If this option is omitted, **ftserver** obtains the list of authorized users from the default administrative list file, *dcelocal*/**var/dfs/admin.ft**.

-verbose
: Directs the **ftserver** process to provide a detailed report on what it is doing by displaying messages on standard error.

-help
: Prints the online help for this command. All other valid options specified with this option are ignored.

  The **help** and **apropos** commands available with all command suites are also available with the **ftserver** command. See the **bos help** and **bos apropos** reference pages for examples of using these commands.

Description

The Fileset Server, or **ftserver** process, handles fileset administration operations, such as creating, deleting, moving, and replicating filesets. The **ftserver** process must be run on all machines that export data for use in the global namespace. A machine that runs the Fileset Server, the File Exporter (which is initialized by the **fxd** process), and the **dfsbind** process is considered a DFS File Server machine. The Fileset Server is usually started and controlled by the BOS Server; if it is not, execute the **ftserver** process as a background process. The binary file for the **ftserver** process resides in *dcelocal*/**bin/ftserver**.

The first time it is initialized, **ftserver** creates the *dcelocal*/**var/dfs/admin.ft** administrative list file if the file does not already exist. The principals and groups listed in the **admin.ft** administrative list are authorized to administer filesets on the machine. Because some operations, such as fileset moves, are accomplished by two Fileset Servers communicating, server principal names must also appear in the **admin.ft** list.

For simplified administration, create one **admin.ft** administrative list that contains the server principal names of all machines in the administrative domain. The same **admin.ft** list can then be used by all **ftserver** processes in the domain.

When it is started, **ftserver** creates the *dcelocal* **/var/dfs/adm/FtLog** event log file if the file does not already exist. It then appends messages to the file. If the file exists when the **ftserver** is started, the process moves it to the **FtLog.old** file in the same directory (overwriting the current **FtLog.old** file if it exists) before creating a new version to append messages to.

Use the **fts statftserver** command to check the status of the Fileset Server on any server machine.

Privilege Required

You must be logged in as **root** on the local machine.

Output

If problems are encountered during initialization, the **ftserver** process displays error messages on standard error output. The **ftserver** process keeps an event log in the file *dcelocal* **/var/dfs/adm/FtLog**.

If run with the **-verbose** option, the **ftserver** process provides a detailed report on what it is doing by displaying messages on standard error.

Related Information

Commands: **dfsbind(8dfs)**, **fts statftserver(8dfs)**, **fxd(8dfs)**.

Files: **admin.ft(4dfs)**, **FtLog(4dfs)**.

fxd

Purpose

fxd – Initializes the File Exporter and starts associated kernel daemons

Synopsis

fxd -admingroup *group* [**-mainprocs** *number_of_background_daemons*] [**-tokenprocs** *number_of_token_daemons*] [**-hostlife** *client_timeout*] [**-hostrpc** *client_rpc_timeout*] [**-pollinterval** *server_poll_period*] [**-maxlife** *max_hostlife*] [**-maxrpc** *max_hostrpc*] [**-notsr**][**-minlocalprotectlevel** *level*] [**-maxlocalprotectlevel** *level*] [**-minremoteprotectlevel** *level*] [**-maxremoteprotectlevel** *level*] [**-verbose**][**-help**]

Options

-admingroup *group*

Specifies the name of the group that can administer the File Exporter on this machine. Members of the specified group can effectively change the permissions, owner, and owning group of any file system object exported from the machine. A group from the local cell can be specified by a full or abbreviated group name (for example, **/.../***cellname /group_name* or just *group_name*); a group from a foreign cell can be specified only by a full group name. The **-admingroup** option performs a function similar to that of the administrative lists associated with DFS server processes, such as the Fileset Server and the Fileset Location Server, that run in the user space.

-mainprocs *number_of_background_daemons*

Specifies the number of main kernel processes (File Exporter kernel daemons) to run on the machine. These File Exporter kernel daemons are responsible for receiving and servicing RPC requests for data and tokens from DFS clients. Specify an integer greater than 0 (zero) to indicate the number of main kernel daemons to perform these services. If this option is omitted, four main kernel daemons perform these services.

-tokenprocs *number_of_token_daemons*

Specifies the number of token-revocation kernel processes (File Exporter kernel daemons) to run on the machine. These File Exporter kernel

daemons are responsible for responding to RPCs from DFS clients that are themselves responding to token revocation requests. Specify an integer greater than 0 (zero) to indicate the number of kernel daemons to perform these services. If this option is omitted, two kernel daemons perform these services.

-hostlife *client_timeout*
Specifies the host lifetime the File Exporter assigns to each client that contacts it. The host lifetime is the length of time for which the File Exporter considers a client to be alive. Each client must contact the File Exporter within this amount of time to renew its host lifetime. As long as a client's host lifetime has not expired, the File Exporter cannot revoke its tokens without its permission.

By default, the File Exporter assigns each client a host lifetime of 2 minutes. Specify an integer to indicate a number of seconds to serve as the host lifetime. The host lifetime must be greater than 0 (zero) and less than or equal to the maximum host lifetime (specified with the **-maxlife** option) and the host RPC lifetime (specified with the **-hostrpc** option).

-hostrpc *client_rpc_timeout*
Specifies the host RPC lifetime the File Exporter assigns to each client that contacts it. The host RPC lifetime is the length of time for which the File Exporter guarantees to attempt an RPC to a client before it revokes its tokens. The File Exporter can revoke the tokens of any client whose host RPC lifetime has expired without contacting the client.

By default, the File Exporter assigns each client a host RPC lifetime of 2 minutes. Specify an integer to indicate a number of seconds to serve as the host RPC lifetime. The host RPC lifetime must be greater than or equal to the host lifetime (specified with the **-hostlife** option) and less than or equal to the maximum host RPC lifetime (specified with the **-maxrpc** option).

-pollinterval *server_poll_period*
Specifies the polling interval the File Exporter assigns to each client that contacts it. The polling interval is the frequency with which each client that has tokens from the File Exporter is to poll it in the event that it cannot be reached. Each client sends an RPC to the File Exporter with this frequency until it can again contact it.

By default, the File Exporter assigns each client a polling interval of 3 minutes. Specify an integer greater than 0 (zero) to indicate a number of seconds to serve as the polling interval.

-maxlife *max_hostlife*

Specifies the maximum host lifetime the File Exporter can grant a client. A client can request a host lifetime larger than that specified with the **-hostlife** option, but the File Exporter will not grant a host lifetime greater than the value specified with this option.

By default, the File Exporter uses a value of 2 minutes as the maximum host lifetime. Specify an integer to indicate a number of seconds to serve as the maximum host lifetime. The maximum host lifetime must be greater than or equal to the host lifetime (specified with the **-hostlife** option) and less than or equal to the maximum host RPC lifetime (specified with the **-maxrpc** option).

-maxrpc *max_hostrpc*

Specifies the maximum host RPC lifetime the File Exporter can grant a client. A client can ask for a host RPC lifetime larger than that specified with the **-hostrpc** option, but the File Exporter will not grant a host RPC lifetime greater than the value specified with this option.

By default, the File Exporter uses a value of 2 minutes as the maximum host RPC lifetime. Specify an integer to indicate a number of seconds to serve as the maximum host RPC lifetime. The maximum host RPC lifetime must be greater than or equal to the host RPC lifetime (specified with the **-hostrpc** option) and the maximum host lifetime (specified with the **-maxlife** option).

-notsr Specifies that the File Exporter is to forego token state recovery when it is restarted. If this option is specified, the File Exporter accepts requests for new tokens as soon as it can again be contacted by clients. By default, it provides a brief token state recovery period during which it accepts requests only to reestablish tokens from clients that held them before it was restarted. (It bases the duration of its period of token state recovery on the greater of its **-pollinterval** or **-maxlife**, plus 20 seconds.)

This option is useful primarily for debugging purposes. Use it sparingly, as it prevents the File Exporter from maintaining consistent token state across File Server machine restarts.

-minlocalprotectlevel *level*
Specifies the minimum acceptable DCE RPC authentication level for communications between the File Exporter and clients within the same cell. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Security subsection in the Description section.

-maxlocalprotectlevel *level*
Specifies the maximum acceptable DCE RPC authentication level for communications between the File Exporter and clients in the local cell. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Security subsection in the Description section.

-minremoteprotectlevel *level*
Specifies the minimum acceptable DCE RPC authentication level for communications between the File Exporter and clients in foreign cells. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Security subsection in the Description section.

-maxremoteprotectlevel *level*
Specifies the maximum acceptable DCE RPC authentication level for communications between the File Exporter and clients in foreign cells. The *level* is set either as an integer value between 0 and 6, the complete string defining the authentication level, or an abbreviation of that string. For a description of the various DCE RPC levels, see the Security subsection in the Description section.

-verbose Directs **fxd** to produce more detailed information about its actions during initialization and as it creates kernel daemons.

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

The **help** and **apropos** commands available with all command suites are also available with the **fxd** command. See the **bos help** and **bos apropos** reference pages for examples using these commands.

Description

The **fxd** command initializes the File Exporter on a File Server machine and starts all kernel daemons, such as those for garbage collection, required by the File Exporter. During initialization, it also passes the File Exporter such information as the name of the local cell, information about the local Fileset Database machines, and the identity of the group that can administer the File Exporter. The File Exporter uses this information to communicate with other processes, such as the Fileset Location (FL) Servers on Fileset Database machines, and to ensure that only privileged users administer data in filesets exported from the machine.

The File Exporter must be run on all machines that export data for use in the global namespace. A machine that runs the File Exporter, the Fileset Server (**ftserver** process), and the **dfsbind** process is considered to be a DFS File Server machine. The File Exporter is typically run by adding the **fxd** command to the proper start-up file (**/etc/rc** or its equivalent). The **dfsbind** process must be run before the **fxd** process in a start-up file. The binary file for the **fxd** process resides in *dcelocal*/**bin/fxd**. The process automatically places itself in the background once its initialization is complete.

The **-mainprocs** and **-tokenprocs** options can be used to alter the default number of kernel daemons running on the server machine, as follows:

- The **-mainprocs** option specifies the number of main kernel daemons that run on the machine to service RPC requests for data and tokens from DFS clients. The default number of main kernel daemons is four, which is usually sufficient to handle RPC requests from many DFS client machines. Use the **-mainprocs** option to increase the number of main kernel daemons dedicated to servicing RPC requests if the machine is to support a large number of DFS clients.
- The **-tokenprocs** option specifies the number of kernel daemons dedicated to responding to RPCs from DFS clients that are themselves responding to token revocation requests from the File Exporter. The default number of kernel daemons dedicated to this task is two. If the **-mainprocs** option is used to increase the number of main kernel daemons, use the **-tokenprocs** option to increase the number of kernel daemons dedicated to handling responses to token revocation requests accordingly.

On most system types, these kernel daemons appear as nameless entries in the output of the **ps** command (or its equivalent). However, because some of the kernel daemons run as threads rather than processes, not all of them show up in the output of the **ps** command.

The **-admingroup** option is used to associate system administrators with the **fxd** process. Members of the group specified with the **-admingroup** option have the necessary ACL and UNIX permissions to change the permissions of any file or directory object exported from the machine. They have the equivalent of the ACL **c** permission on the objects in each exported DCE LFS fileset, and they can effectively change the mode bits on the objects in each exported non-LFS fileset. (To change the permissions on an object that resides in a lower-level directory of an exported fileset, a member of the group may need to provide the group with the necessary permissions on directories in the path that leads to the object.) Members of the group can also change the owner and owning group of any object exported from the machine. Note that, while similar in many respects, inclusion in the group specified with the **-admingroup** option and being logged in as **root** are *not* equivalent.

Place only highly trusted users in the group associated with the **fxd** process. Members of the group generally constitute a subset of the users in other DFS administrative lists such as the **admin.bos** file. For simplified administration, the same group can be specified with the **-admingroup** options of all **fxd** commands issued in a domain.

The **fxd** command includes a number of options that affect the File Exporter's management of tokens. The following two sections describe only those token-related issues germane to the **fxd** command's options. Tokens, their management by the File Exporter, and their benefits and implications are described in Part 1 of this manual.

Token Management

Token management refers to the File Exporter's use of tokens to synchronize access to data and metadata on a File Server machine. The File Exporter uses tokens to track the clients that have accessed data from the machine and the types of operations they are permitted to perform on the data. When a client wants to access or change data on a File Server machine, it contacts the File Exporter on the machine to request the necessary tokens for the data. If the File Exporter can grant the client the requested tokens, the client in turn can use the tokens to access the data from the File Exporter.

Many factors affect the File Exporter's ability to grant a client's request for tokens. The File Exporter can always grant requests for tokens that do not conflict with those already held by another client. If requested tokens do conflict with existing tokens held by another client, the File Exporter tries to revoke the existing tokens. If it can revoke the existing tokens, it grants those requested; if it cannot, it either places the request in a queue or refuses it. (The choice is the client's.)

When its tokens are revoked, a client such as the Cache Manager flushes cached data for which the tokens applied, writing any modified data back to the File Server machine. Among the factors that affect the File Exporter's ability to revoke existing

tokens are the various lifetimes it associates with the tokens it grants and the clients to which it grants them. The following list briefly introduces these values, of which the latter two can be modified with options of the **fxd** command:

Token lifetime
: Specifies the length of time for which a token is valid. The File Exporter needs to revoke only valid tokens. Once a token has expired, the File Exporter does not need to revoke it; it can simply grant new tokens as if the expired token did not exist.

Host lifetime
: Specifies the length of time for which the File Exporter considers a client to be alive. A client must contact the File Exporter to renew its host lifetime before it expires. As long as a client's host lifetime has not expired, the File Exporter cannot revoke its tokens without its permission.

Host RPC lifetime
: Specifies the length of time for which the File Exporter agrees to attempt to make an RPC to a client before it revokes its tokens. The client's response to the RPC renews its host lifetime, meaning the File Exporter cannot revoke its tokens without its permission. If the client fails to respond to the RPC but its host lifetime has not expired, the File Exporter cannot revoke its tokens; if it fails to respond and its host lifetime has expired, the File Exporter can revoke any tokens it holds without contacting it further. The File Exporter can revoke a client's tokens without contacting it once its host RPC lifetime has expired. A client's host RPC lifetime must be at least as long as its host lifetime.

In general, the following rules apply to the File Exporter's revocation of valid tokens:

1. If the client's host lifetime has not expired, the File Exporter tries to contact the client; it must have the client's permission to revoke its tokens.
2. If the client's host lifetime has expired but its host RPC lifetime has not, the File Exporter tries to contact the client one time. If the client responds, the File Exporter cannot revoke its tokens without its permission; otherwise, the File Exporter can revoke any tokens held by the client without contacting it further.
3. If the client's host RPC lifetime has expired, the File Exporter can revoke its tokens without contacting it.

