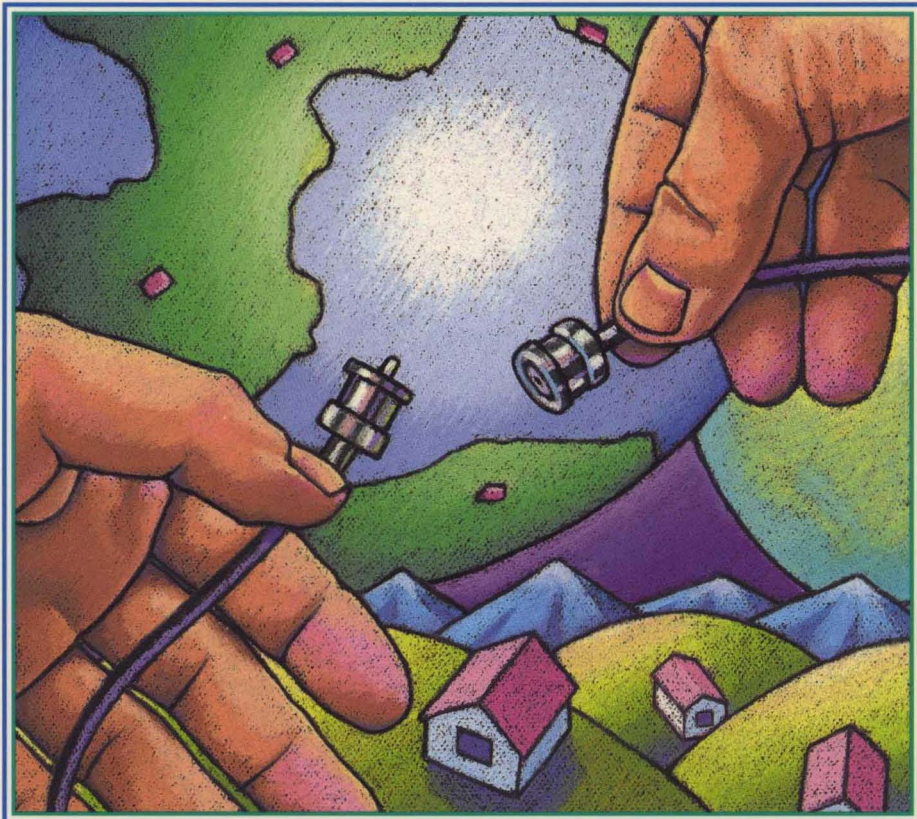


OSF™ DCE

OSF™ DCE Administration Reference



OPEN SOFTWARE FOUNDATION

OSF™ DCE Administration Reference

Revision 1.0

Open Software Foundation



P T R Prentice Hall, Englewood Cliffs, New Jersey 07632

Cover design: **BETH FAGAN**
Cover illustration: **STEVE LEWONTIN**

This book was formatted with troff



Published by Prentice-Hall, Inc.
A Simon & Schuster Company
Englewood Cliffs, New Jersey 07632

The information contained within this document is subject to change without notice.

OSF MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

OSF shall not be liable for errors contained herein, or for any direct or indirect, incidental, special or consequential damages in connection with the furnishing, performance, or use of this material.

Copyright ©1993 Open Software Foundation, Inc.

This documentation and the software to which it relates are derived in part from materials supplied by the following:

- © Copyright 1990, 1991 Digital Equipment Corporation
- © Copyright 1990, 1991 Hewlett-Packard Company
- © Copyright 1989, 1990, 1991 Transarc Corporation
- © Copyright 1990, 1991 Siemens Nixdorf Informationssysteme AG
- © Copyright 1990, 1991 International Business Machines Corporation
- © Copyright 1988, 1989 Massachusetts Institute of Technology
- © Copyright 1988, 1989 The Regents of the University of California

All rights reserved.
Printed in the U.S.A.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN: 0-13-643818-0

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

THIS DOCUMENT AND THE SOFTWARE DESCRIBED HEREIN ARE FURNISHED UNDER A LICENSE, AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. TITLE TO AND OWNERSHIP OF THE DOCUMENT AND SOFTWARE REMAIN WITH OSF OR ITS LICENSORS.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc.

UNIX is a registered trademark of UNIX System Laboratories, Inc. in the U.S. and other countries.

DEC, DIGITAL, and ULTRIX are registered trademarks of Digital Equipment Corporation.

DECstation 3100 is a trademark of Digital Equipment Corporation.

HP, Hewlett-Packard, and LaserJet are trademarks of Hewlett-Packard Company.

Network Computing System and PasswdEtc are registered trademarks of Hewlett-Packard Company.

AFS and Transarc are registered trademarks of the Transarc Corporation.

Episode is a trademark of the Transarc Corporation.

AIX and RISC System/6000 are trademarks of International Business Machines Corporation.

IBM is a registered trademark of International Business Machines Corporation.

DIR-X is a trademark of Siemens Nixdorf Informationssysteme AG.

NFS, Network File System, SunOS and Sun Microsystems are trademarks of Sun Microsystems, Inc.

X/Open is a trademark of the X/Open Company Limited in the U.K. and other countries.

PostScript is a trademark of Adobe Systems Incorporated.

FOR U.S. GOVERNMENT CUSTOMERS REGARDING THIS DOCUMENTATION AND THE ASSOCIATED SOFTWARE.

These notices shall be marked on any reproduction of this data, in whole or in part.

NOTICE: Notwithstanding any other lease or license that may pertain to, or accompany the delivery of, this computer software, the rights of the Government regarding its use, reproduction and disclosure are as set forth in Section 52.227-19 of the FARS Computer Software-Restricted Rights clause.

RESTRICTED RIGHTS NOTICE: Use, duplication, or disclosure by the Government is subject to the restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013.

RESTRICTED RIGHTS LEGEND: Use, duplication or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the rights in Technical Data and Computer Software clause in DAR 7-104.9(a). This computer software is submitted with "restricted rights." Use, duplication or disclosure is subject to the restrictions as set forth in NASA FAR SUP 18-52.227-79 (April 1985) "Commercial Computer Software-Restricted Rights (April 1985)." If the contract contains the Clause at 18-52.227-74 "Rights in Data General" then the "Alternate III" clause applies.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract.

Unpublished - All rights reserved under the Copyright Laws of the United States.

This notice shall be marked on any reproduction of this data, in whole or in part.

Contents

Preface	ix
Audience	ix
Applicability	x
Purpose	x
Document Usage	x
Related Documents	x
Typographic and Keying Conventions	xi
Pathnames of Directories and Files in DCE	
Documentation	xii
Problem Reporting	xii
Permutated Index	xiii
Chapter 1. Remote Procedure Call Commands	1-1
intro	1-2
rpccp	1-4
add element	1-15
add entry	1-18
add member	1-20
export	1-22
help	1-26
import	1-28
remove element	1-31
remove entry	1-34
remove group	1-35
remove mapping	1-36
remove member	1-40
remove profile	1-42
show entry	1-43
show group	1-45
show mapping	1-47
show profile	1-50

show server	1-54
unexport	1-56
rpcd	1-59
Chapter 2. Directory Service Commands	2-1
intro	2-2
add directory	2-4
add object	2-6
cdsadv	2-8
cdsbrowser	2-10
cdsclerk	2-11
cdscp	2-13
cdsd	2-22
clear cached server	2-24
clear clearinghouse	2-25
create child	2-26
create clearinghouse	2-27
create directory	2-29
create link	2-30
create object	2-32
create replica	2-33
define cached server	2-34
delete child	2-36
delete clearinghouse	2-37
delete directory	2-39
delete link	2-40
delete object	2-41
delete replica	2-42
disable clerk	2-43
disable server	2-44
dump clerk cache	2-45
gdad	2-46
list child	2-48
list clearinghouse	2-50
list directory	2-52
list link	2-54
list object	2-56
remove directory	2-58
remove link	2-60
remove object	2-61
set cdscp confidence	2-63
set cdscp preferred clearinghouse	2-64
set directory	2-65
set directory to new epoch	2-67
set directory to skulk	2-69
set link	2-70
set object	2-72
show cached clearinghouse	2-74

show cached server	2-76
show cdscp confidence	2-78
show cdscp preferred clearinghouse	2-79
show cell	2-80
show child	2-82
show clearinghouse	2-84
show clerk	2-87
show directory	2-89
show link	2-93
show object	2-95
show replica	2-97
show server	2-101
intro	2-104
gdscacheadm	2-105
gdscmxl	2-107
gdsdirinfo	2-110
gdsditadm	2-113
gdsstep	2-119
gdssysadm	2-121
Chapter 3. Distributed Time Service Commands	3-1
intro	3-2
advertise	3-3
change	3-4
create	3-5
delete	3-6
disable	3-7
dtscp	3-8
dtscd	3-11
enable	3-13
exit	3-15
help	3-16
quit	3-17
set	3-18
show	3-22
synchronize	3-30
unadvertise	3-31
update	3-32
Chapter 4. Security Service Commands	4-1
intro	4-2
passwd_override	4-3
v5srvtab	4-7
intro	4-8
passwd_export	4-9
passwd_import	4-11
rgy_edit	4-15
sec_admin	4-27

sec_clientd	4-29
sec_create_db	4-30
secd	4-32
Chapter 5. Distributed File Service Commands	5-1
intro	5-2
BakLog	5-4
BosConfig	5-5
BosLog	5-8
CacheInfo	5-9
CacheItems	5-11
DFSLog	5-12
FMSLog	5-14
FileLog	5-16
FilesetItems	5-17
FILog	5-18
FtLog	5-19
NoAuth	5-20
RepLog	5-22
SalvageLog	5-23
TE	5-24
TL	5-26
TapeConfig	5-28
UpLog	5-30
Vn	5-31
admin.bak	5-33
admin.bos	5-35
admin.fl	5-37
admin.ft	5-39
admin.up	5-41
dfsatab	5-43
dfstab	5-45
intro	5-47
bak	5-51
bak adddump	5-56
bak addftentry	5-59
bak addftfamily	5-63
bak addhost	5-65
bak apropos	5-67
bak deletedump	5-68
bak dump	5-70
bak dumpinfo	5-74
bak ftinfo	5-76
bak help	5-78
bak labeltape	5-80
bak lsdumps	5-82
bak lsftfamilies	5-85
bak lshosts	5-86

bak readlabel	5-88
bak restoredb	5-90
bak restoredisk	5-92
bak restoreft	5-95
bak rmdump	5-99
bak rmftentry	5-100
bak rmftfamily	5-101
bak rmhost	5-102
bak savedb	5-104
bak scantape	5-105
bak setexp	5-109
bak status	5-111
bak verifydb	5-114
bakserver	5-116
bos	5-118
bos addadmin	5-121
bos addkey	5-123
bos create	5-125
bos delete	5-128
bos exec	5-130
bos gckey	5-132
bos genkey	5-134
bos getdates	5-136
bos getlog	5-138
bos getrestart	5-140
bos install	5-142
bos lsadmin	5-144
bos lskeys	5-146
bos prune	5-149
bos restart	5-151
bos rmadmin	5-154
bos rmkey	5-156
bos setauth	5-158
bos setrestart	5-160
bos shutdown	5-163
bos start	5-165
bos startup	5-167
bos status	5-169
bos stop	5-172
bos uninstall	5-174
bossserver	5-176
butc	5-178
cm	5-181
cm checkfilesets	5-183
cm debug	5-184
cm flush	5-187
cm flushfileset	5-188

cm getcachesize	5-189
cm getdevok	5-191
cm getsetuid	5-193
cm lscellinfo	5-195
cm lsstores	5-196
cm resetstores	5-197
cm setcachesize	5-198
cm setdevok	5-200
cm setsetuid	5-202
cm sysname	5-204
dfsbind	5-206
dfsd	5-209
dfsexport	5-216
flserver	5-220
fms	5-222
fts	5-225
fts addsite	5-230
fts aggrinfo	5-233
fts clone	5-236
fts clonesys	5-238
fts create	5-241
fts crfldbentry	5-244
fts crmount	5-247
fts crserverentry	5-252
fts delete	5-255
fts delfldbentry	5-258
fts delmount	5-261
fts delserverentry	5-262
fts dump	5-264
fts edserverentry	5-267
fts lock	5-271
fts lsaggr	5-273
fts lsfdb	5-275
fts lsft	5-279
fts lsheader	5-283
fts lsmount	5-287
fts lsreplicas	5-289
fts lsserverentry	5-291
fts move	5-293
fts release	5-296
fts rename	5-298
fts restore	5-300
fts rmsite	5-305
fts setquota	5-307
fts setrepinfo	5-309
fts statftserver	5-316
fts statrepserver	5-318

fts syncfdb	5-320
fts syncserv	5-323
fts unlock	5-326
fts unlockfdb	5-328
fts update	5-330
fts zap	5-332
ftserver	5-334
fxd	5-336
growaggr	5-339
newaggr	5-342
repserver	5-347
salvage	5-349
scout	5-355
upclient	5-359
upserver	5-362
Chapter 6. DCE Commands	6-1
dce_config	6-2
Index	Index-1

List of Tables

Table 4-1. Locksmith Account Changes Made by the Security Server 4-8
Table 4-2. Registry Policy Changes Made by the Security Server 4-8

Preface

The *OSF DCE Administration Reference* provides complete and detailed reference information to help system and network administrators use the correct syntax for OSF™ Distributed Computing Environment (DCE) administration commands.

Audience

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

Applicability

This is Revision 1.0 of this document. It applies to the OSF™ DCE Version 1.0 offering and related updates. See your software license for details.

Purpose

The purpose of this reference is to assist system and network administrators with using the correct syntax for DCE administration commands.

Document Usage

This reference is organized into six chapters.

- For DCE Remote Procedure Call commands, see Chapter 1.
- For DCE Directory Service commands, see Chapter 2.
- For DCE Distributed Time Service commands, see Chapter 3.
- For DCE Security Service commands, see Chapter 4.
- For DCE Distributed File Service commands, see Chapter 5.
- For DCE commands, see Chapter 6.

Related Documents

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

- *Introduction to DCE*
- *DCE User's Guide and Reference*
- *DCE Application Development Guide*

- *DCE Application Development Reference*
- *DCE Administration Guide*
- *DCE Porting and Testing Guide*

Typographic and Keying Conventions

This document uses the following typographic conventions:

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

Italic *Italic* words or characters represent variable values that you must supply.

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.

<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.

Pathnames of Directories and Files in DCE Documentation

For a list of the pathnames for directories and files referred to in this document, see the *DCE Administration Guide* and the *OSF DCE Release Notes*.

Problem Reporting

If you have any problems with the software or documentation, please contact your software vendor's customer service department.

Permuted Index

Initializes the Basic OverSeer (BOS) Server process bossserver: bossserver(8dfs)
 generated by the Basic OverSeer (BOS) Server /Contains messages BosLog(4dfs)
 monitored by the Basic OverSeer (BOS) Server /processes to be BosConfig(4dfs)
 list for the Basic OverSeer (BOS) Server /the administrative admin.bos(4dfs)
 bak addftfamily: Creates a new (empty) fileset family in the/ bak addftfamily(8dfs)
 /tape size and end of file (EOF) mark size for a tape drive fms(8dfs)
 the Fileset Location Server (FL Server) /Initializes flserver(8dfs)
 for Global Directory Service (GDS) administration /to commands intro(8gds)
 modifiable, set-valued attribute (including application-defined/ /a add object(8cds)
 /or single-valued attribute (including application:defined/ remove directory(8cds)
 /or single-valued attribute (including application:defined/ remove object(8cds)
 modifiable, set-valued attribute (including application:defined/ /a add directory(8cds)
 /the specifies object entries (including clearinghouse object/ list object(8cds)
 machine keytab file /krb5/v5srvtab: The server and v5srvtab(5sec)
 /location mappings for filesets accessed by the Cache Manager FilesetItems(4dfs)
 fts statftserver: Reports on the activity of a Fileset Server fts statftserver(8dfs)
 modifiable, set-valued attribute/ add directory: Adds a value to a add directory(8cds)
 profile in a name service entry;/ add element: Adds an element to a add element(8rpc)
 entry to the name service/ add entry: Adds a name service add entry(8rpc)
 group in a name service entry;/ add member: Adds a member to a add member(8rpc)
 modifiable, set-valued attribute/ add object: Adds a value to a add object(8cds)
 server to an administrative/ bos addadmin: Adds a user, group, or bos addadmin(8dfs)
 the dump hierarchy bak adddump: Defines a dump level in bak adddump(8dfs)
 family entry in a fileset/ bak addftentry: Defines a fileset bak addftentry(8dfs)
 (empty) fileset family in/ bak addftfamily: Creates a new bak addftfamily(8dfs)
 entry to the Backup Database bak addhost: Adds a Tape Coordinator bak addhost(8dfs)
 /the system as a global server by adding the server's entry to the/ advertise(8dts)
 server encryption key and/ bos addkey: Converts a string into a bos addkey(8dfs)
 in/ show cached server: Displays address information of a server show cached server(8cds)

name service entry;/ add member:	Adds a member to a group in a	add member(8rpc)
name service database add entry:	Adds a name service entry to the	add entry(8rpc)
read/write DCE LFS/ fts addsite:	Adds a replication site for a	fts addsite(8dfs)
the Backup Database bak addhost:	Adds a Tape Coordinator entry to	bak addhost(8dfs)
an administrative/ bos addadmin:	Adds a user, group, or server to	bos addadmin(8dfs)
set-valued/ add directory:	Adds a value to a modifiable,	add directory(8cds)
set:valued attribute/ add object:	Adds a value to a modifiable,	add object(8cds)
name service entry;/ add element:	Adds an element to a profile in a	add element(8rpc)
into a server encryption key and	adds it to a key file /a string	bos addkey(8dfs)
Generates a random key and	adds it to a key file /genkey:	bos genkey(8dfs)
for a read/write DCE LFS/ fts	addsite: Adds a replication site	fts addsite(8dfs)
specified/ update:	Gradually adjusts the clock on the	update(8dts)
administrative list for the/	admin.bak: Contains the	admin.bak(4dfs)
administrative list for the/	admin.bos: Contains the	admin.bos(4dfs)
administrative list for the/	admin.fl: Contains the	admin.fl(4dfs)
administrative list for the/	admin.ft: Contains the	admin.ft(4dfs)
administrative list for the/	admin.up: Contains the	admin.up(4dfs)
gdscacheadm: Cache	administration program	gdscacheadm(8gds)
gdsditadm: Directory database	administration program	gdsditadm(8gds)
gdssysadm: Directory system	administration program.	gdssysadm(8gds)
Global Directory Service (GDS)	administration /to commands for	intro(8gds)
Introduction to the DCE Security	administrative commands intro:	intro(8sec)
Introduction to the DCE Security	administrative files intro:	intro(5sec)
OverSeer/ admin.bos: Contains the	administrative list for the Basic	admin.bos(4dfs)
Backup/ admin.bak: Contains the	administrative list for the	admin.bak(4dfs)
Fileset/ admin.fl: Contains the	administrative list for the	admin.fl(4dfs)
Fileset/ admin.ft: Contains the	administrative list for the	admin.ft(4dfs)
Update/ admin.up: Contains the	administrative list for the	admin.up(4dfs)
a user, group, or server to an	administrative list /Adds	bos addadmin(8dfs)
a user, group, or server from an	administrative list /Removes	bos rmdadmin(8dfs)
groups, and servers from an	administrative list /the users,	bos lsadmin(8dfs)
sec_admin: Security	administrative tool	sec_admin(8sec)
as a global server by adding the/	advertise: Configures the system	advertise(8dts)
daemon cdsadv: Starts the	advertisement and solicitation	cdsadv(8cds)
the entire contents of an	aggregate /restoredisk: Restores	bak restoredisk(8dfs)
or repair the structure of an	aggregate /to verify, recover,	salvage(8dfs)
newaggr: Initialize a DCE LFS	aggregate.	newaggr(8dfs)
Increases the size of a DCE LFS	aggregate. growaggr:	growaggr(8dfs)
that have/ dfsatab: Lists DCE LFS	aggregates and non-LFS partitions	dfsatab(4dfs)
that/ dfstab: Specifies DCE LFS	aggregates and non-LFS partitions	dfstab(4dfs)
fts lsaggr: Lists all exported	aggregates and partitions on a/	fts lsaggr(8dfs)
/disk space information about	aggregates and partitions/	fts aggrinfo(8dfs)
to/ dfsexport: Exports DCE LFS	aggregates or non-LFS partitions	dfsexport(8dfs)
information about aggregates/ fts	aggrinfo: Displays disk space	fts aggrinfo(8dfs)
/a directory's replica set,	allowing you to designate a new/	set directory to new epoch(8cds)
on the local node change:	Alters the epoch number and time	change(8dts)
/current size of the cache, the	amount of cache in use, and the/	cm getcachesize(8dfs)
/set:valued attribute (including	application:defined attributes)/	add object(8cds)
of a/ /attribute (including	application:defined attributes)	remove directory(8cds)
of an/ /attribute (including	application:defined attributes)	remove object(8cds)
/set-valued attribute (including	application:defined attributes)/	add directory(8cds)