The following options of the **fxd** command can be used to modify the lifetimes the File Exporter assigns to its clients. By default, the File Exporter use values of 2 minutes for each of these lifetimes.

-hostlife Specifies each client's default host lifetime. The **-hostlife** must be greater than 0 (zero) and less than or equal to both the **-maxlife** and the **-hostrpc**.

-maxlife Specifies the maximum host lifetime the File Exporter will grant to a client that asks for one larger than the default specified with the **-hostlife** option. The **-maxlife** must be greater than or equal to the **-hostlife** and less than or equal to the **-maxrpc**.

-hostrpc Specifies each client's default host RPC lifetime. The **-hostrpc** must be greater than or equal to the **-hostlife** and less then or equal to the **-maxrpc**.

-maxrpc Specifies the maximum host RPC lifetime the File Exporter will grant to a client that asks for one larger than the default specified with the **-hostrpc** option. The **-maxrpc** must be greater than or equal to both the **-maxlife** and the **-hostrpc**.

If you use one of these options to modify a default lifetime value, be careful not to violate any of the dependency rules described in the previous list. In some cases, the command can adjust values not modified by the user to ensure that the dependencies are not violated, as follows:

- If you increase the value of **-hostlife** without specifying **-maxlife**, **-hostrpc**, or **-maxrpc**, the command increases the other three values as necessary.
- If you increase the value of **-maxlife** without specifying **-maxrpc**, the command increases the value of **-maxrpc** as necessary.
- If you increase the value of **-hostrpc** without specifying **-maxrpc**, the command increases the value of **-maxrpc** as necessary.
- If you decrease the value of **-maxlife** without specifying **-hostlife**, the command decreases the value of **-hostlife** as necessary.
- If you decrease the value of **-maxrpc** without specifying **-hostrpc**, the command decreases the value of **-hostrpc** as necessary.
- If you specify multiple values that explicitly violate one or more of the dependency rules, the command fails.

- If you specify a value that implicitly violates one or more of the dependency rules and the command cannot adjust other values to compensate for the violation, the command fails.

The command displays an appropriate message if it adjusts a value that was not specified or if it fails because specified values violate the previously defined rules.

Token State Recovery

Token state recovery refers to clients regaining their tokens following a network failure or File Server machine restart. In either of these situations, each client that cannot contact the File Exporter polls the File Exporter at regular intervals. When it can again reach the File Exporter, the client attempts to recover tokens it had before it lost contact. The frequency with which each client tries to contact the File Exporter in these cases is defined with the **-pollinterval** option of the **fxd** command; by default, each client polls the File Exporter every 3 minutes.

In the case of a network failure, a client may be unable to prevent its host lifetime from expiring before it can again contact the File Exporter. Once communication is restored, the client must either reclaim its tokens or request new ones, as necessary. The client may need to compete for its tokens with other clients to which the tokens were granted while it could not reach the File Exporter.

In the case of a File Exporter restart, the File Exporter loses all knowledge of tokens it granted. For a brief period after it restarts, it refuses all requests for new tokens from all clients. During this period, it accepts requests only to reestablish tokens from those clients that held them before it was restarted. The File Exporter gives those clients that held tokens before it was restarted the chance to recover their tokens without having to compete with other clients that could request the same tokens.

The File Exporter bases the length of its period of token state recovery after a restart on the **-maxlife** or the **-pollinterval**, whichever is greater (it adds 20 seconds to the value it chooses to compensate for its own initialization time). The larger of these two values ensures that each client that had tokens has an opportunity to contact the File Exporter before the File Exporter accepts requests for new tokens from all clients. (Within this time, each client will contact the File Exporter either to renew its host lifetime or to poll the File Exporter.)

If the File Exporter receives many requests to reestablish tokens just prior to the end of its token state recovery period, it dynamically extends the original length of the period. If many clients continue to contact it during the extension, the File Exporter continues to extend the period incrementally, to a maximum of twice its original length.

(Note that, if a client is restarted for any reason, it loses all knowledge of the tokens it possessed prior to the restart; recovery of its tokens is not possible.)

Security

The **-minlocalprotectlevel**, **-maxlocalprotectlevel**, **-minremoteprotectlevel**, and **-maxremoteprotectlevel** options set the minimum and maximum RPC authentication bounds for communications between the File Exporter and clients. These bounds are used in negotiating an RPC authentication level for communications with clients. Two sets of bounds are maintained: a set that governs communications with clients within the same cell, and a second set that governs communications with clients in foreign cells.

In operation, the File Exporter and client (Cache Manager) interact to arrive at a mutually acceptable authentication level for communications. The negotiation starts with an RPC using the initial authentication level sent from the Cache Manager to the File Exporter. If the initial authentication level is outside the minimum or maximum bounds set through the **fxd** command, the File Exporter returns a response to the Cache Manager specifying that the authentication level is either too low or too high. The Cache Manager then decreases or increases its authentication level accordingly and retries the RPC. This process continues until the Cache Manager either adjusts its RPCs to an acceptable security level or the File Exporter requests a security level below the minimum set at the Cache Manager (causing the Cache Manager to refuse communications with the File Exporter). Once the Cache Manager and File Exporter have negotiated a security level, the Cache Manager stores this information so that it does not need to renegotiate this level for further communications with the File Exporter.

In addition, administrators can also set advisory bounds on a per-fileset basis. At present, these advisory levels serve only to bias the Cache Manager's selection of an initial authentication level (they may be enforced in a future version of DFS). Advisory bounds are set through the **fts setprotectlevels** command and are stored in the FLDB record for that fileset.

Note that the use of this command does not preclude communications with File Servers running earlier versions of DFS.

The various authentication levels are set by specifying either an integer value between 0 and 6, a complete string specifying the authentication level, or an abbreviation of that string as the *level* argument for the various command options. The following lists the various authentication levels:

- **rpc_protect_level_default** or **default** or **0**: Use the DCE default authentication level.

- **rpc_protect_level_none** or **none** or **1**: Perform no authentication.
- **rpc_protect_level_connect** or **connect** or **2**: Authenticate only when the Cache Manager establishes a connection with the File Server.
- **rpc_protect_level_call** or **call** or **3**: Authenticate only at the beginning of each RPC received.
- **rpc_protect_level_pkt** or **pkt** or **4**: Ensure that all data received is from the expected host.
- **rpc_protect_level_pkt_integrity** or **pkt_integrity** or **5**: Authenticate and verify that none of the data transferred has been modified.
- **rpc_protect_level_pkt_privacy** or **pkt_privacy** or **6**: Perform authentication as specified by all of the previous levels and also encrypt each RPC argument value.

Note that there is a trade-off between selecting higher security and performance. The higher levels of security require more overhead and increase the response time in file operations with File Servers.

The default values of the File Exporter and Cache Manager are such that, if they are not changed, the File Exporter and Cache Manager will negotiate to the packet integrity level. The default File Exporter values are as follows:

- The default minimum authentication level for communications with clients in the local cell is set to none.
- The default maximum authenticationn level for communications with clients in the local cell is set to packet privacy.
- The default minimum authentication level for communications with clients in foreign cells is set to none.
- The default maximum authentication level for communications with clients in foreign cells is set to packet privacy.

The default Cache Manager settings are as follows:

- The default initial authentication level for communications with File Exporters in the local cell is set to packet integrity.
- The default minimum authentication level for communications with File Exporters in the local cell is set to none.
- The default initial authentication level for communications with File Exporters in foreign cells is set to packet integrity.

- The default minimum authentication level for communications with File Exporters in foreign cells is set to packet.

Given that both Cache Manager default initial authentication levels are set to packet integrity and that this level is within the default bounds set at the File Exporter, the default authentication level is therefore packet integrity. If you set the minimum bound at the File Exporter higher than packet integrity, any Cache Managers from a version of DFS previous to 1.2.2 will not be able to communicate with that File Exporter.

Privilege Required

The issuer must be logged in as **root** on the local machine.

Cautions

If you restart the File Exporter with the **fxd** command's **-notsr** option, the File Exporter does not enter token state recovery; clients do not have a protected opportunity to reestablish their tokens after the restart. Similarly, if you restart the File Exporter using different values for the command's lifetime or polling interval values, the File Exporter may not remain in token state recovery long enough to provide all clients an opportunity to reestablish their tokens after it is restarted. (Until they reestablish contact with the File Exporter, clients continue to use the previous lifetime and polling interval values, which may be too long if the File Exporter is directed to use shorter values when it is restarted.)

If you set the minimum RPC authentication level for communications with clients in either local or foreign cells to higher than packet integrity, the affected clients that are running a version of DFS previous to 1.2.2 will not be able to communicate with the File Exporter.

Output

The command sends error messages to standard error output (**stderr**) if problems are encountered during initialization. It also displays error messages if you specify values for its lifetime-related options that violate the dependencies mentioned in the section on Token Management. Finally, it displays warning messages if it adjusts one or more of its lifetime values to compensate for an option you specify.

Examples

The following line, entered in the appropriate initialization file (**/etc/rc** or its equivalent) on a File Server machine, starts the **fxd** process on the local machine. The **cell_fileset** group is specified as the administrative group for the File Exporter on the machine. The **dfsbind** process must be run before the **fxd** process in a start-up file.

```
fxd -admin cell_fileset
```

The previous command line can be modified as follows to increase the host RPC lifetime, maximum host lifetime, and maximum host RPC lifetime associated with the File Exporter:

```
fxd -admin cell_fileset -hostrpc 180 -maxlife 240
```

These options change the File Exporter's lifetime values, as follows:

- The **-hostrpc** option explicitly increases the host RPC lifetime to 3 minutes.
- The **-maxlife** option explicitly increases the maximum host lifetime to 4 minutes. It also causes the command to implicitly increase the maximum host RPC lifetime to 4 minutes. (Note that, had the **-maxlife** option been omitted, the command would have implicitly increased the maximum host RPC lifetime to 3 minutes to match the increase to the host RPC lifetime.)

Related Information

Commands: **dfsbind(8dfs)**, **ftserver(8dfs) fts_setprotectlevels(8dfs)**.

growaggr

Purpose

growaggr – Increases the size of a DCE LFS aggregate

Synopsis

growaggr -aggregate *name* [**-aggrsize** *blocks*] [**-noaction**][**-help**]

Options

-aggregate *name*

Specifies the device name or aggregate name of the DCE LFS aggregate whose size is to be increased. These names are specified in the first and second fields of the entry for the aggregate in the *dcelocal* **/var/dfs/dfstab** file. The specified aggregate does not need to be exported, nor does any fileset on the aggregate need to be mounted locally or in the global namespace.

-aggrsize *blocks*

Specifies the total number of 1024-byte blocks to be available on the specified aggregate. The number of 1024-byte blocks specified with this option cannot exceed the total size of the disk partition on which the aggregate resides, and it must be at least three DCE LFS blocks greater than the current size of the aggregate. (The number of bytes in a DCE LFS block is defined on a per-aggregate basis with the **-blocksize** option of the **newaggr** command when an aggregate is created.)

Include the **-noaction** option with this option to determine if the specified aggregate size is valid without changing the current size of the aggregate. Omit both this option and the **-noaction** option to increase the size of the aggregate to the total size of the disk partition on which it resides.

-noaction Used without the **-aggrsize** option, this option directs the command to display the total number of 1024-byte blocks on the disk partition on which the specified aggregate resides. Used with the **-aggrsize** option, this option determines if the specified aggregate size is valid. The current size of the specified aggregate is not affected if this option is used.

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

The **help** and **apropos** commands available with all command suites are also available with the **growaggr** command. See the **bos help** and **bos apropos** reference pages for examples using these commands.

Description

The **growaggr** command is used to increase the size of an existing DCE LFS aggregate. The aggregate whose size is to be increased is specified with the **-aggregate** option. The binary file for the **growaggr** command resides in *dcelocal* **/bin/growaggr**.

The **-aggrsize** option is used to specify the total size to make the aggregate. Specify the size as a number of 1024-byte blocks. The size specified with this option cannot exceed the total size of the disk partition on which the aggregate resides. The specified size also must be at least three DCE LFS disk blocks greater than the current size of the aggregate. If it is not, the command displays the minimum size in 1024-byte blocks that can be specified. (The number of bytes in a DCE LFS block is defined on a per-aggregate basis with the **-blocksize** option of the **newaggr** command when an aggregate is initialized. It must be a power of 2 between 1024 and 65,536.)

If the **-noaction** option is included with the command, the present size of the aggregate is not affected. Combine the **-aggrsize** and **-noaction** options to achieve the following results:

- Specify only the **-aggrsize** option to increase the size of the aggregate to the specified size, as described previously.
- Specify only the **-noaction** option to determine the total number of 1024-byte blocks on the partition on which the aggregate resides.
- Specify both the **-aggrsize** and **-noaction** options to determine if the size specified with the **-aggrsize** option is valid (within the limits defined previously).
- Omit both the **-aggrsize** and **-noaction** options to increase the size of the aggregate to the total size of the disk partition on which it resides.

In operating systems that support logical volumes, the command is useful for increasing the size of an aggregate when the size of the logical volume on which the aggregate resides is increased. It can also be used to increase the size of an aggregate that was deliberately made smaller than the size of the partition or logical volume on which it resides.

The command does not affect any data or filesets that already reside on the aggregate to be grown.

Privilege Required

If the **-noaction** option is *not* included with the command, the issuer must have both the read and write permissions for the device (disk partition) on which the specified aggregate resides; if the **-noaction** option is included with the command, the issuer needs only the read permission for the device on which the aggregate resides. An issuer who is logged in as **root** on the machine on which the aggregate resides always has the necessary privilege to issue this command.

Related Information

Commands: **newaggr(8dfs)**.

Files: **dfstab(4dfs)**.

newaggr

Purpose

newaggr – Initializes a DCE LFS aggregate

Synopsis

newaggr -aggregate *name* **-blocksize** *bytes* **-fragsize** *bytes* [**-initialempty** *blocks*] [**-aggrsize** *blocks*] [**-logsize** *blocks*] [**-overwrite**][**-verbose**][**-noaction**][**-help**]

Options

-aggregate *name*

Specifies the device name or aggregate name of the disk partition to be initialized as a DCE LFS aggregate. These names are specified in the first and second fields of the entry for the aggregate in the *dcelocal* **/var/dfs/dfstab** file.

-blocksize *bytes*

Specifies the number of bytes to be available in DCE LFS blocks on the aggregate (also referred to as the blocking factor). The value provided must be a power of 2 between 1024 and 65,536.

The number controls how disks are addressed in DCE LFS. No file larger than 2^{31} blocks can be read or written. (Other considerations, chiefly I/O speed versus disk utilization, also constrain the maximum file size.)