containing a specified/ bak
node to the time specified by the
a read/write DCE LFS fileset and
/the File Exporter and start
fts lsmount: Lists the filesets
write/ cm resetstores: Cancels
value to a modifiable, set-valued
value to a modifiable, set-valued
a set-valued or single-valued
a set-valued or single-valued
show clearinghouse: Displays
specified/ show link: Displays
specified/ show child: Displays
CDS clerk/ show clerk: Displays
specified/ show object: Displays
specified/ show replica: Displays
server/ show server: Displays
show directory: Displays
of a modifiable, single-valued
of a modifiable, single-valued
/of a modifiable, single-valued
a soft link's timeout value
entry show entry: Shows the NSI
/(including application:defined
/(including application:defined
/(including application-defined
/(including application:defined
/and obtain and maintain user
/setauth: Enables or disables DFS
NoAuth: Indicates that DFS
server/ bos getrestart: Lists
makes an existing clearinghouse
Database bak savedb: Creates a
bak restoredb: Restores a
bak ftinfo: Queries the
a Tape Coordinator entry to the
new (empty) fileset family in the
record of a dump set from the
the dump hierarchy from the
fileset families defined in the
a Tape Coordinator entry from the
Deletes a dump level from the
Deletes a fileset family from the
Creates a backup copy of the
Checks the status of the
Tape Coordinator entries in the
Restores a backup copy of the
bakserver: Initializes the
the administrative list for the
messages generated by the
/Removes a specified read/write or
apropos: Shows each help entry bak apropos(8dfs)
argument /on the specified server update(8dts)
associated FLDB entry /Creates fts create(8dfs)
associated kernel daemons fxd(8dfs)
associated with mount points fts lsmount(8dfs)
attempts by the Cache Manager to cm resetstores(8dfs)
attribute (including/ /Adds a add directory(8cds)
attribute (including/ /Adds a add object(8cds)
attribute (including/ /value from remove directory(8cds)
attribute (including/ /value from remove object(8cds)
attribute information about the/ show clearinghouse(8cds)
attribute information about the show link(8cds)
attribute information about the show child(8cds)
attribute information about the show clerk(8cds)
attribute information about the show object(8cds)
attribute information about the show replica(8cds)
attribute information about the show server(8cds)
attribute information about the/ show directory(8cds)
attribute of a directory /value set directory(8cds)
attribute of a soft link /value set link(8cds)
attribute of an object entry set object(8cds)
attribute remove link: Removes remove link(8cds)
attributes of a name service show entry(8rpc)
attributes) of a directory add directory(8cds)
attributes) of a directory remove directory(8cds)
attributes) of an object entry add object(8cds)
attributes) of an object entry remove object(8cds)
authentication information dfsbind(8dfs)
authorization checking for all/ bos setauth(8dfs)
authorization checking is/ NoAuth(4dfs)
automatic restart times for bos getrestart(8dfs)
available /local server system or create clearinghouse(8cds)
backup copy of the Backup bak savedb(8dfs)
backup copy of the Backup/ bak restoredb(8dfs)
Backup Database about a fileset bak ftinfo(8dfs)
Backup Database /addhost: Adds bak addhost(8dfs)
Backup Database /Creates a bak addftfamily(8dfs)
Backup Database /Deletes the bak deletedump(8dfs)
Backup Database /Displays bak lsdumps(8dfs)
Backup Database /Lists bak lsftfamilies(8dfs)
Backup Database /rmhost: Removes bak rmhost(8dfs)
Backup Database bak rmdump: bak rmdump(8dfs)
Backup Database bak rmtfamily: bak rmtfamily(8dfs)
Backup Database bak savedb: bak savedb(8dfs)
Backup Database bak verifydb: bak verifydb(8dfs)
Backup Database. /lshosts: Lists bak lshosts(8dfs)
Backup Database. bak restoredb: bak restoredb(8dfs)
Backup Server bakserver(8dfs)
Backup Server /Contains admin.bak(4dfs)
Backup Server BakLog: Contains BakLog(4dfs)
backup version of a DCE LFS/ fts delete(8dfs)

DCE LFS/	fts clone: Creates backup version of a read/write	fts clone(8dfs)
read/write/	fts clonesys: Creates backup versions of all indicated	fts clonesys(8dfs)
Lists information about specified	backups bak dumpinfo: Lists information	bak dumpinfo(8dfs)
in the dump hierarchy	bak adddump: Defines a dump level	bak adddump(8dfs)
family entry in a fileset family	bak addftentry: Defines a fileset	bak addftentry(8dfs)
(empty) fileset family in the/	bak addftfamily: Creates a new	bak addftfamily(8dfs)
Coordinator entry to the Backup/	bak addhost: Adds a Tape	bak addhost(8dfs)
entry containing a specified/	bak apropos: Shows each help	bak apropos(8dfs)
bak: Introduction to the	bak command suite	bak(8dfs)
/help: Shows syntax of specified	bak commands or lists functional/	bak help(8dfs)
functional descriptions of all	bak commands /commands or lists	bak help(8dfs)
record of a dump set from the/	bak deletedump: Deletes the	bak deletedump(8dfs)
fileset family at a specific/	bak dump: Dumps a specific	bak dump(8dfs)
about specified backups	bak dumpinfo: Lists information	bak dumpinfo(8dfs)
Database about a fileset	bak ftinfo: Queries the Backup	bak ftinfo(8dfs)
specified bak commands or lists/	bak help: Shows syntax of	bak help(8dfs)
on a tape	bak labeltape: Creates the label	bak labeltape(8dfs)
hierarchy from the Backup/	bak lsdumps: Displays the dump	bak lsdumps(8dfs)
families defined in the Backup/	bak lsftfamilies: Lists fileset	bak lsftfamilies(8dfs)
Coordinator entries in the/	bak lshosts: Lists Tape	bak lshosts(8dfs)
and size from a tape's label	bak readlabel: Displays the name	bak readlabel(8dfs)
copy of the Backup Database.	bak restoredb: Restores a backup	bak restoredb(8dfs)
entire contents of an aggregate	bak restoredisk: Restores the	bak restoredisk(8dfs)
from tape	bak restoreft: Restores filesets	bak restoreft(8dfs)
from the Backup Database	bak rmdump: Deletes a dump level	bak rmdump(8dfs)
family entry from a fileset/	bak rmftentry: Deletes a fileset	bak rmftentry(8dfs)
family from the Backup Database	bak rmftfamily: Deletes a fileset	bak rmftfamily(8dfs)
Coordinator entry from the/	bak rmhost: Removes a Tape	bak rmhost(8dfs)
of the Backup Database	bak savedb: Creates a backup copy	bak savedb(8dfs)
information from a tape	bak scantape: Extracts dump set	bak scantape(8dfs)
date on an existing dump level	bak setexp: Sets the expiration	bak setexp(8dfs)
operation that a Tape/	bak status: Reports on the	bak status(8dfs)
of the Backup Database	bak verifydb: Checks the status	bak verifydb(8dfs)
command suite	bak: Introduction to the bak	bak(8dfs)
generated by the Backup Server	BakLog: Contains messages	BakLog(4dfs)
Server	bakserver: Initializes the Backup	bakserver(8dfs)
/Creates registry database entries	based on information in UNIX/	passwd_import(8sec)
/the administrative list for the	Basic OverSeer (BOS) Server	admin.bos(4dfs)
processes to be monitored by the	Basic OverSeer (BOS) Server	BosConfig(4dfs)
/messages generated by the	Basic OverSeer (BOS) Server	BosLog(4dfs)
bossserver: Initializes the	Basic OverSeer (BOS) Server/	bossserver(8dfs)
processes or only those with new	binaries /BOS Server restarts all	bos setrestart(8dfs)
bos prune: Removes old	binary and core files from/	bos prune(8dfs)
Installs a new version of a	binary file bos install:	bos install(8dfs)
Uses the former version of a	binary file bos uninstall:	bos uninstall(8dfs)
Lists timestamps on versions of	binary files bos getdates:	bos getdates(8dfs)
UUID from a/	import: Imports binding information and an object	import(8rpc)
interface/	export: Exports binding information for an	export(8rpc)
identifiers./	unexport: Removes binding information, interface	unexport(8rpc)
show server: Shows the	binding information, interface/	show server(8rpc)
or server to an administrative/	bos addadmin: Adds a user, group,	bos addadmin(8dfs)