-fragsize *bytes*

Specifies the number of bytes to be available in DCE LFS fragments on the aggregate. The value provided must be a power of 2 between 1024 and the number of bytes specified with **-blocksize**.

The unit of storage allocation in DCE LFS is the fragment, so this value controls the granularity of storage allocated to files. In other words, it affects the amount of space lost due to fragmentation.

-initialempty *blocks*

Specifies the number of DCE LFS blocks that DCE LFS leaves empty at the beginning of the disk partition when it initializes the aggregate.

The value provided must be an integer between 0 (zero) and 65,536 divided by the number of bytes specified with **-blocksize**. For example, if the value provided with **-blocksize** is 2048, the value provided with **-initialempty** cannot exceed 32 (65,536 divided by 2048).

The empty blocks reserved with this option are often used for a bootstrapping program. For this reason, the reserved blocks are often referred to as *bootblocks*.

If this option is omitted, one block is left empty at the beginning of the partition.

-aggrsize *blocks*

Specifies the total number of DCE LFS blocks to be available on the aggregate. Because this value cannot exceed the size of the disk partition, it can be used only to restrict the size of the aggregate. It must be large enough to accommodate at least the log and any blocks left empty at the beginning of the partition.

If this option is omitted, the default is the total number of DCE LFS blocks on the disk partition being initialized as a DCE LFS aggregate.

-logsize *blocks*

Specifies the number of DCE LFS blocks to be reserved for the log on the aggregate. This value cannot exceed the number of DCE LFS blocks used for **-aggrsize**, and it must contain at least enough blocks for the log to be initially created.

If this option is omitted, the default is 1% of the total number of DCE LFS blocks on the aggregate (the number of DCE LFS blocks used for **-aggrsize**).

-overwrite Specifies that any existing file system found on the partition can be overwritten when the aggregate is initialized. If this option is specified, an existing file system on the disk partition is automatically overwritten; the issuer is not prompted for confirmation.

If this option is omitted and an existing file system is found on the partition, the command displays a message informing the issuer that the **-overwrite** option must be used to overwrite an existing file system. It then terminates with an exit code of at least 16 without overwriting the existing file system.

-verbose Directs the command to provide more information on its actions as it executes. The information is displayed on standard output (**stdout**) unless it is directed elsewhere.

-noaction Directs the command to display information about what it would do without actually modifying the partition. Include the other options as you would to actually execute the command. The command displays the default values it would use for its options and informs the issuer if the disk partition already contains a file system.

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

The **help** and **apropos** commands available with all command suites are also available with the **newaggr** command. See the **bos help** and **bos apropos** reference pages for examples using these commands.

Description

The **newaggr** command is used to initialize a partition on the local disk of a machine for use as an aggregate with DCE LFS. The partition to be initialized as a DCE LFS aggregate is specified with the **-aggregate** option. The **newaggr** command formats the specified partition by creating the metadata structure used by DCE LFS for access control list (ACL) support, logging, and multiple fileset operations. It also creates temporary space on the disk used by the DCE LFS log for faster restarts after system failures. The binary file for the **newaggr** command resides in *dcelocal*/**bin/newaggr**.

An aggregate is a collection of DCE LFS disk blocks made up of the space available on the partition on which it resides. Each disk block on an aggregate has a fixed size specified with the **-blocksize** option. The **-blocksize** option specifies the number of bytes in each DCE LFS block. The value specified with this option must be a power of 2 between 1024 (1 kilobyte) and 65,536 (64 kilobytes).

Each block can be further decomposed into fragments. Each fragment on an aggregate has a fixed size specified with the **-fragsize** option. The **-fragsize** option specifies the number of bytes in each fragment. The value specified with this option must be a power of 2 between 1024 (1 kilobyte) and the value specified with the **-blocksize** option.

DCE LFS manages blocks and fragments as variable-length containers for the storage of user and system data. It manages filesets created on the aggregate as logically independent collections of data. Each fileset consists of a hierarchical collection of

files residing entirely within a single aggregate. DCE LFS obtains blocks for each fileset from a common allocation pool. As a result, filesets can share blocks (if the blocks are copy-on-write or if each fileset uses only a fragment of the block).

The **-initialempty** option can be used to reserve a number of empty blocks at the beginning of a partition. The empty blocks are referred to as *bootblocks* because they are often used for bootstrapping programs. The value provided with the **-initialempty** option must be an integer between 0 (zero) and 65,536 divided by the value specified with the **-blocksize** option. By default, one block is left empty.

The **-aggrsize** option can be used to restrict the number of DCE LFS blocks in the aggregate. By default, all of the blocks available on the disk partition to be initialized are used in the aggregate. The value specified with the **-aggrsize** option cannot exceed the size of the partition being initialized. It must be large enough to accommodate at least the log and any blocks left empty at the beginning of the partition.

The **-logsize** option can be used to specify the number of DCE LFS blocks to be reserved for the log on the aggregate. By default, 1% of the total number of DCE LFS blocks on the aggregate is reserved for the log. The value specified with the **-logsize** option cannot exceed the number of DCE LFS blocks used for the **-aggrsize** option, and it must specify at least enough blocks for the log to be initially created.

DCE LFS also reserves a variable amount of disk space on the aggregate. By default, DCE LFS reserves 2 megabytes of disk space on an aggregate. However, no less than 1% or no more than 10% of the total size of an aggregate is ever reserved; for example, only 1.5 megabytes are reserved on an aggregate whose total size is only 15 megabytes.

Reserved disk space is used for internal purposes. For example, the reserved space is used to avoid potential problems with routine administrative operations such as fileset moves and clones. The reserved space is not directly accessible to users and administrators. Use the **fts aggrinfo** command to display the total amount of disk space, including the amount of reserved disk space, on an aggregate.

If an existing file system on the disk partition being initialized is to be overwritten, include the **-overwrite** option with the command. The option instructs the command to overwrite any data found on the partition. To prevent an existing file system from being overwritten, omit the **-overwrite** option. If the command encounters an existing file system, it stops the initialization procedure without overwriting the existing file system and reports that it found a file system on the partition. It also instructs you to include the **-overwrite** option with the command to overwrite the resident file system.

Use the **-noaction** option to have the command report whether the partition already contains a file system or to display the values it calculates for the **-aggrsize** and **-logsize** options without actually overwriting a file system or initializing the partition. Specify all of the command's options as you would to actually execute the command, and include the **-noaction** option to display the results of the command without modifying the partition.

The **newaggr** command must be used to initialize a disk partition before the partition can contain DCE LFS filesets. After the disk partition is initialized as a DCE LFS aggregate with this command, an entry can be created for the aggregate in the **dfstab** file, and it can be exported to the DCE namespace with the **dfsexport** command. DCE LFS filesets can then be created on it with the **fts create** command and mounted in the global namespace with the **fts crmount** command.

Because the **newaggr** command overwrites all data on the partition being initialized, the partition must not be mounted locally and it should not contain data when the command is run. If the **newaggr** command is issued with the **-overwrite** option to create a DCE LFS aggregate on a disk partition that already contains a file system, the previous file system is destroyed. However, the command fails if it is run on an aggregate or partition that is currently exported to the DCE namespace, or if it is run on an aggregate that houses a locally mounted fileset. (If necessary, the **dfsexport** command can be used to detach an aggregate or partition from the namespace.)

In operating systems that support logical volumes, the command can be used to initialize a logical volume as a DCE LFS aggregate. In such cases, all of the command's functionality described here with respect to a disk partition applies to the logical volume.

Cautions

Do not use the **newaggr** command to create nonLFS aggregates. Do not use the command on a partition that contains data you want to retain; the command destroys all data on any partition it initializes. Do not use the command on a locally mounted partition; doing so causes the kernel to panic. Finally, do not use the command on a currently exported aggregate or partition, or on an aggregate that houses a locally mounted fileset; the command fails in these cases.

Privilege Required

If the **-noaction** option is *not* included with the command, the issuer must have both the read and write permissions for the device (disk partition) to be initialized as a DCE LFS aggregate; if the **-noaction** option is included with the command, the issuer

needs only the read permission for the specified device. An issuer who is logged in as **root** on the machine on which the specified device resides always has the necessary privilege to issue this command.

Related Information

Commands: **dfsexport(8dfs)**, **fts aggrinfo(8dfs)**, **growaggr(8dfs)**.

Files: **dfstab(4dfs)**.

repserver

Purpose

Initializes the Replication Server process

Synopsis

repserver [**-mainprocs** *number_of_background_daemons*] [**-tokenprocs** *number_of_token_daemons*] [**-help**]

Options

-mainprocs *number_of_background_daemons*
Specifies the number of background daemons to run on the machine. These daemons are responsible for the bulk of the effort required to maintain read-only replicas on the local machine, as well as for receiving and servicing RPC requests from DFS clients. If this option is omitted, four background daemons perform these services.

-tokenprocs *number_of_token_daemons*
Specifies the number of background daemons dedicated to servicing token revocation RPC requests from File Exporters. If this option is omitted, four background daemons service token revocation requests.

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

The **help** and **apropos** commands available with all command suites are also available with the **repserver** command. See the **bos help** and **bos apropos** reference pages for examples using these commands.

Description

The Replication Server, or **repserver** process, in conjunction with the Cache Manager, tracks the currency of replicas and updates the versions of data being used at each replication site. The **repserver** process is used in Release and Scheduled Replication, and must run on any machine that stores read-only replicas of read/write filesets. For simplified administration, run the **repserver** process on all File Server machines.

The **repserver** process is usually started and controlled by the BOS Server; if it is not, execute the **repserver** process as a background process. The binary file for the **repserver** process resides in *dcelocal*/**bin/repserver**.

The **-mainprocs** and **-tokenprocs** options can be used to alter the default number of background daemons running on the server machine, as follows:

-mainprocs Specifies the number of background daemons that run on the machine to maintain read-only replicas housed on the local machine and to service RPC requests from DFS clients. The default number of background daemons is four. Use the **-mainprocs** option to increase the number of background daemons if the machine houses a large number of replicas.

-tokenprocs Specifies the number of background daemons dedicated to handling token revocation RPC requests from the File Exporters on File Server machines. The default number of background daemons dedicated to this task is four. If the **-mainprocs** option is used to increase the number of background daemons dedicated to maintaining replicas and servicing RPC requests from DFS clients, use the **-tokenprocs** option to increase the number of background daemons dedicated to servicing token revocation requests from File Exporters.

When it is started, **repserver** creates the *dcelocal* **/var/dfs/adm/RepLog** event log file if the file does not already exist. It then appends messages to the file. If the file exists when **repserver** is started, the process moves it to the **RepLog.old** file in the same directory (overwriting the current **RepLog.old** file if it exists) before creating a new version to which to append messages.

Use the **fts statrepserver** command to check the status of the Replication Server on any server machine. Use the **fts lsreplicas** command to check the status of fileset replicas.

Privilege Required

The issuer must be logged in as **root** on the local machine.

Output

If problems are encountered during initialization, **repserver** sends error messages to standard error output (**stderr**). The **repserver** process keeps an event log in *dcelocal* **/var/dfs/adm/RepLog**.

Related Information

Commands: **fts lsreplicas(8dfs)**, **fts statrepserver(8dfs)**.

Files: **RepLog(4dfs)**.

salvage

Purpose

salvage – Uses the DFS Salvager to recover, verify, or salvage the structure of a DCE LFS aggregate

Synopsis

salvage - aggregate *name* [**- recoveronly**][{**- verifyonly** | **-salvageonly** }] [**-force**][**-verbose**][**-help**]

OPTIONS

-aggregate *name*
Specifies the device name or aggregate name of the DCE LFS aggregate to be verified, recovered, or salvaged. These names are specified in the first and second fields of the entry for the aggregate in the *dcelocal* **/var/dfs/dfstab** file.

-recoveronly
Directs the Salvager to recover the specified aggregate. The Salvager replays the log of metadata changes that resides on the aggregate. See the Description section for information about using and combining the command's options.

-verifyonly Directs the Salvager to verify the specified aggregate. The Salvager examines the structure of the aggregate to determine if it contains any inconsistencies, reporting any that it finds. See the Description section for information about using and combining the command's options.

-salvageonly Directs the Salvager to salvage the specified aggregate. The Salvager attempts to repair any inconsistencies it finds on the aggregate. See the Description section for information about using and combining the command's options.

-force Executes the Salvager in noninteractive mode. By default, the Salvager prompts for confirmation before proceeding in certain situations (for example, if it believes an aggregate on which it is run may be a nonLFS partition). Use this option to direct the Salvager to proceed with all operations without asking whether it should continue. Use this option

with care; the Salvager's changes can be unpredictable if this option is used with an invalid aggregate.

-verbose Directs the Salvager to produce detailed information about the aggregate as it executes. The information is useful primarily for debugging purposes. It is displayed on standard output (which can be redirected). Use this option alone or with any combination of the available options.

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

The **help** and **apropos** commands available with all command suites are also available with the **salvage** command. See the **bos help** and **bos apropos** reference pages for examples using these commands.

DESCRIPTION

The *dcelocal*/**bin/salvage** command invokes the DFS Salvager on the DCE LFS aggregate specified with the **-aggregate** option. Following a system restart, the Salvager employs the DCE LFS log mechanism to return consistency to a file system by running recovery on the aggregate on which the file system resides. Recovery is the replaying of the log on the aggregate; the log records all changes made to metadata as a result of operations such as file creation and deletion. If problems are detected in the basic structure of the aggregate, if the log mechanism is damaged, or if the storage medium of the aggregate is suspect, the **salvage** command must be used to verify or repair the structure of the aggregate.

Use the command's **-recoveronly**, **-verifyonly**, and **-salvageonly** options to indicate the operations the Salvager is to perform on the specified aggregate, as follows:

- Specify the **-recoveronly** option to run recovery on the aggregate without attempting to find or repair any inconsistencies found on it. Recovery is the replaying of the log on the aggregate. Use this option to quickly return consistency to an aggregate that does not need to be salvaged; this represents the normal production use of the Salvager. Unless the contents of the log or the physical structure of the aggregate is damaged, replaying the log is an effective guarantee of a file system's integrity.
- Specify the **-verifyonly** option to determine whether the structure of the aggregate contains any inconsistencies without running recovery or attempting to repair any inconsistencies found on the aggregate. Use this option to assess the extent of the damage to an aggregate. The Salvager makes no modifications to an aggregate

during verification. Note that it is normal for the Salvager to find errors when it verifies an aggregate that has not been recovered; the presence of an unrecovered log on an aggregate makes the findings of the Salvager, positive or negative, of dubious worth.