into a server encryption key and/
 bos: Introduction to the
 in the BosConfig file and start/
 processes from the BosConfig/
 server machine
 server encryption keys from a/
 key and adds it to a key file
 versions of binary files
 for a server process
 restart times for server/
 version of a binary file
 groups, and servers from an/
 encryption key information from/
 core files from dcllocal/bin and/
 on a server machine
 group, or server from an/
 encryption keys from a key file
 /the date and time at which the
 DFS authorization checking for/
 time at which the BOS Server/
 without changing their status/
 setting their status flags to/
 changing their status flags to/
 of server processes on a server/
 of server processes on a server/
 changing their status flags in/
 version of a binary file
 command suite
 /their status flags to Run in the
 /Creates a new process in the
 /their status flags in the
 their status flags in the
 their status flags in the
 Deletes server processes from the
 processes to be monitored by the/
 generated by the Basic OverSeer/
 OverSeer (BOS) Server process
 cdsbrowser: Starts the CDS
 execution information from the
 Lists error messages from the
 Coordinator process
 fts dump: Converts a fileset to a
 restore: Converts a file from a
 gdsheadm:
 /size of the cache, the amount of
 daemons dfsd: Initializes the
 /filesets that contain data the
 /or reports the status of the
 cached from/ cm flush: Forces the
 cm flushfileset: Force the
 cm checkfilesets: Forces the
 bos addkey: Converts a string bos addkey(8dfs)
 bos command suite bos(8dfs)
 bos create: Creates a new process bos create(8dfs)
 bos delete: Deletes server bos delete(8dfs)
 bos exec: Executes a command on a bos exec(8dfs)
 bos gckeys: Removes obsolete bos gckeys(8dfs)
 bos genkey: Generates a random bos genkey(8dfs)
 bos getdates: Lists timestamps on bos getdates(8dfs)
 bos getlog: Examines the log file bos getlog(8dfs)
 bos getrestart: Lists automatic bos getrestart(8dfs)
 bos install: Installs a new bos install(8dfs)
 bos lsadmin: Lists the users, bos lsadmin(8dfs)
 bos lskeys: Displays server bos lskeys(8dfs)
 bos prune: Removes old binary and bos prune(8dfs)
 bos restart: Restarts processes bos restart(8dfs)
 bos rmdadmin: Removes a user, bos rmdadmin(8dfs)
 bos rmkey: Removes server bos rmkey(8dfs)
 BOS Server restarts all processes/ bos setrestart(8dfs)
 bos setauth: Enables or disables bos setauth(8dfs)
 bos setrestart: Sets the date and bos setrestart(8dfs)
 bos shutdown: Stops processes bos shutdown(8dfs)
 bos start: Starts processes after bos start(8dfs)
 bos startup: Starts processes by bos startup(8dfs)
 bos status: Displays the status bos status(8dfs)
 bos status: Displays the status bos status(8dfs)
 bos stop: Stops processes after bos stop(8dfs)
 bos uninstall: Uses the former bos uninstall(8dfs)
 bos: Introduction to the bos bos(8dfs)
 BosConfig file and in memory bos start(8dfs)
 BosConfig file and start it bos create(8dfs)
 BosConfig file to NotRun bos stop(8dfs)
 BosConfig file /without changing bos shutdown(8dfs)
 BosConfig file /without changing bos startup(8dfs)
 BosConfig file bos delete: bos delete(8dfs)
 BosConfig: Defines server BosConfig(4dfs)
 BosLog: Contains messages BosLog(4dfs)
 bossserver: Initializes the Basic bossserver(8dfs)
 Browser utility on the local/ cdsbrowser(8cdfs)
 butc process /Lists TL(4dfs)
 butc process TE_device_name: TE(4dfs)
 butc: Initializes a Tape butc(8dfs)
 bytestream format and place it in/ fts dump(8dfs)
 bytestream format to fileset/ fts fts restore(8dfs)
 Cache administration program gdsheadm(8gdfs)
 cache in use, and the type of/ cm getcachesize(8dfs)
 Cache Manager and start related dfsd(8dfs)
 Cache Manager cannot write back/ cm lsstores(8dfs)
 Cache Manager debugging trace cm debug(8dfs)
 Cache Manager to discard data cm flush(8dfs)
 Cache Manager to discard data/ cm flushfileset(8dfs)
 Cache Manager to update/ cm checkfilesets(8dfs)

to a/ /Cancels attempts by the machines in cells known to the filesets are honored by the for filesets accessed by the device files are honored by the the initial configuration of the of a server in the local clerk's of cache in use, and the type of about each V file in a disk of a server in the local clerk's defined from the local clerk's	Cache Manager to write data back	cm resetstores(8dfs)
Sets the size of a disk the contents of the clerk a chunk of data cached in a disk /Shows the current size of the the clerk cache dump clerk information about the specified current information about/ show /the Cache Manager to discard data /the Cache Manager to discard data	Cache Manager /database server	cm lscellinfo(8dfs)
Vn: Contains a chunk of data of a server in the local/ define information of a server in/ show of a server that you had/ clear configuration of the Cache/ about each V file in a disk/ rpcd: DCE Remote Procedure /the confidence level of clerk /current confidence level of clerk Manager to write/ cm resetstores: contain data the Cache Manager synchronize the/ synchronize: execution. exit: execution quit: system cdsbrowser: Starts the /attribute information about the intro: Introduction to the /read requests that result from /read requests that result from /calls issued as a result of /of clerk calls resulting from cdscp: Starts the maintain user/ dfsbind: Resolves cdsd: Restarts the interface between clients and the and solicitation daemon Browser utility on the local/ between clients and the CDS/ current confidence level of/ show confidence level of clerk/ set Specifies a preferred/ set	Cache Manager /files on specified	cm setdevok(8dfs)
	Cache Manager /location mappings	FilesetItem(4dfs)
	Cache Manager /Shows whether	cm getdevok(8dfs)
	Cache Manager CacheInfo: Defines	CacheInfo(4dfs)
	cache /address information	show cached server(8cds)
	cache /of the cache, the amount	cm getcachesize(8dfs)
	cache /Records information	CacheItems(4dfs)
	cache /server: Creates knowledge	define cached server(8cds)
	cache /that you had specifically	clear cached server(8cds)
	cache cm setcachesize:	cm setcachesize(8dfs)
	cache dump clerk cache: Displays	dump clerk cache(8cds)
	cache Vn: Contains	Vn(4dfs)
	cache, the amount of cache in/	cm getcachesize(8dfs)
	cache: Displays the contents of	dump clerk cache(8cds)
	cached clearinghouse /current	show cached clearinghouse(8cds)
	cached clearinghouse: Displays	show cached clearinghouse(8cds)
	cached from a file or directory	cm flush(8dfs)
	cached from filesets containing/	cm flushfileset(8dfs)
	cached in a disk cache	Vn(4dfs)
	cached server: Creates knowledge	define cached server(8cds)
	cached server: Displays address	show cached server(8cds)
	cached server: Removes knowledge	clear cached server(8cds)
	CacheInfo: Defines the initial	CacheInfo(4dfs)
	CacheItems: Records information	CacheItems(4dfs)
	Call daemon	rpcd(8rpc)
	calls issued as a result of CDS/	set cdscp confidence(8cds)
	calls resulting from CDS control/	show cdscp confidence(8cds)
	Cancels attempts by the Cache	cm resetstores(8dfs)
	cannot write back to a File/ /that	cm lstores(8dfs)
	Causes the DTS entity to	synchronize(8dts)
	Causes the DTSCP to complete	exit(8dts)
	Causes the DTSCP to complete	quit(8dts)
	CDS Browser utility on the local	cdsbrowser(8cds)
	CDS clerk on the local system	show clerk(8cds)
	CDS commands	intro(8cds)
	CDS control program commands	set cdscp preferred clearinghouse(8cds)
	CDS control program commands	show cdscp preferred clearinghouse(8cds)
	CDS control program commands	set cdscp confidence(8cds)
	CDS control program commands	show cdscp confidence(8cds)
	CDS control program	cdscp(8cds)
	CDS pathnames and obtain and	dfsbind(8dfs)
	CDS server	cdsd(8cds)
	CDS server cdsclerk: Manages the	cdsclerk(8cds)
	cdsadv: Starts the advertisement	cdsadv(8cds)
	cdsbrowser: Starts the CDS	cdsbrowser(8cds)
	cdsclerk: Manages the interface	cdsclerk(8cds)
	cdscp confidence: Displays the	show cdscp confidence(8cds)
	cdscp confidence: Sets the	set cdscp confidence(8cds)
	cdscp preferred clearinghouse:	set cdscp preferred clearinghouse(8cds)

Displays the preferred/ show cdscp preferred clearinghouse: show cdscp preferred clearinghouse(8cds)
program cdscp: Starts the CDS control cdscp(8cds)
cdsd: Restarts the CDS server cdsd(8cds)
/information you need to create a cell entry in either DNS or GDS show cell(8cds)
adding the server's entry to the cell profile /a global server by advertise(8dts)
the global server entry from the cell profile /Removes unadvertise(8dts)
you need to create a cell/ show cell: Displays the information show cell(8cds)
/Shows database server machines in cells known to the Cache Manager cm lscellinfo(8dfs)
and time on the local node change: Alters the epoch number change(8dts)
fts setrepinf: Sets or changes replication type and/ fts setrepinf(8dfs)
modifiable/ set directory: Changes the value of a set directory(8cds)
modifiable/ set link: Changes the value of a set link(8cds)
modifiable/ set object: Changes the value of a set object(8cds)
/shutdown: Stops processes without changing their status flags in/ bos shutdown(8dfs)
/flags to Run in memory without changing their status flags in/ bos startup(8dfs)
bos stop: Stops processes after changing their status flags in/ bos stop(8dfs)
bos startup: Starts processes by changing their status flags to/ bos startup(8dfs)
entity set: Modifies characteristics for the DTS set(8dts)
Manager to update/ cm checkfilesets: Forces the Cache cm checkfilesets(8dfs)
/or disables DFS authorization checking for all DFS server/ bos setauth(8dfs)
Indicates that DFS authorization checking is disabled NoAuth: NoAuth(4dfs)
Database bak verifydb: Checks the status of the Backup bak verifydb(8dfs)
whose names match the specified child name /the child pointers list child(8cds)
replica/ create child: Creates a child pointer at the master create child(8cds)
delete child: Deletes a child pointer from the namespace delete child(8cds)
information about the specified child pointer /Displays attribute show child(8cds)
/child: Displays a list of all the child pointers whose names match/ list child(8cds)
the master replica of the/ create child: Creates a child pointer at create child(8cds)
from the namespace delete child: Deletes a child pointer delete child(8cds)
child pointers whose names/ list child: Displays a list of all the list child(8cds)
information about the/ show child: Displays attribute show child(8cds)
cache Vn: Contains a chunk of data cached in a disk Vn(4dfs)
knowledge of a server that you/ clear cached server: Removes clear cached server(8cds)
knowledge of the specified/ clear clearinghouse: Removes clear clearinghouse(8cds)
system or makes an existing clearinghouse available /server create clearinghouse(8cds)
requests/ /Displays the preferred clearinghouse for satisfying read show cdscp preferred clearinghouse(8cds)
server/ /Deletes the specified clearinghouse from the local delete clearinghouse(8cds)
/knowledge of the specified clearinghouse from the server's/ clear clearinghouse(8cds)
/whose names match the specified clearinghouse name list clearinghouse(8cds)
/object entries (including clearinghouse object entries) list object(8cds)
create clearinghouse: Creates a clearinghouse on the local/ create clearinghouse(8cds)
satisfying/ /Specifies a preferred clearinghouse to use for set cdscp preferred clearinghouse(8cds)
replica of a directory from a clearinghouse /a read-only delete replica(8cds)
information about the specified clearinghouse /Displays attribute show clearinghouse(8cds)
about the specified cached clearinghouse /information show cached clearinghouse(8cds)
directory in the specified clearinghouse /of an existing create replica(8cds)
clearinghouse on the/ create clearinghouse: Creates a create clearinghouse(8cds)
specified clearinghouse/ delete clearinghouse: Deletes the delete clearinghouse(8cds)
all the clearinghouses/ list clearinghouse: Displays a list of list clearinghouse(8cds)
information about the/ show clearinghouse: Displays attribute show clearinghouse(8cds)
information about/ show cached clearinghouse: Displays current show cached clearinghouse(8cds)

preferred/ show cdscp preferred	clearinghouse: Displays the	show cdscp preferred clearinghouse(8cds)
of the specified/ clear	clearinghouse: Removes knowledge	clear clearinghouse(8cds)
preferred/ set cdscp preferred	clearinghouse: Specifies a	set cdscp preferred clearinghouse(8cds)
the/ /Displays a list of all the	clearinghouses whose names match	list clearinghouse(8cds)
/the current confidence level of	clerk calls resulting from CDS/	show cdscp confidence(8cds)
Displays the contents of the	clerk cache dump clerk cache:	dump clerk cache(8cds)
contents of the clerk cache dump	clerk cache: Displays the	dump clerk cache(8cds)
of/ /Sets the confidence level of	clerk calls issued as a result	set cdscp confidence(8cds)
/information about the CDS	clerk on the local system	show clerk(8cds)
disable clerk: Stops the	clerk on the local system	disable clerk(8cds)
of a server in the local	clerk's cache /Creates knowledge	define cached server(8cds)
defined from the local	clerk's cache /had specifically	clear cached server(8cds)
of a server in the local	clerk's cache /information	show cached server(8cds)
information about the CDS/ show	clerk: Displays attribute	show clerk(8cds)
local system disable	clerk: Stops the clerk on the	disable clerk(8cds)
sec_clientd: Security	client daemon	sec_clientd(8sec)
Server upclient: Initializes the	client portion of the Update	upclient(8dfs)
/Manages the interface between	clients and the CDS server	cdsclerk(8cds)
/the DTS entity to synchronize the	clock in the system where the/	synchronize(8dts)
update: Gradually adjusts the	clock on the specified server/	update(8dts)
a read/write DCE LFS fileset fits	clone: Creates backup version of	fts clone(8dfs)
of all indicated read/write/ fts	clonesys: Creates backup versions	fts clonesys(8dfs)
Cache Manager to update/	cm checkfilesets: Forces the	cm checkfilesets(8dfs)
cm: Introduction to the	cm command suite	cm(8dfs)
debugging output from the	cm debug command /Contains	DFSLog(4dfs)
reports the status of the Cache/	cm debug: Enables, disables, or	cm debug(8dfs)
Manager to discard data cached/	cm flush: Forces the Cache	cm flush(8dfs)
Manager to discard data cached/	cm flushfileset: Force the Cache	cm flushfileset(8dfs)
current size of the cache, the/	cm getcachesize: Shows the	cm getcachesize(8dfs)
files are honored by the Cache/	cm getdevok: Shows whether device	cm getdevok(8dfs)
setuid programs on specified/	cm getsetuid: Shows the status of	cm getsetuid(8dfs)
server machines in cells known/	cm lscellinfo: Shows database	cm lscellinfo(8dfs)
contain data the Cache Manager/	cm lsstores: Lists filesets that	cm lsstores(8dfs)
by the Cache Manager to write/	cm resetstores: Cancels attempts	cm resetstores(8dfs)
a disk cache	cm setcachesize: Sets the size of	cm setcachesize(8dfs)
device files on specified/	cm setdevok: Specifies whether	cm setdevok(8dfs)
setuid programs on specified/	cm setsetuid: Enables or disables	cm setsetuid(8dfs)
CPU/OS type	cm sysname: Reports or set the	cm sysname(8dfs)
command suite	cm: Introduction to the cm	cm(8dfs)
the clock in the system where the	command is entered /synchronize	synchronize(8dts)
bos exec: Executes a	command on a server machine	bos exec(8dfs)
bak: Introduction to the bak	command suite	bak(8dfs)
bos: Introduction to the bos	command suite	bos(8dfs)
cm: Introduction to the cm	command suite	cm(8dfs)
fts: Introduction to the fts	command suite	fts(8dfs)
or the options of a specified	command /a list of commands	help(8rpc)
output from the cm debug	command /Contains debugging	DFSLog(4dfs)
Lists the output of the fms	command FMSLog:	FMSLog(4dfs)
Service/ intro: Introduction to	commands for Global Directory	intro(8gds)
/Shows syntax of specified bak	commands or lists functional/	bak help(8dfs)
help: Displays a list of	commands or the options of a/	help(8rpc)

intro: Introduction to the CDS commands intro(8cds)
 intro: Introduction to the DFS commands intro(8dfs)
 from CDS control program commands /clerk calls resulting show cdscp confidence(8cds)
 a result of CDS control program commands /clerk calls issued as set cdscp confidence(8cds)
 daemon and RPC control program commands /Introduction to DCE RPC intro(8rpc)
 descriptions of all bak commands /or lists functional bak help(8dfs)
 result from CDS control program commands /read requests that set cdscp preferred clearinghouse(8cds)
 result from CDS control program commands /read requests that show cdscp preferred clearinghouse(8cds)
 the DCE Security administrative commands intro: Introduction to intro(8sec)
 to the DCE DTS control program commands Intro: Introduction intro(8dts)
 help information about DTSCP commands. help: Displays help(8dts)
 quit: Causes the DTSCP to complete execution quit(8dts)
 exit: Causes the DTSCP to complete execution. exit(8dts)
 /confidence: Displays the current confidence level of clerk calls/ show cdscp confidence(8cds)
 set cdscp confidence: Sets the confidence level of/ show cdscp confidence level of clerk calls/ set cdscp confidence: Sets the confidence set cdscp confidence(8cds)
 CacheInfo: Defines the initial configuration of the Cache/ CacheInfo(4dfs)
 drives on a/ TapeConfig: Defines configuration parameters for tape TapeConfig(4dfs)
 server by adding the/ advertise: Configures the system as a global advertise(8dts)
 dce_config: Installs, configures, and starts up DCE dce_config(8dce)
 cm lsstores: Lists filesets that contain data the Cache Manager/ cm lsstores(8dfs)
 /apropos: Shows each help entry containing a specified string bak apropos(8dfs)
 discard data cached from filesets containing specified files or/ /to cm flushfileset(8dfs)
 in a disk cache Vn: Contains a chunk of data cached Vn(4dfs)
 the cm debug command DFSLog: Contains debugging output from DFSLog(4dfs)
 the Backup Server BakLog: Contains messages generated by BakLog(4dfs)
 the Basic OverSeer (BOS)/ BosLog: Contains messages generated by BosLog(4dfs)
 the File Exporter FileLog: Contains messages generated by FileLog(4dfs)
 the Fileset Location/ FILog: Contains messages generated by FILog(4dfs)
 the Fileset Server FtLog: Contains messages generated by FtLog(4dfs)
 the Replication Server RepLog: Contains messages generated by RepLog(4dfs)
 the server portion of the/ UpLog: Contains messages generated by UpLog(4dfs)
 DFS Salvager SalvageLog: Contains output generated by the SalvageLog(4dfs)
 for the Backup Server admin.bak: Contains the administrative list admin.bak(4dfs)
 for the Basic/ admin.bos: Contains the administrative list admin.bos(4dfs)
 for the Fileset/ admin.fl: Contains the administrative list admin.fl(4dfs)
 for the Fileset Server admin.ft: Contains the administrative list admin.ft(4dfs)
 for the Update Server admin.up: Contains the administrative list admin.up(4dfs)
 restoredisk: Restores the entire contents of an aggregate bak bak restoredisk(8dfs)
 dump clerk cache: Displays the contents of the clerk cache dump clerk cache(8cds)
 requests that result from CDS control program commands /read set cdscp preferred clearinghouse(8cds)
 requests that result from CDS control program commands /read show cdscp preferred clearinghouse(8cds)
 /to DCE RPC daemon and RPC control program commands intro(8rpc)
 control program commands /clerk set cdscp confidence(8cds)
 control program commands /of show cdscp confidence(8cds)
 control program commands Intro: intro(8dts)
 control program cdscp(8cds)
 control program dtscp(8dts)
 control program rpccp(8rpc)
 system interface/ gdsctl: Controls and edits transport gdsctl(8gds)

format to fileset/ fts restore:	Converts a file from a bytestream	fts restore(8dfs)
bytestream format and/ fts dump:	Converts a fileset to a	fts dump(8dfs)
encryption key and/ bos addkey:	Converts a string into a server	bos addkey(8dfs)
bak lshosts:	Lists Tape Coordinator entries in the Backup/	bak lshosts(8dfs)
bak rmhost:	Removes a Tape Coordinator entry from the Backup/	bak rmhost(8dfs)
bak addhost:	Adds a Tape Coordinator entry to the Backup/	bak addhost(8dfs)
on the operation that a Tape	Coordinator is executing /Reports	bak status(8dfs)
for tape drives on a Tape	Coordinator machine /parameters	TapeConfig(4dfs)
butc:	Initializes a Tape Coordinator process	butc(8dfs)
bak savedb:	Creates a backup copy of the Backup Database	bak savedb(8dfs)
bak restoredb:	Restores a backup copy of the Backup Database.	bak restoredb(8dfs)
bos prune:	Removes old binary and/	bos prune(8dfs)
cm sysname:	Reports or set the CPU/OS type	cm sysname(8dfs)
DNS/ /the information you need to	create a cell entry in either	show cell(8cds)
pointer at the master replica of/	create child: Creates a child	create child(8cds)
clearinghouse on the local/	create clearinghouse: Creates a	create clearinghouse(8cds)
directory	create directory: Creates a	create directory(8cds)
and optionally specifies an/	create link: Creates a soft link	create link(8cds)
entry	create object: Creates an object	create object(8cds)
of an existing directory in the/	create replica: Creates a replica	create replica(8cds)
the BosConfig file and start/ bos	create: Creates a new process in	bos create(8dfs)
LFS fileset and associated/ fts	create: Creates a read/write DCE	fts create(8dfs)
entity on the specified node	create: Creates the DCE DTS	create(8dts)
Backup Database bak savedb:	Creates a backup copy of the	bak savedb(8dfs)
master replica of/ create child:	Creates a child pointer at the	create child(8cds)
local/ create clearinghouse:	Creates a clearinghouse on the	create clearinghouse(8cds)
create directory:	Creates a directory	create directory(8cds)
FLDB fts crfldbentry:	Creates a fileset entry in the	fts crfldbentry(8dfs)
fileset fts crmount:	Creates a mount point for a	fts crmount(8dfs)
family in the/ bak addftfamily:	Creates a new (empty) fileset	bak addftfamily(8dfs)
BosConfig file and/ bos create:	Creates a new process in the	bos create(8dfs)
fileset and/ fts create:	Creates a read/write DCE LFS	fts create(8dfs)
directory in/ create replica:	Creates a replica of an existing	create replica(8cds)
FLDB. fts crserverentry:	Creates a server entry in the	fts crserverentry(8dfs)
optionally/ create link:	Creates a soft link and	create link(8cds)
create object:	Creates an object entry	create object(8cds)
read/write DCE LFS/ fts clone:	Creates backup version of a	fts clone(8dfs)
indicated/ fts clonesys:	Creates backup versions of all	fts clonesys(8dfs)
the local/ define cached server:	Creates knowledge of a server in	define cached server(8cds)
files passwd_export:	Creates local password and group	passwd_export(8sec)
based on/ passwd_import:	Creates registry database entries	passwd_import(8sec)
specified node create:	Creates the DCE DTS entity on the	create(8dts)
specified entry does not exist,	creates the entry /entry; if the	add member(8rpc)
specified entry does not exist,	creates the entry /entry; if the	export(8rpc)
specified entry does not exist,	creates the entry. /entry; if the	add element(8rpc)
bak labeltape:	Creates the label on a tape	bak labeltape(8dfs)
sec_create_db:	Registry database creation utility	sec_create_db(8sec)
entry in the FLDB fts	crfldbentry: Creates a fileset	fts crfldbentry(8dfs)
for a fileset fts	crmount: Creates a mount point	fts crmount(8dfs)
entry in the FLDB. fts	crserverentry: Creates a server	fts crserverentry(8dfs)
/cdscp confidence: Displays the	current confidence level of clerk/	show cdscp confidence(8cds)