- Specify the **-recoveronly** and **-verifyonly** options to run recovery on the aggregate and then analyze its structure without attempting to repair any inconsistencies found on it. Use these options if you believe replaying the log can return consistency to the aggregate, but you want to verify the consistency of the aggregate after recovery is run. Recovering an aggregate and then verifying its structure represents a cautious application of the Salvager.
- Specify the **-salvageonly** option to attempt to repair any inconsistencies found in the structure of the aggregate without first running recovery on it. Use this option if you believe the log is damaged or replaying the log will not return consistency to the aggregate and may in fact further damage it. Under normal circumstances, do not salvage an aggregate without first recovering it.
- Omit the **-recoveronly**, **-verifyonly** , and **-salvageonly** options to run recovery on the aggregate and then attempt to repair any inconsistencies found in the structure of the aggregate. Because recovery eliminates inconsistencies in an undamaged file system, an aggregate is typically recovered before it is salvaged. In general, it is good first to recover and then to salvage an aggregate if a machine panics or experiences a hardware failure.

  Omit these three options if you believe the log should be replayed before attempts are made to repair any inconsistencies found on the aggregate. (Omitting the three options is equivalent to specifying the **-recoveronly** and **-salvageonly** options.)

The following rule summarizes the interaction of the **-recoveronly** , **-verifyonly**, and **-salvageonly** options: The **salvage** command runs recovery on an aggregate and attempts to repair it *unless* one of the three options is specified; once one of these options is specified, you must explicitly request any operation you want the Salvager to perform on the aggregate.

The basic function of the Salvager is similar to that of the UNIX **fsck** program. The Salvager recovers a DCE LFS aggregate and repairs problems it detects in the structure of the aggregate. It does not verify or repair the format of user data contained in files on the aggregate. If it makes changes, the Salvager displays the pathnames of the files affected by the modifications, when the pathnames can be determined. The owners of the files can then verify the files' contents, and the files can be restored from backups if necessary.

The Salvager verifies the structure of an aggregate by examining all of the anodes, directories, and other metadata in each fileset on the aggregate. An anode is an area on the disk that provides information used to locate data such as files, directories, ACLs, and other types of file system objects. Each fileset contains an arbitrary number of anodes, all of which must reside on the same aggregate. By following the links between the various types of anodes, the Salvager can determine whether the organization of an aggregate and the filesets it contains is correct and make repairs if necessary.

Not all aggregates can be salvaged. In cases of extensive damage to the structure of the metadata on an aggregate or damage to the physical disk that houses an aggregate, the Salvager cannot repair inconsistencies. Also, the Salvager cannot verify or repair damage to user data on an aggregate. The Salvager cannot detect problems that modified the contents of a file but did not damage the structure of an aggregate or change the metadata of the aggregate.

Like the UNIX **fsck** command, the Salvager analyzes the consistency of an aggregate by making successive passes through the aggregate. With each successive pass, the Salvager examines and extracts a different type of information from the blocks and anodes on the aggregate. Later passes of the Salvager use information found in earlier passes to help in the analysis.

Unlike the **fsck** command, the Salvager does not normally prompt for additional information as it executes. It typically performs the requested operation without prompting for input or pausing to verify any changes before it makes them. It prompts for confirmation only in the following cases:

- It believes the specified aggregate does not contain a DCE LFS file system. This can occur if it finds a nonLFS superblock whose creation time is more recent than the creation time of the DCE LFS superblock.
- It finds that the size of the aggregate recorded in the DCE LFS superblock exceeds the capacity of the partition on which the aggregate resides.

At the prompt, you can choose to cancel or continue the operation. If you continue the operation under either of these circumstances and the aggregate proves to be invalid, unpredictable results can ensue. The best response in either case is to cancel the operation and attempt to determine the cause of the problem.

If you are confident that you want the Salvager to continue in any case, you can include the **-force** option with the command. This option directs the Salvager to perform the requested operation without prompting for confirmation. Exercise caution when using the **-force** option; the Salvager can produce unpredictable results if this option is used with an invalid aggregate.

In general, the Salvager exits with an error code of at least **16** without analyzing a partition that it is sure is not a DCE LFS aggregate. It also exits with an error code of **16** if an aggregate to be recovered or salvaged is currently exported to the global namespace, or if a fileset on the aggregate to be recovered or salvaged is mounted locally. (If necessary, you can use the **dfsexport** command to detach an exported aggregate from the namespace.)

As the Salvager executes, it maintains a number of internal lists. Each list consists of anodes that failed verification in specific ways. When it initially scans an aggregate, the Salvager marks as "unsafe" anodes with which it encounters problems. The Salvager later attempts to determine the actual pathnames associated with these anodes to include the pathnames in the lists. When it has finished salvaging, the Salvager displays any nonempty lists. It also returns one of a number of informative exit codes, depending on the inconsistencies it found and the repairs it made. More information about the lists and exit codes displayed by the Salvager appears later in this reference page.

Internal structures maintained by the Salvager require a minimum of 1 megabyte of swap space. However, the total amount of swap space required by the Salvager depends largely on the size of the aggregate being salvaged and the extent of the damage to the aggregate.

Privilege Required

The privileges required depend on whether the **-recoveronly**, **-verifyonly**, or **salvageonly** option is specified with the command: If just the **-verifyonly** option is included, the issuer needs only the read permission for the specified device (aggregate); if the **-recoveronly** or **-salvageonly** option is included, or if all three of these options are omitted, the issuer must have both the read and write permissions for the specified device. An issuer who is logged in as **root** on the machine on which the specified device resides always has the necessary privilege to issue the command.

CAUTIONS

The Salvager can be used to salvage only DCE LFS aggregates. If it is executed on a nonLFS partition, it exits with an error code of at least **16** without performing any operations. Use the UNIX **fsck** program or its equivalent to verify or restore consistency to nonLFS disk partitions.

By default, the Salvager asks for confirmation before proceeding with operations on aggregates that it suspects are nonLFS partitions or whose indicated sizes exceed the capacities of the partitions on which they reside. The command’s **-force** option can

be used to direct the Salvager to continue without prompting in these cases. Do not include the **-force** option under normal conditions; the Salvager can make undesirable changes if the option is used with an invalid aggregate.

If the Salvager is used to recover or salvage an aggregate that is currently exported, it exits with an error code of **16** without performing the operation. Use the **dfsexport** command to detach an aggregate from the global namespace if necessary before recovering or salvaging it. (The Salvager can be used to verify the structure of a currently exported aggregate, but this is not a good practice; the results may be misleading.) The Salvager also exits with an error code of **16** if a fileset on an aggregate to be recovered or salvaged is mounted locally.

OUTPUT

The Salvager sends output to both **stdout** and **stderr**. When it is started, the Salvager displays the device name of the aggregate on which it is run and the operation it is to perform. For example, the Salvager displays the following message if it is directed to recover an aggregate:

```
Will run recovery on device
```

Similarly, the Salvager displays the following message if it is directed to verify an aggregate:

```
Verifying device
```

If you specify the **-verbose** option with the command, the Salvager generates the following information about the aggregate:

- Physical information about the configuration of the aggregate
- Header information from the aggregate, including the major and minor number of the device on which the aggregate was created, and the date and time at which the aggregate was created
- Information about how space in the aggregate is allocated, including
  - The total size of the aggregate in blocks

— The block size

— The fragment size

— The number of the first block in the aggregate

— The location of the principal superblock for the aggregate

— The number of logical blocks in the aggregate

If you use the Salvager to recover an aggregate and the log on the aggregate does not need to be replayed, the Salvager displays only the introductory message described previously. If the log does need to be replayed and the Salvager can successfully recover the aggregate, the Salvager displays the following messages:

```
Recovery statistics
  statistics
Ran recovery on device
```

In the output, *statistics* consists of a few lines of information about the log and its replaying, and *device* is the device name of the aggregate. If it cannot run recovery for any reason, the Salvager displays an appropriate exit code. (All Salvager exit codes are listed at the end of this section.)

The Salvager can display much more output if it is asked to verify or salvage an aggregate on which it finds metadata errors. As it verifies or salvages a damaged aggregate, it displays a message similar to the following for each fileset in which it encounters metadata problems:

```
In volume fileset (avl #integer)
  in anode (#integer)
     description
```

It displays the first line once for each fileset, repeating the second and third lines once for each problem anode in the fileset. The output provides the following information:

fileset The name and ID number of each affected fileset.

avl #*integer* A pointer to the anode for the fileset.

in anode (#*integer***)**
A pointer to the anode for a file or other object in the fileset.

description A brief description of the problem the Salvager found with the anode. If it was used to salvage the aggregate, the Salvager also describes any actions it took to repair the anode.

When it has finished executing, the Salvager lists each file whose metadata it found to be damaged, many of which it likely repaired if it salvaged the aggregate. For each file, it displays a line of the form

```
condition   fileset:pathname volume index: integer anode index: integer
```

The output provides the following information:

condition A string that describes the state of the file or its metadata. (Information about the possible conditions follows this list.)

fileset The name of the fileset in which the affected file resides. In some cases, the Salvager cannot determine the fileset name.

pathname The pathname of the file, relative to the root directory of the fileset. In some cases, the Salvager cannot reconstruct the pathname for a file.

volume index
A pointer to the anode for the fileset. (This information can be used to identify earlier message displayed by the Salvager that are related to this file.)

anode index A pointer to the anode for the file. (This information can be used to identify earlier message displayed by the Salvager that are related to this file.)

The following conditions accompany the files most in need of attention:

oughtRestore
Files in which one or more block references in the associated anode were removed or changed. Because it is unlikely such files contain all of their original data, these files should be restored from existing backups. This condition applies only to files on salvaged aggregates.

mayRestore Files to which modifications were made (for example, files whose ACLs or property lists were changed). The owners of these files should verify their contents, or a system administrator should simply restore them from backups if a directory listing indicates that they have not been

modified since the last backup was made. This condition also applies only to files on salvaged aggregates.

zeroLinkCnt
: Files whose link counts should be 0 (zero). These files were deleted but not closed when the system crashed or were orphaned by the Salvager as it made repairs to the file system. The system will delete them when the aggregate is exported.

badLinkCnts
: Files whose link counts were inconsistent with the number of references found to them. These files should be examined, if possible, or simply restored.

The Salvager can list a file more than once if it determines that multiple conditions apply to the file. It can also display one or more additional conditions (such as **badAcls** or **badPlists**), but files with which the additional conditions are associated are typically already covered by one or more of the conditions just described. Information in the additional lists is useful primarily for debugging purposes.

The Salvager also returns one of various exit codes to summarize its actions and findings. It returns the exit codes in the form of bits, which it uses to indicate the state of the aggregate. It can set multiple bits, but, in general, the higher the bit, the greater the severity of the aggregate's problems. (The higher bit always takes precedence when interpreting the output.) The Salvager can return the following exit codes:

All bits off
: The Salvager found no problems. It displays a message that includes **Done** and **Checks out**. The command need not be run again.

First bit (**0x1**) set
: The Salvager found one or more problems. It displays a message that includes **Done** and **Some inconsistencies found**. Run the command on the aggregate without the **-verifyonly** option to attempt to correct the problems.

Second bit (**0x2**) set
: The Salvager found one or more problems and fixed them. It displays a message that includes **Done** and **Some inconsistencies repaired** . The command need not be run again. (Note that if the second bit is set, the first bit is usually also set; because the higher bit takes precedence, you do not need to run the command again.)

Third bit (**0x4**) set
: The Salvager found one or more problems and fixed some of them. It displays a message that includes **Incomplete** and **Some repairs made**. Some problems were more severe and require a subsequent salvage to be repaired; run the command on the aggregate without the **-verifyonly** option to attempt to correct the problems.

Fourth bit (**0x8**) set
: The Salvager found the aggregate to be irreparably damaged. It displays a message that begins **Problem**. Use the **newaggr** command to reinitialize the aggregate, and reconstruct the data from existing backups if possible.

Fifth bit (**0x10**) set
: The Salvager found some serious problem that prevents it from running on the aggregate; for example, the attempted recovery of the aggregate failed because of damage to the log, or the attempted salvage of the aggregate failed because the aggregate is not a DCE LFS aggregate, it is currently exported, or it contains a locally mounted fileset. The Salvager displays a message that begins **Problem**. Attempt to determine the cause of the problem.

Including the **-verbose** option with the command produces more detailed information about the aggregate as the command executes. However, the additional information is useful primarily for debugging purposes.

EXAMPLES

The following command instructs the Salvager to recover the DCE LFS aggregate whose device name is **/dev/lv01**. This example represents the most-common application of the Salvager.

```
# salvage /dev/lv01 -recover
```

The following command instructs the Salvager to analyze the structure of the aggregate to determine if it contains any inconsistencies without running recovery or attempting to repair the inconsistencies:

```
# salvage /dev/lv01 -verify
```

The following command directs the Salvager to repair any inconsistencies it finds on the aggregate without first running recovery:

```
# salvage /dev/lv01 -salvage
```

RELATED INFORMATION

Commands: **dfsexport(8dfs)**, **newaggr(8dfs)**.

Files: **dfstab(4dfs)**.

scout

Purpose

scout – Initializes the **scout** program

Synopsis

scout -server *machine*… [**-basename** *common_prefix*] [**-host**][**-frequency** *seconds*] [**-attention** *stat/threshold_pair*]… [**-debug** *filename*] [**-help**]

Options

-server *machine*
Specifies each File Server machine whose File Exporter is to be monitored. Use one of the following to indicate each File Server machine:

- The machine's DCE pathname (for example, */.../abc.com/hosts/fs1*). If you use the **-basename** option to specify a pathname prefix common to all machines to be monitored, you need to provide only the unique suffix of each machine name; you can omit the common DCE pathname prefix.
- The machine's host name (for example, **fs1.abc.com** or **fs1**).
- The machine's IP address (for example, **11.22.33.44**).

-basename *common_prefix*
Specifies the DCE pathname prefix (for example, */.../abc.com/hosts*) common to all File Server machines specified with the **-server** option. Do not include the **/** (slash) that separates the prefix from the unique part of each machine name; it is included automatically with the **-basename** option. The basename, if specified, is displayed in the banner line.

Use this option only if you are specifying the DCE pathname of each File Server machine to be monitored. Omit this option if you are specifying the host names or IP addresses of one or more machines.

-host Displays the name of the machine running the **scout** program in the banner line. This is useful if you are logged into the machine remotely. By default, **scout** does not display this name.

-frequency *seconds*

Indicates how often the **scout** program is to probe the File Exporters. Specify a positive integer as a value in seconds; the default is 60 seconds.