entity show: Displays current information about the DTS show(8dts)
 /cached clearinghouse: Displays current information about the/ show cached clearinghouse(8cds)
 cm getcachesize: Shows the current size of the cache, the/ cm getcachesize(8dfs)
 intro: Introduction to DCE RPC daemon and RPC control program/ intro(8rpc)
 /Displays information on GDS daemon processes and processes/ gdsdirinfo(8gds)
 dtsd: Restarts the DTS daemon dtsd(8dts)
 gdad: Starts the GDA daemon gdad(8cds)
 rpcd: DCE Remote Procedure Call daemon rpcd(8rpc)
 sec_clientd: Security client daemon sec_clientd(8sec)
 advertisement and solicitation daemon cdsadv: Starts the cdsadv(8cds)
 and start associated kernel daemons /the File Exporter fxd(8dfs)
 Cache Manager and start related daemons dfds: Initializes the dfds(8dfs)
 /by the Cache Manager to write data back to a File Server and/ cm resetstores(8dfs)
 /the Cache Manager to discard data cached from a file or/ cm flush(8dfs)
 the Cache Manager to discard data cached from filesets/ /Force cm flushfileset(8dfs)
 Vn: Contains a chunk of data cached in a disk cache Vn(4dfs)
 /Lists filesets that contain data the Cache Manager cannot/ cm lsstores(8dfs)
 Server and discard the unstored data /write data back to a File cm resetstores(8dfs)
 bak ftinfo: Queries the Backup Database about a fileset bak ftinfo(8dfs)
 gdsditadm: Directory database administration program gdsditadm(8gds)
 sec_create_db: Registry database creation utility sec_create_db(8sec)
 passwd_import: Creates registry database entries based on/ passwd_import(8sec)
 passwd_override: The registry database override file passwd_override(5sec)
 known to/ cm lscellinfo: Shows database server machines in cells cm lscellinfo(8dfs)
 rgy_edit: Edits the registry database rgy_edit(8sec)
 Coordinator entry to the Backup Database /addhost: Adds a Tape bak addhost(8dfs)
 fileset family in the Backup Database /Creates a new (empty) bak addfamily(8dfs)
 of a dump set from the Backup Database /Deletes the record bak deletedump(8dfs)
 families defined in the Backup Database /Lists fileset bak lsftfamilies(8dfs)
 dump hierarchy from the Backup Database /lsdumps: Displays the bak lsdumps(8dfs)
 entry from the name service database /Removes a name service remove entry(8rpc)
 Coordinator entry from the Backup Database /rmhost: Removes a Tape bak rmhost(8dfs)
 service entry to the name service database add entry: Adds a name add entry(8rpc)
 a dump level from the Backup Database bak rmdump: Deletes bak rmdump(8dfs)
 a fileset family from the Backup Database bak rmfamily: Deletes bak rmfamily(8dfs)
 a backup copy of the Backup Database bak savedb: Creates bak savedb(8dfs)
 Checks the status of the Backup Database bak verifydb: bak verifydb(8dfs)
 Coordinator entries in the Backup Database. /lshosts: Lists Tape bak lshosts(8dfs)
 a backup copy of the Backup Database. /restoredb: Restores bak restoredb(8dfs)
 Server/ bos setrestart: Sets the date and time at which the BOS bos setrestart(8dfs)
 bak setexp: Sets the expiration date on an existing dump level bak setexp(8dfs)
 Intro: Introduction to the DCE DTS control program commands intro(8dts)
 node create: Creates the DCE DTS entity on the specified create(8dts)
 node disable: Stops the DCE DTS entity on the specified disable(8dts)
 delete: Deletes the DCE DTS entity delete(8dts)
 growaggr: Increases the size of a DCE LFS aggregate. growaggr(8dfs)
 newaggr: Initialize a DCE LFS aggregate. newaggr(8dfs)
 partitions that/ dfsatab: Lists DCE LFS aggregates and non-LFS dfsatab(4dfs)
 partitions/ dfstab: Specifies DCE LFS aggregates and non-LFS dfstab(4dfs)
 partitions to/ dfsexport: Exports DCE LFS aggregates or non-LFS dfsexport(8dfs)
 fts create: Creates a read/write DCE LFS fileset and associated/ fts create(8dfs)

/read-only version of a read/write	DCE LFS fileset at the local site	fts release(8dfs)
without/ fts zap: Removes a	DCE LFS fileset from its site	fts zap(8dfs)
/Displays the status of	DCE LFS fileset replicas	fts lsreplicas(8dfs)
fts move: Moves a read/write	DCE LFS fileset to another site	fts move(8dfs)
read/write or backup version of a	DCE LFS fileset /a specified	fts delete(8dfs)
replication site for a read/write	DCE LFS fileset /addsite: Adds a	fts addsite(8dfs)
backup version of a read/write	DCE LFS fileset /clone: Creates	fts clone(8dfs)
of replicas of a read/write	DCE LFS fileset /immediate update	fts update(8dfs)
and parameters for a read/write	DCE LFS fileset /replication type	fts setrepinfo(8dfs)
a replication site and read-only	DCE LFS fileset /rmsite: Removes	fts rmsite(8dfs)
maximum quota for a read/write	DCE LFS fileset /Sets the	fts setquota(8dfs)
of all indicated read/write	DCE LFS filesets. /versions	fts clonesys(8dfs)
or non-LFS partitions to the	DCE namespace /DCE LFS aggregates	dfsexport(8dfs)
rpcd:	DCE Remote Procedure Call daemon	rpcd(8rpc)
program/ intro: Introduction to	DCE RPC daemon and RPC control	intro(8rpc)
intro: Introduction to the	DCE Security administrative files	intro(5sec)
intro: Introduction to the	DCE Security administrative/	intro(8sec)
secd: The	DCE Security Server	secd(8sec)
configures, and starts up	DCE dce_config: Installs,	dce_config(8dce)
old binary and core files from	dcelocal/bin and/ /prune: Removes	bos prune(8dfs)
core files from dcelocal/bin and	dcelocal/var/dfs/adm /binary and	bos prune(8dfs)
and starts up DCE	dce_config: Installs, configures,	dce_config(8dce)
debugging output from the cm	debug command DFSLog: Contains	DFSLog(4dfs)
reports the status of the/ cm	debug: Enables, disables, or	cm debug(8dfs)
debug command DFSLog: Contains	debugging output from the cm	DFSLog(4dfs)
the status of the Cache Manager	debugging trace /or reports	cm debug(8dfs)
knowledge of a server in the/	define cached server: Creates	define cached server(8cds)
server that you had specifically	defined from the local clerk's/ /a	clear cached server(8cds)
/Lists fileset families	defined in the Backup Database	bak lsftfamilies(8dfs)
hierarchy bak adddump:	Defines a dump level in the dump	bak adddump(8dfs)
a fileset family bak addtentry:	Defines a fileset family entry in	bak addtentry(8dfs)
for tape drives on a/ TapeConfig:	Defines configuration parameters	TapeConfig(4dfs)
monitored by the/ BosConfig:	Defines server processes to be	BosConfig(4dfs)
of the Cache Manager CacheInfo:	Defines the initial configuration	CacheInfo(4dfs)
pointer from the namespace	delete child: Deletes a child	delete child(8cds)
specified clearinghouse from the/	delete clearinghouse: Deletes the	delete clearinghouse(8cds)
directory	delete directory: Deletes a	delete directory(8cds)
entry	delete link: Deletes a soft link	delete link(8cds)
read-only replica of a directory/	delete object: Deletes an object	delete object(8cds)
from the BosConfig file bos	delete replica: Deletes a	delete replica(8cds)
entity	delete: Deletes server processes	bos delete(8dfs)
read/write or backup version/ fts	delete: Deletes the DCE DTS	delete(8dts)
a dump set from the Backup/ bak	delete: Removes a specified	fts delete(8dfs)
namespace delete child:	deletedump: Deletes the record of	bak deletedump(8dfs)
delete directory:	Deletes a child pointer from the	delete child(8cds)
Backup Database bak rmdump:	Deletes a directory	delete directory(8cds)
from a fileset/ bak rmtentry:	Deletes a dump level from the	bak rmdump(8dfs)
Backup Database bak rmtfamily:	Deletes a fileset family entry	bak rmtentry(8dfs)
directory from a/ delete replica:	Deletes a fileset family from the	bak rmtfamily(8dfs)
FLDB fts delserverentry:	Deletes a read-only replica of a	delete replica(8cds)
	Deletes a server entry from the	fts delserverentry(8dfs)

delete link: Deletes a soft link delete link(8cds)
 delete object: Deletes an object entry delete object(8cds)
 BosConfig file bos delete: Deletes server processes from the bos delete(8dfs)
 delete: Deletes the DCE DTS entity delete(8dts)
 from the Backup/ bak deletedump: Deletes the record of a dump set bak deletedump(8dfs)
 delete clearinghouse: Deletes the specified/ delete clearinghouse(8cds)
 entry from the FLDB fts delflbentry: Removes a specified fts delflbentry(8dfs)
 fts delmount: Removes a mount point fts delmount(8dfs)
 entry from the FLDB fts delserverentry: Deletes a server fts delserverentry(8dfs)
 /bak commands or lists functional descriptions of all bak commands bak help(8dfs)
 to/ /replica set, allowing you to designate a new master replica or set directory to new epoch(8cds)
 file (EOF) mark size for a/ fms: Determines tape size and end of fms(8dfs)
 Cache/ cm getdevok: Shows whether device files are honored by the cm getdevok(8dfs)
 cm setdevok: Specifies whether device files on specified/ cm setdevok(8dfs)
 bos setauth: Enables or disables DFS authorization checking for/ bos setauth(8dfs)
 disabled NoAuth: Indicates that DFS authorization checking is NoAuth(4dfs)
 intro: Introduction to the DFS commands intro(8dfs)
 intro: Introduction to DFS files intro(4dfs)
 or repair the/ salvage: Uses the DFS Salvager to verify, recover, salvage(8dfs)
 Contains output generated by the DFS Salvager SalvageLog: SalvageLog(4dfs)
 /authorization checking for all DFS server processes on a machine bos setauth(8dfs)
 and non-LFS partitions that have/ dfsatab: Lists DCE LFS aggregates dfsatab(4dfs)
 and obtain and maintain user/ dfsbind: Resolves CDS pathnames dfsbind(8dfs)
 Manager and start related/ dfsd: Initializes the Cache dfsd(8dfs)
 aggregates or non-LFS partitions/ dfsexport: Exports DCE LFS dfsexport(8dfs)
 from the cm debug command DFSLog: Contains debugging output DFSLog(4dfs)
 aggregates and non-LFS/ dfstab: Specifies DCE LFS dfstab(4dfs)
 /Displays a list of all the directories whose names match the/ list directory(8cds)
 containing specified files or directories. /from filesets cm flushfileset(8dfs)
 program gdsditadm: Directory database administration gdsditadm(8gds)
 /Deletes a read-only replica of a directory from a clearinghouse delete replica(8cds)
 to skulk: Starts the skulk of a directory immediately /directory set directory to skulk(8cds)
 /Creates a replica of an existing directory in the specified/ create replica(8cds)
 whose names match the specified directory name /the directories list directory(8cds)
 /to commands for Global Directory Service (GDS)/ intro(8gds)
 program. gdssysadm: Directory system administration gdssysadm(8gds)
 Reconstructs a directory's/ set directory to new epoch: set directory to new epoch(8cds)
 skulk of a directory/ set directory to skulk: Starts the set directory to skulk(8cds)
 create directory: Creates a directory create directory(8cds)
 delete directory: Deletes a directory delete directory(8cds)
 the master replica of the parent directory /a child pointer at create child(8cds)
 attributes) of a directory /application:defined add directory(8cds)
 information about the specified directory /application:defined remove directory(8cds)
 data cached from a file or directory /Displays attribute show directory(8cds)
 single-valued attribute of a directory /Manager to discard cm flush(8dfs)
 you/ /to new epoch: Reconstructs a directory /value of a modifiable, set directory(8cds)
 modifiable, set-valued/ add directory's replica set, allowing set directory to new epoch(8cds)
 modifiable, single-valued/ set directory: Adds a value to a add directory(8cds)
 create directory: Changes the value of a set directory(8cds)
 delete directory: Creates a directory create directory(8cds)
 delete directory: Deletes a directory delete directory(8cds)

the directories whose names/ list directory: Displays a list of all list directory(8cds)
information about the/ show directory: Displays attribute show directory(8cds)
set-valued or/ remove directory: Removes a value from a remove directory(8cds)
the local system disable clerk: Stops the clerk on disable clerk(8cds)
on the local system disable server: Stops the server disable server(8cds)
on the specified node disable: Stops the DCE DTS entity disable(8dts)
DFS authorization checking is disabled NoAuth: Indicates that NoAuth(4dfs)
checking/ bos setauth: Enables or disables DFS authorization bos setauth(8dfs)
cm setsetuid: Enables or disables setuid programs on/ cm setsetuid(8dfs)
of the Cache/ cm debug: Enables, disables, or reports the status cm debug(8dfs)
or/ /Forces the Cache Manager to discard data cached from a file cm flush(8dfs)
/Force the Cache Manager to discard data cached from filesets/ cm flushfileset(8dfs)
data back to a File Server and discard the unstored data /write cm resetstores(8dfs)
about each V file in a disk cache /Records information CacheItems(4dfs)
setcachesize: Sets the size of a disk cache cm cm setcachesize(8dfs)
a chunk of data cached in a disk cache Vn: Contains Vn(4dfs)
fts agrinfo: Displays disk space information about/ fts agrinfo(8dfs)
pointers whose names/ list child: Displays a list of all the child list child(8cds)
links whose names/ list link: Displays a list of all the soft list link(8cds)
directories/ list directory: Displays a list of all the list directory(8cds)
list clearinghouse: Displays a list of all the/ list clearinghouse(8cds)
the options of a specified/ help: Displays a list of commands or help(8rpc)
server in/ show cached server: Displays address information of a show cached server(8cds)
about the specified/ show child: Displays attribute information show child(8cds)
about the/ show clearinghouse: Displays attribute information show clearinghouse(8cds)
about the CDS clerk/ show clerk: Displays attribute information show clerk(8cds)
about the/ show directory: Displays attribute information show directory(8cds)
about the specified/ show link: Displays attribute information show link(8cds)
about the specified/ show object: Displays attribute information show object(8cds)
about the/ show replica: Displays attribute information show replica(8cds)
about the server/ show server: Displays attribute information show server(8cds)
about/ show cached clearinghouse: Displays current information show cached clearinghouse(8cds)
about the DTS entity show: Displays current information show(8dts)
about aggregates/ fts agrinfo: Displays disk space information fts agrinfo(8dfs)
DTSCP commands. help: Displays help information about help(8dts)
daemon processes and/ gdsdirinfo: Displays information on GDS gdsdirinfo(8gds)
information from a/ bos lskeys: Displays server encryption key bos lskeys(8dfs)
clerk cache dump clerk cache: Displays the contents of the dump clerk cache(8cds)
level of/ show cdscp confidence: Displays the current confidence show cdscp confidence(8cds)
the Backup Database bak ls.dumps: Displays the dump hierarchy from bak ls.dumps(8dfs)
to create a cell/ show cell: Displays the information you need show cell(8cds)
tape's label bak readlabel: Displays the name and size from a bak readlabel(8dfs)
cdscp preferred clearinghouse: Displays the preferred/ show show cdscp preferred clearinghouse(8cds)
Replication/ fts statrepserver: Displays the status of a fts statrepserver(8dfs)
fileset replicas fts lsreplicas: Displays the status of DCE LFS fts lsreplicas(8dfs)
processes on a/ bos status: Displays the status of server bos status(8dfs)
processes on a/ bos status: Displays the status of server bos status(8dfs)
to create a cell entry in either DNS or GDS /information you need show cell(8cds)
file (EOF) mark size for a tape drive /tape size and end of fms(8dfs)
/configuration parameters for tape drives on a Tape Coordinator/ TapeConfig(4dfs)
Intro: Introduction to the DCE DTS control program commands intro(8dts)

dtscp: DTS control program dtscp(8dts)
 dtsd: Restarts the DTS daemon dtsd(8dts)
 create: Creates the DCE DTS entity on the specified node create(8dts)
 disable: Stops the DCE DTS entity on the specified node disable(8dts)
 enable: Starts the DTS entity on the specified node. enable(8dts)
 clock in/ synchronize: Causes the DTS entity to synchronize the synchronize(8dts)
 delete: Deletes the DCE DTS entity delete(8dts)
 Modifies characteristics for the DTS entity set: set(8dts)
 current information about the DTS entity show: Displays show(8dts)
 Displays help information about DTSCP commands. help: help(8dts)
 quit: Causes the DTSCP to complete execution quit(8dts)
 exit: Causes the DTSCP to complete execution. exit(8dts)
 dtscp: DTS control program dtscp(8dts)
 dtsd: Restarts the DTS daemon dtsd(8dts)
 contents of the clerk cache dump clerk cache: Displays the dump clerk cache(8cds)
 bak lsdumps: Displays the dump hierarchy from the Backup/ bak lsdumps(8dfs)
 Defines a dump level in the dump hierarchy bak adddump: bak adddump(8dfs)
 Database bak rmdump: Deletes a dump level from the Backup bak rmdump(8dfs)
 bak adddump: Defines a dump level in the dump hierarchy bak adddump(8dfs)
 fileset family at a specific dump level /Dumps a specific bak dump(8dfs)
 expiration date on an existing dump level bak setexp: Sets the bak setexp(8dfs)
 Database /Deletes the record of a dump set from the Backup bak deletedump(8dfs)
 bak scantape: Extracts dump set information from a tape bak scantape(8dfs)
 bytestream format and place/ fts dump: Converts a fileset to a fts dump(8dfs)
 family at a specific dump/ bak dump: Dumps a specific fileset bak dump(8dfs)
 specified backups bak dumpinfo: Lists information about bak dumpinfo(8dfs)
 at a specific dump/ bak dump: Dumps a specific fileset family bak dump(8dfs)
 fts edserventry: Edits a server entry in the FLDB fts edserventry(8dfs)
 rgy_edit: Edits the registry database rgy_edit(8sec)
 library/ gdscomxl: Controls and edits transport system interface gdscomxl(8gds)
 entry in the FLDB fts edserventry: Edits a server fts edserventry(8dfs)
 need to create a cell entry in either DNS or GDS /you show cell(8cds)
 /Removes specified elements from either the local or a remote/ remove mapping(8rpc)
 /Shows the elements of the either the local or a remote/ show mapping(8rpc)
 service/ add element: Adds an element to a profile in a name add element(8rpc)
 profile in a name service/ add element: Adds an element to a add element(8rpc)
 elements from a profile remove element: Removes selected remove element(8rpc)
 /profile: Removes all profile elements and the profile from the/ remove profile(8rpc)
 remove element: Removes selected elements from a profile remove element(8rpc)
 remove mapping: Removes specified elements from either the local or/ remove mapping(8rpc)
 show profile: Shows the elements of a profile show profile(8rpc)
 or a/ show mapping: Shows the elements of the either the local show mapping(8rpc)
 the specified node. enable: Starts the DTS entity on enable(8dts)
 authorization/ bos setauth: Enables or disables DFS bos setauth(8dfs)
 programs on/ cm setsetuid: Enables or disables setuid cm setsetuid(8dfs)
 status of the Cache/ cm debug: Enables, disables, or reports the cm debug(8dfs)
 /Converts a string into a server encryption key and adds it to a/ bos addkey(8dfs)
 key/ bos lskeys: Displays server encryption key information from a bos lskeys(8dfs)
 /gckeys: Removes obsolete server encryption keys from a key file bos gckeys(8dfs)
 bos rmkey: Removes server encryption keys from a key file bos rmkey(8dfs)
 fms: Determines tape size and end of file (EOF) mark size for a/ fms(8dfs)