-attention *stat/threshold_pair*

Specifies a list of attention settings (statistic and threshold value pairs). The **scout** program highlights any value for a statistic that exceeds its specified threshold; the highlighting is removed when the value goes below the threshold. The pairs can appear in any order. Legal statistic/ threshold pairs are

conn *connections*

The maximum number of connections that principals can have open to the File Exporter before the value is highlighted. Enter a threshold for this statistic in the form of a positive integer.

fetch *number_of_fetches*

The maximum number of fetches (requests to send data) the File Exporter can service before the value is highlighted. Enter a threshold for this statistic in the form of a positive integer. The highlighting is removed when the File Exporter is restarted, at which time the value returns to **0** (zero).

store *number_of_stores*

The maximum number of stores (requests to store data) the File Exporter can accept before the value is highlighted. Enter a threshold for this statistic in the form of a positive integer. The highlighting is removed when the File Exporter is restarted, at which time the value returns to **0** (zero).

ws *active_client_machines*

The maximum number of active client machines the File Exporter can serve before the value is highlighted. *Active* indicates those machines that communicated with the File Exporter in the past 15 minutes. Enter a threshold for this statistic in the form of a positive integer.

disk *percent_full***%**

The maximum percentage of an aggregate that can contain data before the value is highlighted. This threshold is

applied to all exported aggregates and partitions on a File Server machine being monitored. Legal thresholds are the integers between 0 (zero) and 99; the default is 95%. *You must enter the % (percent sign) with this threshold.* If the % (percent sign) is absent, **scout** interprets the number as a number of kilobyte blocks. Use this threshold or use **disk** *minimum_blocks_free*.

disk *minimum_blocks_free*
The minimum number of kilobyte blocks that must be available on an aggregate before the value is highlighted. This threshold is applied to all exported aggregates and partitions on a File Server machine being monitored. Enter a threshold for this statistic in the form of a positive integer. Use this threshold or use **disk** *percent_full***%**.

-debug *filename*
Enables debugging output and directs it to the specified *filename*. Provide a complete pathname for *filename*; the current working directory is used by default. If this option is omitted, no debugging output is written.

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

The **help** and **apropos** commands available with all command suites are also available with the **scout** command. See the **bos help** and **bos apropos** reference pages for examples of these commands.

Description

The **scout** command displays statistics gathered from the File Exporter running in the kernel on each File Server machine specified with the **-server** option. Usage statistics are also displayed about exported aggregates and partitions on the File Server machine being monitored. The **scout** program can be run on any DFS client or server machine. The binary file for the program resides in *dceshared* **/bin/scout**.

To change attention settings (statistic and threshold pairs), you must stop and restart the **scout** program. In addition, **scout** does not store the settings from previous executions; you must specify the desired settings each time you start the program.

Both terminals and windowing systems that emulate terminals can display **scout** statistics. The **scout** display uses reverse video and cursor addressing; therefore, the

display environment must support these features. The issuer must set the **TERM** environment variable to the correct terminal type or to one with similar characteristics.

To stop the **scout** program, enter the interrupt command (**<Ctrl-c>** or its equivalent) for your operating system in the **scout** window.

The **scout** program can display statistics in either a dedicated window or on a plain screen if a windowing environment is unavailable. The **scout** screen has three main parts: the Banner Line, the Statistics Display Region, and the Message/Probe Line.

The Banner Line at the top of the window or screen displays the word **Scout**, indicating the program is running. The name of the machine executing **scout** is displayed if the **-host** option is specified, and the basename of the File Server machines being monitored is displayed if the **-basename** option is specified.

The Statistics Display Region displays the statistics **scout** has gathered for each File Exporter. The region is divided into six columns, one column for each of the five statistic and threshold pairs used with the **-attention** option, and one column for the name of each File Server machine being monitored. In addition to highlighting any value that exceeds its specified attention threshold, **scout** highlights the name of any File Server machine whose File Exporter fails to respond to **scout**'s probes. The name remains highlighted until the machine resumes responding to **scout**'s probes.

The Message/Probe Line at the bottom of the window or screen indicates how many times **scout** has probed the File Exporters for statistics. Use the **-frequency** option to specify how frequently **scout** is to probe the File Exporters.

Privilege Required

No privileges are required.

Examples

The following **scout** command causes the program to monitor the File Exporters on File Server machines **fs1** and **fs2** in the cell *abc.com*. The **scout** program probes the File Exporters every 30 seconds and writes debugging information to the file named **scout.one** in the current working directory.

```
$ scout -server fs1 fs2 -basename /.../abc.com/hosts -frequency 30 -debug scout.one
```

The following command causes **scout** to monitor the same two machines. The **scout** program highlights an entry in the **Fetch** column if the File Exporter services 20,000 or more fetches, and it highlights an entry in the **Store** column if the File Exporter accepts 10,000 or more stores.

```
$ scout -server fs1 fs2 -b /.../abc.com/hosts -attention fetch 16 store 8
```

udebug

Purpose

udebug – Displays Ubik status information relevant to the specified DFS database server

Synopsis

udebug - **rpcgroup** *RPC_server_group* [- **server** *machine*] [- **long**][-**help**]

Options

-rpcgroup*RPC_server_group*
Specifies the RPC server group of the Ubik database servers whose status information you want to display. By convention, this is **/.:/fs** for the **flserver** processes and **/.:/subsys/dce/dfs/bak** for the **bakserver** processes.

-server *machine*
Names the machine containing the database server whose Ubik status information is to be displayed; if a machine name is omitted, the command uses the name of the local machine. Specify the server machine using the full DCE pathname, abbreviated host name, or IP address.

-long Directs the command to provide additional information about the other database servers in the specified RPC server group. This flag is *not* necessary if the server specified with the **-server** option is the Ubik synchronization site because the information about the other database servers is provided automatically.

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

Description

The **udebug** command displays Ubik status information on the specified server in the specified RPC server group. If the specified server is the synchronization site or the

-long flag is used with the command, the command displays information on all of the servers in the RPC server group.

Privilege Required

No privileges are required.

Output

The output for the **udebug** command always provides the following information for the specified database server:

- The IP address of the specified server machine. In the first example, this is **192.56.207.146**.
- The difference in seconds between the clock on the specified server machine and the machine on which the **udebug** command was run. In the first example, this is **0**.

  Note: If the message ******clock may be bad** appears, the difference between the two clocks is greater than 40 seconds, and you must synchronize the clocks on all of the server machines in the RPC server group.
- The IP address of the server machine that this server voted for to be the synchronization site and the time that the vote was cast. In the first example, this is **192.56.207.26** at **-10**.

  Note: Unless noted otherwise, all time is calculated and displayed as the number of seconds before (negative) or after (positive) the current time according to the clock on the local machine on which the **udebug** command is run.
- The time at which the last round of sync-site voting began. In the first example, this is **–11**.
- The version of the database in use on this server machine. In the first example, this is **750478963.1**.
- Whether the server is the synchronization site; if it is, the duration of the synchronization site status and the number of servers in the RPC server group are also provided. In the first example, the message **I am not sync site** indicates that the server is not the synchronization site.
- If the server is *not* the synchronization site, the following information is displayed:

— The IP address of the lowest server in the RPC server group and the time that a beacon was last sent from that server to the specified server. In the first example, this is **192.56.207.26** at **-10**.

— The IP address of the synchronization site and the time that a beacon was last sent from that server. In the first example, this is **192.56.207.26** at **-10**.

If the server is the synchronization site, the current state of the server is displayed, using one of the following flags. In the second example, this is **1f**.

— **1** – Indicates that the server is the synchronization site.

— **3** – Indicates that the server is the synchronization site and that it has found the latest version of the database.

— **7** – Indicates that the server is the synchronization site and that it has fetched the latest version of the database.

— **f** – Indicates that the server is the synchronization site and a quorum has been reached in the RPC server group, but the synchronization site has not distributed the latest version of the database to all servers in the RPC server group.

— **1f** – Indicates that server is the synchronization site, a quorum has been reached in the RPC server group, and the synchronization site has distributed the latest version of the database to all servers in the RPC server group.

- The version of the database in use at the synchronization site. In the first example, this is **750478963.1**.
- The total number of database pages locked and the number of database pages locked for write purposes on the server. (Anything other than a 0 indicates database activity.) In the first example, this is **0** and **0**.
- The time that the server was the synchronization site, if it ever has been, or a message indicating that the server has never been the synchronization site. In the first example, the message **This server has never been sync site** indicates that the server has never been the synchronization site.

If the **udebug** command specifies the synchronization site of the RPC server group or if the **-long** option is used with the command, the following additional information is displayed for each of the other database servers in the RPC server group:

- The IP address of each server machine. In the second example, the first server machine listed has the IP address **192.56.207.36** .

- The version of the database in use on each server machine. (A value of **0.0** indicates that the server does not have a version of the database.) In the second example, the first server listed uses the database version **750478963.1**.
- The last time a vote was received from this server by the server specified with the **-server** option. In the second example, the server with IP address **192.56.207.26** received a vote from the first server with IP address **192.56.207.36** at **-8**.
- The last time a beacon requesting a vote was sent to each server. In the second example, the first server received a beacon at **-9**.
- The last vote, yes or no, cast by each server. In the second example, the first server cast a **yes** vote.
- A flag (**dbcurrent**) indicating whether the version of the database in use on each server machine is current with the synchronization site; 0 indicates no, 1 indicates yes. In the second example, the first server has a current version of the database.
- A flag (**up**) indicating whether the corresponding server process on each server machine is up; 0 indicates no, 1 indicates yes. In the second example, the first server is up.
- A flag (**beaconSince**) indicating whether a response (vote) to the latest beacon was sent by each server to the synchronization site. In the second example, the first server sent a response to the latest beacon.

Examples

The following command displays information on a specified database server that is not a synchronization site:

```
$ udebug /.:/fs fs2
```

```
Host 192.56.207.146, his time is 0
Vote: Last yes vote for 192.56.207.26 at -10 (sync site);
Last vote started at -11
Local db version is 750478963.1
I am not sync site
Lowest host 192.56.207.26 at -10
```

```
Sync host 192.56.207.26 at -10
Sync site’s db version is 750478963.1
0 locked pages, 0 of them for write
This server has never been sync site
```

The following command displays information on a specified database server that is a synchronization site; the output also provides information on the other database servers in the RPC server group:

$ udebug /.:/fs fs4

```
Host 192.56.207.26, his time is 0
Vote: Last yes vote for 192.56.207.26 at -9 (sync site);
Last vote started
at -9
Local db version is 750478963.1
I am sync site until 81 (4 servers)
Recovery state 1f
Sync site’s db version is 750478963.1
0 locked pages, 0 of them for write
This server last became sync site at -38195


Server 192.56.207.36: (db 750478963.1)
last vote rcvd at -8, last beacon sent at -9, last vote was yes
dbcurrent=1, up=1 beaconSince=1


Server 192.56.207.146: (db 750478963.1)
last vote rcvd at -8, last beacon sent at -9, last vote was yes
dbcurrent=1, up=1 beaconSince=1


Server 192.56.207.94: (db 750478963.1)
last vote rcvd at -8, last beacon sent at -9, last vote was yes
dbcurrent=1, up=1 beaconSince=1
```

Related Information

Commands: **bakserver(8dfs)**, **flserver(8dfs)**.

upclient

Purpose

upclient – Initializes the client portion of the Update Server

Synopsis

upclient -server *machine* **-path** {*filename* | *directory_name*}… [**-time** *frequency*] [**-file** *log_file*] [**-verbose**][**-help**]

Options

-server *machine*
Specifies the DCE pathname of the machine (for example, */.../abc.com/hosts/fs1*) whose files are to be periodically checked. The machine should be either the System Control machine for the cell or domain or the Binary Distribution machine for the local machine's CPU/operating system type.

-path {*filename* | *directory_name*}
Names each file or directory to be checked periodically on the local disk of the machine specified with the **-server** option. If multiple paths are supplied, they must be unique, disjoint trees in the file system. Paths are examined from left to right; paths that intersect with previous paths used in the command are logged as errors (if a log file is specified with the **-log** option) and ignored.

If you specify a directory, the **upclient** process recursively checks all files and directories located beneath the specified directory. Therefore, you can specify a / (slash) to check all files and directories on the local disk of the machine specified with the **-server** option.

-time *frequency*
Specifies in number of seconds how often the **upclient** process is to check each file or directory specified with the **-path** option. The default is 300 seconds (5 minutes).

-file *log_file* Names the log file on the local machine to which errors are to be written. Because multiple **upclient** processes can be run on one machine, choose

a distinct filename for the log. If this option is omitted, no errors are logged.

-verbose Directs the **upclient** process to produce detailed information about its actions each time it checks for new versions of files (as specified with the **-time** option). The process lists each file and directory object it checks and any changes it makes to local versions of these objects. The output is sent to standard error.

-help Prints the online help for this command. All other options specified with this option are ignored.

The **help** and **apropos** commands available with all command suites are also available with the **upclient** command. See the **bos help** and **bos apropos** reference pages for examples of using these commands.

Description

The **upclient** command initializes the client portion (**upclient** process) of the Update Server. The **upclient** process periodically checks specified files and directories on the local disk of the server machine specified with the **-server** option to be sure they match the corresponding files and directories on the local machine (the machine running the **upclient** process). If a file on the specified server machine does not match the version on the local machine, the **upclient** process requests the newer version from the server portion (**upserver** process) of the Update Server on the specified machine. It then overwrites the local version of the file with the newer version.

The **upclient** process is usually started and controlled by the BOS Server; if it is not, execute the **upclient** process as a background process. The binary file for the **upclient** process resides in */dcelocal*/**bin/upclient**.

The **-time** option specifies how often the **upclient** process is to check for changed versions of files and directories. The **-path** option specifies the files and directories the **upclient** process is to check. If you specify a directory, the **upclient** process recursively checks all files and directories located beneath the specified directory. To check all files and directories on the indicated server machine, specify a **/** (slash) with the **-path** option.

If you specify multiple files and directories with the **-path** option, the paths must be disjoint (nonintersecting). Pathnames specified with the **-path** option are examined from left to right. Any path that intersects with a previous path is logged as an error (if a log file is named with the **-file** option) and ignored. An error also occurs if the

-path option names a file or directory that the **upserver** process on the specified server machine is not directed to distribute. Because multiple **upclient** processes can be run on a single machine, a filename specified with the **-file** option must be distinct.

Finally, the machine running the **upclient** process must be named in the **admin.up** file on the machine running the **upserver** process (the machine specified with the **-server** option). Otherwise, the local machine's **upclient** process is not permitted to access files from the **upserver** process.

Privilege Required

You must be logged in as **root** on the local machine.

Examples

The following command starts the **upclient** process running on the local machine. The process is to check every 180 seconds (3 minutes) for changes to the binary files in the directory **/rs_aix32/bin** on the Binary Distribution machine named */.../abc.com/hosts/fs1*. Errors are written to the file named **/tmp/fs1/UpclientLog** on the local disk of the machine running the **upclient** process.

```
$ upclient -s /.../abc.com/hosts/fs1 -p /rs_aix32/bin -t 180 -l /tmp/fs1/UpclientLog
```

Related Information

Commands: **upserver(8dfs)**.

Files: **admin.up(4dfs)**.

upserver

Purpose

upserver – Initializes the server portion of the Update Server

Synopsis

upserver -path {*filename* | *directory_name*}… [**-adminlist** *filename*] [**-help**]

Options

-path {*filename* | *directory_name*}

Names each file or directory to be distributed (exported) in unencrypted form upon request. If multiple paths are supplied, they must be unique, disjoint trees in the file system. Paths are examined from left to right; paths that intersect with previous paths used in the command are logged as errors and ignored.

All files and subdirectories located beneath a specified directory can be distributed from the local machine. Therefore, you can specify a / (slash) to allow all files and directories on the local disk of the machine to be distributed.

-adminlist *filename*

Specifies the file that contains server principals authorized to request files from the local machine. If you do not specify the complete pathname of a file, the file is assumed to reside in the current working directory. If this option is omitted, the **upserver** process uses the default file (*dcelocal*/**var/dfs/admin.up**).

-help Prints the online help for this command. All other options specified with this option are ignored.

The **help** and **apropos** commands available with all command suites are also available with the **upserver** command. See the **bos help** and **bos apropos** reference pages for examples of using these commands.

Description

The **upserver** command initializes the server portion (**upserver** process) of the Update Server. The **upserver** process distributes files from the local disk of a machine in response to requests from the client portion (**upclient** process) of the Update Server running on other machines. An **upserver** process should be run on the System Control machine for the cell or domain and on the Binary Distribution machine for each CPU/ operating system type.

The **upserver** process is usually started and controlled by the BOS Server; if it is not, execute the **upserver** process as a background process. The binary file for the **upserver** process resides in *dcelocal*/**bin/upserver**.

The **-path** option specifies the files and directories the **upserver** process can distribute from the local disk of the machine on which it is run. The **upserver** process can distribute all files and subdirectories located beneath a specified directory on the local machine; an **upclient** process can request and receive any file from the specified directory. Specify a / (slash) to allow all files and directories on the local disk of the machine to be distributed.

If the **-path** option names only a single file from a directory, an **upclient** process can request and receive only that file. An **upclient** process that requests the entire directory in which the file resides receives no files. If you specify multiple files and directories, the paths must be disjoint (nonintersecting). Paths are examined from left to right; any path that intersects with a previous path is logged as an error and ignored.

The **upserver** process writes error messages to the *dcelocal*/**var/dfs/adm/UpLog** event log file. When the **upserver** process is started, it creates the **UpLog** file if the file does not already exist. It then appends messages to the file. If the file exists when the **upserver** process is started, the process moves it to the **UpLog.old** file in the same directory (overwriting the current **UpLog.old** file if it exists) before creating a new version to which to append messages.

Only one **upserver** process should be run on a machine at one time. The **upserver** process automatically creates the *dcelocal*/**var/dfs/admin.up** file if the file does not already exist. A machine must be named in the **admin.up** file for its **upclient** process to be permitted to access files from the **upserver** process.

Privilege Required

You must be logged in as **root** on the local machine.

Examples

The following command specifies that files from the directories **/rs_aix32/bin** and */usr/mike*, which reside on the local disk of the machine, are to be exported upon request from **upclient** processes. The indicated paths are nonintersecting, so the command executes as intended.

```
$ upserver -path /rs_aix32/bin /usr/mike
```

The following command specifies that files from the directories **/rs_aix32/bin**, */usr/mike/public*, and */usr/mike*, which are located on the local disk, are to be exported upon request. However, because the path */usr/mike/public* is a subset of the path */usr/mike*, the command logs an error in the **UpLog** file and ignores the */usr/mike* path. The */usr/mike/public* path is exported as requested.

```
$ upserver -path /rs_aix32/bin /usr/mike/public /usr/mike
```

Had */usr/mike* been specified before */usr/mike/public* in the previous command, the */usr/mike/public* path would have been logged as an error in the **UpLog** file and ignored. In this case, the */usr/mike* path would have been exported as intended.

Related Information

Commands: **upclient(8dfs)**.

Files: **admin.up(4dfs)**, **UpLog(4dfs)**.

Appendix A

The DFS/NFS Secure Gateway

The Distributed File Service/Network File System (DFS/NFS) Secure Gateway provides a mechanism for granting authenticated access to the DFS filespace from an NFS client. The DFS/NFS Secure Gateway allows users to access data in the DFS filespace from a machine that is configured as an NFS client but not as a DCE client.

To use the DFS/NFS Secure Gateway for authenticated access to DFS, you must configure at least one Gateway Server machine. A Gateway Server machine must be a DFS client in the DCE cell to which access is to be provided. One function of a Gateway Server machine is to export the root of the DCE global namespace, **/...**, via NFS. On each NFS client from which users are to access DFS, you then mount **/...**. All users of the NFS clients then have unauthenticated access to DFS.

The DFS/NFS Secure Gateway recognizes the **@sys** and **@host** variables on the NFS client system. This allows the Gateway system to resolve pathnames to binaries and other system-dependent files correctly, based on the user's login system name and system type.

The **@host** variable contains the name of the NFS client, as returned by **gethostname()**, that is accessing the DFS namespace. The **@sys** variable, which is a

unique name derived from **uname()**, describes the machine architecture and OS type. Examples of **@sys** values include **pmax_osf1** and **rs_aix32**.

The primary function of a Gateway Server machine is to provide DCE authentication to users of NFS clients. NFS users who have valid accounts in the registry database of the DCE cell authenticate to DCE to gain authenticated access to DFS. Depending on the needs of your users and the security considerations of your DCE cell, you can provide local authentication to DCE from Gateway Server machines, remote authentication to DCE from NFS clients, or both. Local and remote authentication work as follows:

- *Local authentication* to DCE from Gateway Server machines is provided via the **dfsgw add** command. With local authentication, you can allow users to issue the **dfsgw add** command to authenticate themselves, or you can control access to DFS by allowing only system administrators to provide authentication via the **dfsgw add** command. (The **dfsgw** command suite includes additional commands to provide for central administration from Gateway Server machines.)

  Local authentication requires little configuration, but it provides a limited approach to authentication. Configuration consists only of installing the **dfsgw** commands on the Gateway Server machines. However, authentication requires either administrative intervention or remote access to the Gateway Server machine (via the **telnet** program, for example); the latter approach results in user passwords being sent over the network in the clear.

- *Remote authentication* to DCE from NFS clients is provided via the **dfs_login** command. With remote authentication, you allow users to issue the **dfs_login** command to authenticate themselves.

  Remote authentication requires additional configuration, but it provides a less burdensome and more secure approach to authentication. Configuration consists of installing and configuring the Gateway Server (**dfsgwd**) process on the Gateway Server machines, installing the **dfs_login** command (and the **dfs_logout** command) on the NFS clients, configuring Kerberos on the NFS clients, and configuring the remote authentication service on both the Gateway Server machines and the NFS clients. However, authentication requires no administrative measures, and user passwords are never sent in the clear.

The **dfsgw add** and **dfs_login** commands both result in authenticated access to DFS from an NFS client. To provide a user with authenticated access, each command obtains a ticket-granting ticket (TGT) for the user from the DCE Security Service. The TGT is used to create a valid login context for the user. The login context includes a Process Activation Group (PAG), which DFS stores in the kernel of the Gateway

Server machine. The PAG identifies the user's TGT; the TGT serves as the user's DCE credentials.

On the Gateway Server machine, an association is created between the UNIX user identification number (UID) of the user and the network address of the NFS client from which DFS access is desired. A mapping is then created between this pair and the PAG created for the user. The mapping is stored as an entry in a local authentication table (AT), which, like the PAG, resides in the kernel of the machine. The mapping provides the user with authenticated access to DFS from the NFS client.

Each mapping grants a user authenticated access only from the specific NFS client for which the mapping exists. For authenticated access from a different NFS client, a user must use the **dfsgw add** or **dfs_login** command to create a new mapping for that client.

A user's DCE credentials are good only for the lifetime of the TGT. The ticket lifetime is dictated by the registry database of the DCE cell. By default, each ticket receives the default ticket lifetime in effect in the registry database. The **dfs_login** command includes a **-l** option that can be used to request a different lifetime, but a requested lifetime is constrained by the policies in effect in the registry database. Once a user's TGT expires, the user must obtain new DCE credentials.

A user who wants to cancel authenticated access to DFS before the credentials expire can issue either the **dfs_logout** command from the NFS client for which the credentials were granted or the **dfsgw delete** command from the Gateway Server machine. Both commands remove the user's entry for the NFS client from the authentication table on the Gateway Server machine. Either command can be used to end the authenticated session, regardless of which command was used to obtain the credentials. Because the authentication table resides in memory, all authenticated sessions are terminated if the machine configured as a Gateway Server is rebooted.

The following sections provide complete instructions for configuring Gateway Server machines and NFS clients to provide NFS users with either local or remote authentication to DCE. The final section in this appendix provides detailed information about how users authenticate to DCE and how they access DFS from an NFS client.

A.1 Configuring Gateway Server Machines

A Gateway Server machine provides authenticated access to the DFS filespace to users on NFS clients. You can configure any machine that is configured as a DFS client and an NFS server as a Gateway Server. Following successful configuration, the machine provides authenticated access to the DFS filespace, and it exports the root of the DCE namespace, **/...**, via NFS.

You can configure multiple Gateway Server machines to provide DFS access from multiple sources. However, users do not randomly select Gateway Server machines from NFS clients. By default, users on an NFS client contact the Gateway Server machine that exports **/...** to the client. If you want to balance the load among multiple Gateway Servers, you must configure your NFS clients so that each client mounts **/...** from a different Gateway Server machine. (Section A.2 provides information about configuring NFS clients.)

Depending on how closely you want to control access to the DFS filespace, configure your Gateway Server machines in one of the following ways:

- Configure the Gateway Server machines so that users *cannot* issue the **dfs_login** command to authenticate to DCE.

  This configuration allows system administrators to manage all DCE authentication from the Gateway Server machines. You can allow users to issue the **dfsgw add** command themselves, or you can limit use of the command to administrators only. To configure a Gateway Server machine without enabling remote authentication via the **dfs_login** command, follow the instructions in Section A.1.1.

- Configure the Gateway Server machines so that users *can* issue the **dfs_login** command to authenticate to DCE.

  This configuration allows users of NFS clients to acquire their own DCE credentials from the NFS clients. To configure a Gateway Server machine and enable remote authentication via the **dfs_login** command, follow the instructions in Section A.1.2.

Before configuring a Gateway Server machine, you must do the following:

- Configure a DCE cell that includes DFS.
- Configure each machine that is to become a Gateway Server as a DFS client and an NFS server.

- Ensure proper synchronization among the system clocks on machines that are to become Gateway Servers, machines configured as NFS clients that are to contact the Gateway Servers, and machines in the DCE cell to be contacted. You must keep the system clocks on these machines synchronized at all times.

Once you have met these prerequisites, you can configure your Gateway Server machines.

A.1.1 Configuring a Gateway Server Without Enabling Remote Authentication

Perform the steps in this section to enable DCE authentication from a Gateway Server machine without enabling it from NFS clients that contact the Gateway Server. Users can authenticate only by issuing the **dfsgw add** command on the Gateway Server machine (or by having a system administrator issue the command for them, if administrators control authentication to the DCE cell).

To allow users of NFS clients to authenticate to DCE from the Gateway Server machine but not from NFS clients that contact the Gateway Server, perform the following steps on the machine to be configured as a Gateway Server:

1. Log in as the local **root** user on the machine.
2. Install the binary file for the **dfsgw** command suite in the directory */dcelocal/* **bin** on the machine. The **dfsgw** command suite provides a local interface to the authentication table maintained on the Gateway Server machine. Commands in the **dfsgw** suite can be used to add, delete, and view mappings in the authentication table. (See Sections A.3.2.2 through A.3.2.4 for information about using these commands.)
3. Export the DCE global root directory, **/...**, via NFS. This is typically accomplished via the **exportfs** command; the exact command and procedure depends on your vendor's implementation of NFS. (See your vendor's NFS documentation for more information.)

The Gateway Server machine is now configured to provide DCE authentication via only the **dfsgw add** command. Repeat these steps on each DFS client that is to be configured as a Gateway Server in this manner. Should you later decide to allow users to authenticate to DCE from NFS clients that contact the Gateway Server, simply perform the steps in Section A.1.2 on the Gateway Server machine.

A.1.2 Using dce_config to Configure the Gateway Server and Enable Remote Authentication

To configure a DFS/NFS Secure Gateway by using **dce_config**, and get the default results, perform the following steps. If you want to manually configure the Gateway and make specific changes, go to Section A.1.3.

On each server machine:

1. Log in as the local **root** user on the machine.
2. Install the binary file for the **dfsgwd** process in the directory *dcelocal*/**bin**. The **dfsgwd** process provides users of NFS clients with a remote interface to the authentication table maintained on the Gateway Server machine.
3. Use the **exportfs command to export the DCE global root** directory (**/...**) via NFS.
4. Add the **dfsgw** service to the Internet services database. The **dfsgw** service provides the login facility for the DFS/NFS Secure Gateway. To add the service, do one of the following:

   - *If you use the* **/etc/services** *file in your environment*, add an entry for the **dfsgw** services to the **/etc/services** file.
   - *If you use a Network Information Service (NIS) services map in your environment*, add an entry for the **dfsgw** service to the NIS services map file on the NIS master. Add the entry to the services map for the first Gateway Server process only; do not add the entry for additional Gateway Server processes or NFS clients.

   In either case, you must add the following entry for the service:

   ```
   dfsgw     438/udp     dlog
   ```

   where **dfsgw** is the name of the service, **438** is the port at which the service receives RPCs, **udp** is the protocol the service uses to communicate, and **dlog** is an alias for the **dfsgw** service.

   Run **dce_config** as follows:

   - From the Main **DCE Configuration** menu, select **4** (DFS Client).

- Enter the cell administrator's name and password.
- Enter **y** at the prompt **Would you like to configure this client as an NFS Gateway?**
- Respond to the prompt **Would you like to use BOS server to monitor and administer the dfsgwd process?** as follows:

  If you want the **dfsgw** server administrator to be able to administer and monitor the **dfsgw** server via BOS commands, enter **y**. Otherwise, enter **n**.

Configuration of the node as an NFS/DFS Gateway Server is complete.

A.1.3 Manually Configuring a Gateway Server and Enabling Remote Authentication

Perform the steps in this section to enable DCE authentication either from a Gateway Server machine or from NFS clients that contact the Gateway Server. Users authenticate from the Gateway Server machine by issuing the **dfsgw add** command; they authenticate from an NFS client by issuing the **dfs_login** command. A Gateway Server machine to be configured in this manner runs the Gateway Server (**dfsgwd**) process. The steps in Section A.1.3.2 configure the **dfsgwd** process on the Gateway Server machine.

A Gateway Server machine configured in this way should also run the Basic OverSeer (BOS) Server to monitor and simplify administration of the **dfsgwd** process. The steps in Section A.1.2.1 configure a BOS Server (**bosserver**) process on the Gateway Server machine. Perform the steps in Section A.1.3.1 only if the BOS Server is not already running on the machine. (Note that you typically run the BOS Server only on DFS servers, but you can also run it on DFS clients. See Chapters 4 and 5 for more information about the BOS Server.)

A.1.3.1 Configuring the BOS Server Process

To configure the BOS Server (**bosserver**) process, perform the following steps on the machine to be configured as a Gateway Server. In all cases, *hostname* is the hostname of the local machine. (Note that you may need to install the **bosserver** binary file on

the machine if it is not already present. See your vendor's installation and configuration documentation for information about installing the binary file.)

1. Authenticate to DCE as a principal who has the following ACL permissions on entries in the registry database:

   - The **i** permission on the directory **hosts/***hostname*.
   - The **m**, **a**, **u**, **g**, and **c** permissions on the principal **hosts/***hostname***/dfs-server**. The principal is created during the configuration steps.
   - The **t** and **M** permissions on the group **subsys/dce/dfs-admin**.
   - The **R**, **t**, and **M** permissions on the organization **none**.
   - The **r** permission on the registry Policy object for the DCE cell.

   This requirement is most easily met by authenticating to a privileged DCE identity (for example, **cell_admin** or a principal who is a member of the group **acct-admin**).

2. Create the principal **hosts/***hostname***/dfs-server**, and create an account for the principal. Use the following **dcecp** commands to create the principal and account in the registry database. In the commands, *password* is the password of the DCE identity to which you are authenticated.

   $ dcecp
   dcecp> principal create hosts/*hostname***/dfs-serverdcecp> **
   account create hosts/*hostname***/dfs-server> **
   -group subsys/dce/dfs-admin > -org *none* ****
   -password*password* **-mypwd** *password*

3. Grant the group **subsys/dce/dfs-admin** the appropriate permissions on the ACL for the **hosts/***hostname***/dfs-server** principal in the registry database:

   dcecp> acl mod /.:/sec/principal/hosts/*hostname***/dfs-server> **
   -add *{group subsys/dce/dfs-admin rcDnfmag}*
   dcecp> exit

4. Use the **su** command to become the local **root** user on the machine:

```
$ su
Password: root_password
```

5. Add a server key for the **hosts**/*hostname*/**dfs-server** principal to the **/krb5/v5srvtab** keytab file on the machine. The **dced** process recognizes the keytab file by the entry name **self**. The command creates the keytab file if the file does not already exist. In the commands, *password* is the password of the DCE identity to which you were authenticated when you created the principal.

```
# dcecp
dcecp> keytab add self -member hosts/hostname/dfs-server \> -key password
dcecp> keytab add self -member hosts/hostname/dfs-server \> \
-random -registry
dcecp> exit
```

6. Remove the **BosConfig** file and any administrative lists that may exist from a previous configuration of the BOS Server on the machine:

```
# rm -f dcelocal/var/dfs/BosConfig
# rm -f dcelocal/var/dfs/admin.*
```

7. Start the **bosserver** process with DFS authorization checking disabled. The process creates a new **BosConfig** file and a new **admin.bos** file, which is the administrative list for the BOS Server.

```
# dcelocal/bin/bosserver -noauth &
```

8. Add the group **subsys/dce/dfs-admin** to the **admin.bos** file:

```
# dcelocal/bin/bos addadmin -server /.:/hosts/hostname -adminlist \
admin.bos -group subsys/dce/dfs-admin
```

9. Enable DFS authorization checking by the BOS Server:

```
# dcelocal/bin/bos setauth -server /.:/hosts/hostname -authchecking on
```

10. Configure the **bosserver** process to start automatically when the system is rebooted by removing the two **#** (number signs) from the following line of the **/etc/rc.dfs** file (or its equivalent):

```
##daemonrunning $DCELOCAL/bin/bosserver
```

The BOS Server process is now fully configured on the machine.

A.1.3.2 Configuring the Gateway Server Process

To configure the Gateway Server (**dfsgwd**) process, perform the following steps on the machine to be configured as a Gateway Server. The steps assume that the BOS Server is already running on the machine. In all of the steps, *hostname* is the hostname of the local machine.

Note: You need to perform some steps only when you configure the first Gateway Server process. Such steps are qualified with the phrase *for the first Gateway Server process*.

1. *If you have not already done so,* perform all of the steps in Section A.1.1 to install the **dfsgw** binary file on the machine and to export **/...** from the machine.
2. *If you have not already done so,* log in as the local **root** user on the machine.
3. Install the binary file for the **dfsgwd** process in the directory *dcelocal*/**bin** on the machine. The **dfsgwd** process provides users of NFS clients with a remote interface to the authentication table maintained on the Gateway Server machine.
4. Add the **dfsgw** service to the Internet services database. The **dfsgw** service provides the login facility for the DFS/NFS Secure Gateway. To add the service, do one of the following:
   - *If you use the* **/etc/services** *file* in your environment, add an entry for the **dfsgw** service to the **/etc/services** file on the machine.
   - *If you use a Network Information Service (NIS) services map* in your environment, add an entry for the **dfsgw** service to the NIS services map file on the NIS master. Add the entry to the services map only *for the first*

Gateway Server process; do not add the entry for additional Gateway Server processes or NFS clients.

In either case, you need to add the following entry for the service:

```
dfsgw     438/udp     dlog
```

where **dfsgw** is the name of the service, **438** is the port at which the service receives RPCs, **udp** is the protocol the service uses to communicate, and **dlog** is an alias for the **dfsgw** service. See the reference page for the **services** file for more information.

5. Authenticate to DCE as a principal who has the following ACL permissions on entries in the registry database:
   - The **i** permission on the directory **hosts/***hostname*.
   - *For the first Gateway Server process*, the **i** permission on the directory **subsys/dce**.
   - The **m**, **a**, **u**, and **g** permissions on the principal **hosts/***hostname***/dfsgw-server**. The principal is created during the configuration steps.
   - The **t** and **M** permissions on the group **subsys/dce/dfsgw-admin**. The group is created during the configuration steps.
   - The **R**, **t**, and **M** permissions on the organization **none**.
   - The **r** permission on the registry Policy object for the DCE cell.

   This requirement is most easily met by authenticating to a privileged DCE identity (for example, **cell_admin** or a principal who is a member of the group **acct-admin**).
6. Invoke the **dcecp** command:

   ```
   $ dcecp
   ```

7. *For the first Gateway Server process*, create the group **subsys/dce/dfsgw-admin** in the registry database. Use the following **dcecp** command to create the group:

```
dcecp> group create subsys/dce/dfsgw-admin
```

8. Create the principal **hosts/***hostname*/**dfsgw-server**, and create an account for the principal. The Gateway Server process communicates as the principal **hosts/***hostname*/**dfsgw-server**. Use the following **dcecp** commands to create the principal and account in the registry database. In the commands, *password* is the password of the DCE identity to which you are authenticated.

```
dcecp> principal create hosts/hostname/dfsgw-serverdcecp> \
account create hosts/hostname/dfsgw-server> \
-group subsys/dce/dfsgw-admin -org none -password password >  \
-mypwd password
dcecp> exit
```

9. Use the **su** command to become the local **root** user on the machine:

```
$ su
Password: root_password
```

10. Add a server key for the **hosts/***hostname*/**dfsgw-server** principal to the **/krb5/v5srvtab** keytab file on the machine. The **dced** process recognizes the keytab file by the entry name **self**. In the commands, *password* is the password of the DCE identity to which you were authenticated when you created the principal.

```
# dcecp
dcecp> keytab add self -member hosts/hostname/dfsgw-server \> \
-key password
dcecp> keytab add self -member hosts/hostname/dfsgw-server \> \
-random -registry
dcecp> exit
```

11. Log out as **root** to return to your authenticated DCE identity.
12. If your current DCE identity is not included in the */dcelocal/***var/dfs/admin.bos** file on the machine, either add the identity to the file or authenticate to DCE as a

principal who is included in the file. You can use the **bos lsadmin** command to list the principals and groups included in the **admin.bos** file:

```
$ dcelocal/bin/bos lsadmin -server /.:/hosts/hostname \
-adminlist admin.bos
```

13. Create a **simple** BOS Server process named **dfsgw** to run the **dfsgwd** server process:

```
$ dcelocal/bin/bos create -server /.:/hosts/hostname -process dfsgw \
-type simple -cmd dcelocal/bin/dfsgwd
```

The Gateway Server process is now fully configured on the machine.

A.2 Configuring NFS Clients to Access DFS

Once you have configured at least one Gateway Server machine according to the instructions in Section A.1, you can configure your NFS clients to provide access to the DFS filespace. Users who have DCE accounts can then authenticate to DCE for authenticated access to DFS from the NFS clients. Authenticating to DCE provides these users with the privileges and permissions associated with their DCE identities.

Depending on how you configured your Gateway Server machines, configure each NFS client that is to provide access to DFS in one of the following ways:

- If you configured your Gateway Servers so that users *cannot* issue the **dfs_login** command to authenticate to DCE, configure your NFS clients without enabling DCE authentication via the **dfs_login** command; follow the instructions in Section A.2.1.
- If you configured your Gateway Servers so that users *can* issue the **dfs_login** command to authenticate to DCE, configure your NFS clients and enable DCE authentication via the **dfs_login** command; follow the instructions in Section A.2.2.

Because the steps in each of these sections mount **/...** on an NFS client, users who do not have DCE accounts can still use the NFS client for unauthenticated access to DFS. (See Section A.3.1 for more information about unauthenticated access; see Section A.3.2 for more information about authenticated access.)

A.2.1 Configuring a Client Without Enabling Remote Authentication

If you configured your Gateway Server machines so that users cannot issue the **dfs_login** command to authenticate to DCE, perform the steps in this section to configure your NFS clients. The steps enable DFS access from an NFS client without enabling DCE authentication from the client. Users can authenticate only via the **dfsgw add** command.

To provide users of an NFS client with access to DFS but not the **dfs_login** command, perform the following steps on the client:

1. Log in as the local **root** user on the machine.
2. Mount the root of the DCE namespace, **/...**, on the machine. In the command, *hostname* is the hostname of a machine that exports **/...**. Each machine configured as a Gateway Server exports **/...**. When users access DFS from an NFS client, they go through the Gateway Server machine that exports **/...** to the client. To achieve proper load balancing if you configure multiple Gateway Server machines, ensure that the mounts of **/...** on your NFS clients are divided evenly among your Gateway Servers. (You can use the NFS automount mechanism with a direct automount map to mount **/...**; see your vendor's NFS documentation for more information.)

   ```
   # mkdir /...
   # mount hostname:/... /...
   ```

3. Create a symbolic link from **/:** to the root of the DFS filespace for the host DCE cell, **/...**/*cellname*/**fs**. In the command, *cellname* is the name of the DCE cell to be accessed from the NFS client (the cell in which the machine that exports **/...** is configured as a DFS client).

```
# ln -s /.../cellname/fs /:
```

4. Verify that the NFS mount of DCE was successful by using the **ls** command to list the contents of **/:**, which leads to the root directory of the DFS filespace. The command should yield the same output from the NFS client that it does from a DFS client of the DCE cell.

```
# ls /:
```

The NFS client is now configured to provide access to DFS but not to allow users of the client to authenticate to DCE with the **dfs_login** command. Repeat these steps on each NFS client to be configured in this manner. Should you later decide to allow users to authenticate to DCE from the NFS client, simply perform the steps in Section A.2.2 on the client.

A.2.2 Configuring a Client and Enabling Remote Authentication

If you configured your Gateway Server machines so that users can issue the **dfs_login** command to authenticate to DCE, perform the steps in this section to configure your NFS clients. The steps enable both DFS access and DCE authentication from an NFS client. Users can authenticate via either the **dfsgw add** command or the **dfs_login** command.

To provide users of an NFS client with access to both DFS and the **dfs_login** command, perform the following steps on the client:

1. *If you have not already done so*, perform all of the steps in Section A.2.1 to mount **/...** on the machine.
2. *If you have not already done so*, log in as the local **root** user on the machine.
3. Install the binary files for the **dfs_login** and **dfs_logout** commands in the directory **/usr/bin** on the machine. These commands provide the following functionality:

   dfs_login Allows users of the NFS client to establish an authenticated session by obtaining DCE credentials on a Gateway Server machine. (See Section A.3.2.1 for information about using this command.)

dfs_logout Allows users on the NFS client to end an authenticated session established with the **dfs_login** command. (See Section A.3.2.1 for information about using this command.)

The **dfs_login** and **dfs_logout** commands use version 5 of Kerberos to communicate with the DCE Security Service.

1. Create the Kerberos configuration file named **/krb5/krb.conf**. The **dfs_login** command reads this file to determine the name of a DCE Security Server that it can contact. This file must be identical to the **/krb5/krb.conf** file on machines in the host DCE cell; copy it from a machine in the DCE cell.

2. Create the Kerberos configuration file named **/krb5/krb.realms**. The Kerberos runtime uses the information in this file to translate Internet domains to the corresponding Kerberos realms. In the file, the Kerberos realm has the same name as the DCE cell. Each line of the file must have the following format:

```
domain  krb-realm
```

   where *domain* is the name of the local Internet domain, and *krb-realm* is the name of the Kerberos realm (the name of the DCE cell to be accessed). For example, in the following **krb.realms** file, **def.com** is the name of the Internet domain, and *abc.com* is the name of the DCE cell. If machines from multiple domains are to contact the DCE cell, you need a separate line for each domain. Note that realm names are case-sensitive.

```
.DEF.COM  abc.com
```

3. *If you use the* **/etc/services** *file* in your environment, add the following entry for the **dfsgw** service to the **/etc/services** file on the machine:

```
dfsgw     438/udp     dlog
```

   where **dfsgw** is the name of the service, **438** is the port at which the service receives RPCs, **udp** is the protocol the service uses to communicate, and **dlog** is an alias for the **dfsgw** service.

If you use an NIS services map in your environment, you added an entry to the services map file when you configured the first Gateway Server process. You do *not* need to add the entry to the services map when you configure NFS clients.

The NFS client is now configured to provide access to DFS and to allow users of the client to authenticate to DCE with the **dfs_login** command. Repeat these steps on each NFS client to be configured in this manner.

A.3 Accessing DFS from an NFS Client

Once a Gateway Server machine and one or more NFS clients are configured according to the instructions in Sections A.1 and A.2, users of the NFS clients can access data in the DFS filespace. Users can access files and directories in DFS by full **/.../***cellname*/**fs** pathnames or by abbreviated pathnames that use the **/:** link to the DFS filespace. The following are equivalent pathnames for the file **myfile** in the DFS filespace of the DCE cell *abc.com*:

/.../abc.com/fs/myfile

/:/myfile

All users have unauthenticated access to DFS. Users who have DCE accounts can authenticate to their DCE identities for authenticated access to DFS. The following subsections provide more information about these two types of access.

When accessing DFS data from a DFS client, the DFS Cache Manager caches local copies of files accessed from File Server machines. When accessing DFS data from an NFS client, NFS background I/O daemons cache local copies of files accessed via the NFS server. The caching of information by the NFS daemons can affect how quickly changes you make to data in DFS become visible to other users.

A.3.1 Unauthenticated Access to DFS

Unauthenticated access is provided to users who access DFS without first authenticating to DCE. For a user who does not have an account in the DCE registry database, unauthenticated access is the only form of access available. Unauthenticated access requires no preliminary steps; users simply access data in DFS from an NFS client.

Unauthenticated users receive the following permissions for objects (files and directories) in the DFS filespace:

- *For objects in nonLFS filesets*, unauthenticated users receive the permissions granted by the **other** mode bits of the object.
- *For objects in DCE LFS filesets*, unauthenticated users receive the permissions granted by the **any_other** entry, if it exists, on the ACL of the object. The **mask_obj** entry filters permissions granted via the **any_other** entry.

When an unauthenticated user creates an object, the object is owned by the user **nobody** and the group **nogroup**. The UID of the user **nobody** is **-2**, and the GID of the group **nogroup** is also **-2**. (Note that identities and ID numbers of an unauthenticated user and group can vary between systems; see your vendor's documentation for more information.)

Unauthenticated access is provided with the DFS/NFS Secure Gateway as a side effect of configuring Gateway Server machines and NFS clients. Unauthenticated access is available without the DFS/NFS Secure Gateway. Simply export **/...** from a DFS client that is also an NFS Server, and mount **/...** on each NFS client from which users are to access DFS.

A.3.2 Authenticated Access to DFS

Authenticated access is available to users who have accounts in the DCE cell. When an authenticated user accesses an object in the DFS filespace, the user receives the permissions associated with the DCE identity to which the user is authenticated. When the user creates an object, the object is owned by the user and the user's primary group.

To authenticate to DCE, you can issue either of the following commands, both of which establish credentials recognized by the DCE Security Service:

- From an NFS client, enter the **dfs_login** command. (See Section A.3.2.1.)
- From a Gateway Server machine, enter the **dfsgw add** command. (See Section A.3.2.2.)

A user who desires authenticated access to DFS must have a principal and account in the registry database of the DCE cell. An entry must exist for the user in the **/etc/passwd** file on the machine configured as a Gateway Server and on each NFS client from which the user is to access DCE. The user's UID in the **/etc/passwd** file must match the user's UID in the DCE registry database. (On a DCE client, the **passwd_export** command can be used to keep **/etc/passwd** files current with respect to the registry database; see the *DCE 1.2.2 Administration Guide—Core Components* for more information.)

The **dfs_login** and **dfsgw add** commands do not obtain a new TGT if you already have a valid TGT in your current login context and you do not request DCE credentials for a different user. However, the commands do allow you to use your existing TGT to establish authenticated access to DFS from additional NFS clients. If you do not already have an entry in the authentication table for an NFS client from which you request authenticated access, the commands create a new entry for you, using the existing TGT as the basis of the new entry; if you already have an entry in the authentication table for the NFS client, the commands have no effect. In either case, the commands do not affect existing entries in the authentication table, and they do not affect the remaining ticket lifetime of your existing TGT.

DCE credentials (tickets) expire after the lifetime specified by the DCE Security Service. Once they expire, the tickets can no longer be used for authenticated access. To end an authenticated session before the ticket lifetime has passed, you can issue either of the following commands:

- From the NFS client from which authenticated access to DFS is provided, enter the **dfs_logout** command. (See Section A.3.2.1.)
- From the Gateway Server machine via which DFS is accessed, enter the **dfsgw delete** command. (See Section A.3.2.2.)

Both commands remove the entry from the authentication table that provides authenticated access from the NFS client. Regardless of which command you used to establish the DCE credentials (**dfs_login** or **dfsgw add**), you can end the authenticated session with the **dfs_logout** command or the **dfsgw delete** command. Neither command affects authenticated access from other NFS clients. If your DCE credentials

are the basis of another entry in the authentication table, you still have authenticated access via that other entry.

To refresh your DCE credentials before they expire, use the **kinit** command to obtain new credentials, then use the **dfs_login** or **dfsgw add** command to replace your existing TGT with the new TGT. This procedure provides you with authenticated access to DFS for the ticket lifetime of your new TGT. If you do not have access to the **kinit** command, you cannot refresh your DCE credentials.

Note that if you configure multiple Gateway Server machines, each server machine houses its own authentication table. The **dfs_login** and **dfs_logout** commands affect entries only in the authentication table maintained on the Gateway Server machine they contact; commands in the **dfsgw** suite affect entries only in the authentication table on the machine on which they are issued.

A.3.2.1 Authenticating to DCE from an NFS Client

The **dfs_login** command authenticates a user to DCE from an NFS client. The command contacts the DCE Security Service to obtain a TGT and a service ticket for the Gateway Server (**dfsgwd**) process for the user. It encrypts the user's TGT with the service ticket and sends these to the Gateway Server process. It also sends the UID of the user who issues the command and the network address of the NFS client from which the command is issued. The Gateway Server process uses this information to create a valid login context, including a PAG, and an entry in the authentication table for the user.

The syntax of the **dfs_login** command follows:

dfs_login [-h *hostname***] [-l** *hh* **[:***mm* **]] [***dce_principal***] [***dce_password***]**

The command includes the following options and arguments:

-h *hostname*

Specifies the hostname of the Gateway Server machine. By default, the command uses the hostname of the machine that exports **/...** to the NFS client. Use this option to contact a different Gateway Server.

-l *hours*[**:***minutes*]

Specifies the lifetime to be assigned to the service ticket obtained with the command. Enter the lifetime as a number of hours and, optionally, minutes. A value specified with this option is subject to the policies in effect in the registry database of the DCE cell. By default, the ticket is assigned the default lifetime assigned to tickets in the DCE cell.

dce_principal

Specifies the DCE principal name of the user who is to be logged into DCE. By default, the command uses the name of the issuer of the command.

dce_password

Provides the DCE password of the specified user. If you do not specify a password, the command prompts for a password if one of the following is true: You name a user other than yourself, you name yourself and you do not already have a valid TGT, or you do not name a user and you do not already have a valid TGT. The command does not prompt for a password if you do not name a different user and you already have a valid TGT.

For example, the user named **ludwig** issues the following **dfs_login** command to authenticate to DCE from an NFS client:

```
$ dfs_login

Password for ludwig@abc.com: password
```

where *password* is the DCE password of the user **ludwig**. In the example, the user **ludwig** does not already have a valid TGT, so the command prompts for the user's password and obtains a TGT for the user. If the login succeeds, the **dfs_login** command returns no messages.

To end the authenticated session before the DCE credentials expire, issue the **dfs_logout** command from the NFS client. The command removes the user's entry from the authentication table on the Gateway Server machine. The command can be issued either by the user whose entry is to be removed from the authentication table

or by a user who is logged into the NFS client as the local **root** user. The command has no effect on authenticated access the user may have from other NFS clients.

The syntax of the **dfs_logout** command follows:

dfs_logout [-h *hostname***] [***dce_principal***]**

The command includes the following option and argument:

-h *hostname* Specifies the hostname of the Gateway Server machine. By default, the command uses the hostname of the machine that exports **/...** to the NFS client. Use this option to contact a different Gateway Server.

dce_principal
Specifies the DCE principal name of the user whose entry is to be removed from the authentication table. By default, the command deletes the entry for the user who issues the command.

For example, the following ends the authenticated session of the issuer of the command:

```
$ dfs_logout
```

See the reference pages for the **dfs_login** and **dfs_logout** commands for detailed information about the use and syntax of the commands.

A.3.2.2 Authenticating to DCE from a Gateway Server Machine

The **dfsgw add** command authenticates a user to DCE from a Gateway Server machine. Users can use the **dfsgw add** command if the **dfs_login** command is not installed on the NFS client from which they desire access to DFS. System administrators can use the command to administer authenticated access to DFS from a Gateway Server machine. Note that for NFS clients not configured to enable DCE authentication, the **dfsgw add** command represents the only avenue to DCE authentication.

The **dfsgw add** command provides essentially the same functionality as the **dfs_login** command. However, unlike the **dfs_login** command, the **dfsgw add** command does not communicate with the Gateway Server (**dfsgwd**) process; it creates the login context and entry in the authentication table for the user. In addition, it requires the issuer to identify the user for whom authenticated access is desired and the NFS client from which the user is to access DFS. Also, the **dfs_login** command allows the issuer to request a ticket lifetime; the **dfsgw add** command does not.

The **dfsgw add** command has the following syntax:

dfsgw add -id *networkID:userID* **[-dceid** *login_name***[**:*password***]]** \
[-af *address_family***]**

The command includes the following options:

-id *networkID:userID*
: Specifies the network address or hostname of an NFS client and the UID of the user who is to be authenticated to DCE from that client.

-dceid *login_name*[*:password*]
: Specifies the DCE principal name and, optionally, the password of the user who is to be authenticated to DCE. The command does not prompt for a principal name and password if you do not specify a principal name and you have a valid TGT; the command does not prompt for a password if you specify your own principal name and you have a valid TGT. The command always prompts for a password if you name a principal other than yourself.

-af *address_family*
: Specifies the style of network address to be used to identify hosts. By default, the command uses the only address family currently supported, **inet** (Internet).

For example, the following **dfsgw add** command obtains DCE credentials for the user **ludwig**, who has UID **7439**, from the NFS client that has network address **15.27.32.40**:

```
$ dfsgw add -id 15.27.32.40:7439 -dceid ludwig
```

```
Enter Password: password
Mapping added successfully, PAG is 41ffffe4
```

where *password* is the DCE password of the user **ludwig**. The command reports that a mapping for the user was successfully added to the authentication table on the Gateway Server machine; the user's PAG is **41ffffe4**.

To end a user's authenticated session from a specified NFS client, issue the **dfsgw delete** command on the Gateway Server machine. The command provides the same functionality from a Gateway Server machine that the **dfs_logout** command provides from an NFS client. The **dfsgw delete** command can be issued either by the user whose entry is to be removed from the authentication table or by a user who is logged into the Gateway Server machine as the local **root** user. The command has no effect on authenticated sessions the user may have for other NFS clients.

The syntax of the **dfsgw delete** command follows:

dfsgw delete -id *networkID:userID* **[-af** *address_family***]**

The command includes the following options:

-id *networkID:userID*
: Specifies the network address or hostname of an NFS client and the UID of the user whose authenticated access from that client is to be canceled.

-af *address_family*
: Specifies the style of network address to be used to identify hosts. By default, the command uses the only address family currently supported, **inet** (Internet).

For example, the following **dfsgw delete** command ends the authenticated session for the user **ludwig** from the NFS client that has network address **15.27.32.40**. The command is issued by the local **root** user on the Gateway Server machine.

 # dfsgw delete -id 15.27.32.40:7439

See the reference pages for the **dfsgw add** and **dfsgw delete** commands for detailed information about the use and syntax of the commands.

A.3.2.3 Determining Whether a Specific User Is Authenticated to DCE

The **dfsgw query** command determines whether a specific user is authenticated to DCE via the Gateway Server machine. The command can be issued either by the user whose authentication is to be determined or by a user who is logged in as the local **root** user on the machine configured as a Gateway Server.

The command looks for an entry for the user in the authentication table on the Gateway Server machine on which it is issued. If your environment includes multiple Gateway Server machines, you must issue the command on the Gateway Server machine whose authentication table is to be examined. The command displays information about a user's entry regardless of whether the user authenticated via the **dfs_login** command or the **dfsgw add** command.

The **dfsgw query** command has the following syntax:

dfsgw query -id *networkID:userID* **[-af** *address_family***]**

The command includes the following options:

-id *networkID:userID*
: Specifies the network address or hostname of an NFS client and the UID of the user whose DCE authentication for that client is to be determined.

-af *address_family*
: Specifies the style of network address to be used to identify hosts. By default, the command uses the only address family currently supported, **inet** (Internet).

For example, the following **dfsgw query** command determines whether the user **ludwig** is authenticated from the NFS client that has network address **15.27.32.40**. The command is issued by the local **root** user on the Gateway Server machine.

```
# dfsgw query -id 15.27.32.40:7439


Mapping found, PAG is 41ffffe4
```

The command reports that a mapping for the user for the specified NFS client exists in the local authentication table. The user's PAG is **41ffffe4**. See the reference page for the **dfsgw query** command for more information about the command.

A.3.2.4 Displaying Information About All Users Who Are Authenticated to DCE

The **dfsgw list** command lists all users who are authenticated to DCE via the Gateway Server machine. The command lists all entries from the authentication table on the Gateway Server machine on which it is issued. If your environment includes multiple Gateway Server machines, you must issue the command on the Gateway Server machine from whose authentication table entries are to be displayed. The command makes no distinction between entries created with the **dfs_login** command and entries created with the **dfsgw add** command. No privileges are required to issue the command.

The **dfsgw list** command has the following syntax:

```
dfsgw list
```

For example, the following **dfsgw list** command displays all of the users currently authenticated to DCE (all users who have entries in the authentication table):

```
$ dfsgw list
Mapping: nfs1.abc.com : ludwig => 41ffffe4 Expires at \
Sat Jul 23 01:33:18 1994
Mapping: nfs2.abc.com : frost => 41ffffa3 Expires at \
Sat Jul 23 08:36:23 1994
Mapping: nfs2.abc.com : wvh => 41ffffbe Expires at \
Sun Jul 24 00:51:21 1994
```

. . .

Note that the **dfsgw list** command provides additional information not available with the **dfsgw query** command, such as the hostname of the NFS client from which each user has DFS access, the principal name of each user who has DFS access, and the date and time at which each user's DCE credentials expire. See the reference page for the **dfsgw list** command for more information about the command.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W