from either the local or a remote	endpoint map /specified elements	remove mapping(8rpc)
the either the local or a remote	endpoint map /the elements of	show mapping(8rpc)
the system where the command is	entered /synchronize the clock in	synchronize(8dts)
bak restoredisk: Restores the	entire contents of an aggregate	bak restoredisk(8dfs)
create: Creates the DCE DTS	entity on the specified node	create(8dts)
disable: Stops the DCE DTS	entity on the specified node	disable(8dts)
enable: Starts the DTS	entity on the specified node.	enable(8dts)
in/ synchronize: Causes the DTS	entity to synchronize the clock	synchronize(8dts)
delete: Deletes the DCE DTS	entity	delete(8dts)
characteristics for the DTS	entity set: Modifies	set(8dts)
current information about the DTS	entity show: Displays	show(8dts)
a particular site with their FLDB	entrie /fileset headers at	fts syncserv(8dfs)
/Lists the specifies object	entries (including clearinghouse/	list object(8cds)
UNIX/ /Creates registry database	entries based on information in	passwd_import(8sec)
/lshosts: Lists Tape Coordinator	entries in the Backup Database.	bak lshosts(8dfs)
Unlocks all specified locked	entries in the FLDB /unlockfldb:	fts unlockfldb(8dfs)
Shows information from fileset	entries in the FLDB fts lsfldb:	fts lsfldb(8dfs)
fts syncfldb: Synchronizes FLDB	entries with fileset headers at a/	fts syncfldb(8dfs)
(including clearinghouse object	entries) /object entries	list object(8cds)
/you need to create a cell	entry in either DNS or GDS	show cell(8cds)
bak apropos: Shows each help	entry containing a specified/	bak apropos(8dfs)
/service entry; if the specified	entry does not exist, creates the/	add element(8rpc)
/service entry; if the specified	entry does not exist, creates the/	add member(8rpc)
/a server entry; if the specified	entry does not exist, creates the/	export(8rpc)
/Deletes a fileset family	entry from a fileset family	bak rmfentry(8dfs)
/Removes a Tape Coordinator	entry from the Backup Database	bak rmhost(8dfs)
/Removes the global server	entry from the cell profile	unadvertise(8dts)
fts lsserverentry: Lists a server	entry from the FLDB	fts lsserverentry(8dfs)
delfldbentry: Removes a specified	entry from the FLDB fts	fts delfldbentry(8dfs)
delsrverentry: Deletes a server	entry from the FLDB fts	fts delserverentry(8dfs)
/entry: Removes a name service	entry from the name service/	remove entry(8rpc)
/Defines a fileset family	entry in a fileset family	bak addfentry(8dfs)
fts edsrverentry: Edits a server	entry in the FLDB	fts edsrverentry(8dfs)
fts lock: Locks a fileset	entry in the FLDB	fts lock(8dfs)
fts unlock: Unlocks an	entry in the FLDB	fts unlock(8dfs)
crflbdbentry: Creates a fileset	entry in the FLDB fts	fts crflbdbentry(8dfs)
crsrverentry: Creates a server	entry in the FLDB. fts	fts crsrverentry(8dfs)
addhost: Adds a Tape Coordinator	entry to the Backup Database bak	bak addhost(8dfs)
server by adding the server's	entry to the cell profile /global	advertise(8dts)
add entry: Adds a name service	entry to the name service/	add entry(8rpc)
create object: Creates an object	entry	create object(8cds)
delete object: Deletes an object	entry	delete object(8cds)
attributes) of an object	entry /application-defined	add object(8cds)
attributes) of an object	entry /application:defined	remove object(8cds)
about the specified object	entry /attribute information	show object(8cds)
and an object UUID from a server	entry /binding information	import(8rpc)
LFS fileset and associated FLDB	entry /Creates a read/write DCE	fts create(8dfs)
from the specified name service	entry /elements and the profile	remove profile(8rpc)
entry does not exist, creates the	entry /entry; if the specified	add member(8rpc)
entry does not exist, creates the	entry /entry; if the specified	export(8rpc)
the fileset header and the FLDB	entry /information from both	fts lsft(8dfs)

and object UUIDs in a server entry /interface identifiers, show server(8rpc)
 and object UUIDs from a server entry /interface identifiers, unexport(8rpc)
 from the specified name service entry /members and the group remove group(8rpc)
 attribute of an object entry /modifiable, single:valued set object(8cds)
 NSI attributes of a name service entry show entry: Shows the show entry(8rpc)
 entry does not exist, creates the entry. /entry; if the specified add element(8rpc)
 to the name service database add entry: Adds a name service entry add entry(8rpc)
 entry from the name/ remove entry: Removes a name service remove entry(8rpc)
 of a name service entry show entry: Shows the NSI attributes show entry(8rpc)
 /to a profile in a name service entry; if the specified entry/ add element(8rpc)
 /to a group in a name service entry; if the specified entry/ add member(8rpc)
 object UUIDs or both to a server entry; if the specified entry/ /or export(8rpc)
 local node change: Alters the epoch number and time on the change(8dts)
 set directory to new epoch: Reconstructs a/ set directory to new epoch(8cds)
 process TE_device_name: Lists error messages from the butc TE(4dfs)
 gdsstep: GDS trace evaluation program gdsstep(8gds)
 server process bos getlog: Examines the log file for a bos getlog(8dfs)
 a new master replica or to exclude a replica /to designate set directory to new epoch(8cds)
 server machine bos exec: Executes a command on a bos exec(8dfs)
 machine bos exec: Executes a command on a server bos exec(8dfs)
 that a Tape Coordinator is executing /on the operation bak status(8dfs)
 butc/ TL_device_name: Lists execution information from the TL(4dfs)
 Causes the DTSCP to complete execution quit: quit(8dfs)
 Causes the DTSCP to complete execution. exit: exit(8dts)
 if the specified entry does not exist, creates the entry /entry; add member(8rpc)
 if the specified entry does not exist, creates the entry /entry; export(8rpc)
 if the specified entry does not exist, creates the entry. /entry; add element(8rpc)
 /local server system or makes an existing clearinghouse available create clearinghouse(8cds)
 /replica: Creates a replica of an existing directory in the/ create replica(8cds)
 Sets the expiration date on an existing dump level bak setexp: bak setexp(8dfs)
 complete execution. exit: Causes the DTSCP to exit(8dts)
 dump level bak setexp: Sets the expiration date on an existing bak setexp(8dfs)
 /link and optionally specifies an expiration time and an extension/ create link(8cds)
 information for an interface/ export: Exports binding export(8rpc)
 partitions/ fts lsaggr: Lists all exported aggregates and fts lsaggr(8dfs)
 /about aggregates and partitions exported from a File Server/ fts aggrinfo(8dfs)
 non-LFS partitions that have been exported /DCE LFS aggregates and dfsatab(4dfs)
 non-LFS partitions that can be exported /DCE LFS aggregates and dfstab(4dfs)
 kernel/ fxd: Initializes the File Exporter and start associated fxd(8dfs)
 messages generated by the File Exporter FileLog: Contains FileLog(4dfs)
 an interface identifier/ export: Exports binding information for export(8rpc)
 non-LFS partitions to/ dfsexport: Exports DCE LFS aggregates or dfsexport(8dfs)
 an expiration time and an extension time /specifies create link(8cds)
 from a tape bak scantape: Extracts dump set information bak scantape(8dfs)
 bak lsftfamilies: Lists fileset families defined in the Backup/ bak lsftfamilies(8dfs)
 /dump: Dumps a specific fileset family at a specific dump level bak dump(8dfs)
 bak rmftentry: Deletes a fileset family entry from a fileset/ bak rmftentry(8dfs)
 bak addftentry: Defines a fileset family entry in a fileset family bak addftentry(8dfs)
 bak rmftfamily: Deletes a fileset family from the Backup Database bak rmftfamily(8dfs)
 /Creates a new (empty) fileset family in the Backup Database bak addftfamily(8dfs)
 family entry from a fileset family /Deletes a fileset bak rmftentry(8dfs)

fileset family entry in a fileset	family bak addfentry: Defines a	bak addfentry(8dfs)
/Determines tape size and end of flags to Run in the BosConfig	file (EOF) mark size for a tape/	fms(8dfs)
a new process in the BosConfig	file and in memory /their status	bos start(8dfs)
associated/ fxd: Initializes the messages generated by the	file and start it /Creates	bos create(8dfs)
bos getlog: Examines the log	File Exporter and start	fxd(8dfs)
fileset/ fts restore: Converts a	File Exporter FileLog: Contains	FileLog(4dfs)
Records information about each V	file for a server process	bos getlog(8dfs)
to discard data cached from a	file from a bytestream format to	fts restore(8dfs)
/Manager to write data back to a	file in a disk cache CacheItems:	CacheItems(4dfs)
and partitions exported from a	file or directory /Cache Manager	cm flush(8dfs)
aggregates and partitions on a	File Server and discard the/	cm resetstores(8dfs)
Manager cannot write back to a	File Server machine /aggregates	fts agrinfo(8dfs)
format and places it in the	File Server machine /all exported	fts lsaggr(8dfs)
status flags in the BosConfig	File Server machine /the Cache	cm lsstores(8dfs)
format and place it in a	file system /format to fileset	fts restore(8dfs)
key information from a key	file to NotRun /changing their	bos stop(8dfs)
server encryption keys from a key	file /a fileset to a bytestream	fts dump(8dfs)
key and adds it to a key	file /Displays server encryption	bos lskeys(8dfs)
The server and machine keytab	file /gckeys: Removes obsolete	bos gckeys(8dfs)
status flags in the BosConfig	file /into a server encryption	bos addkey(8dfs)
status flags in the BosConfig	file /krb5/v5srvtab:	v5srvtab(5sec)
processes from the BosConfig	file /without changing their	bos shutdown(8dfs)
a random key and adds it to a key	file /without changing their	bos startup(8dfs)
a new version of a binary	file bos delete: Deletes server	bos delete(8dfs)
server encryption keys from a key	file bos genkey: Generates	bos genkey(8dfs)
the former version of a binary	file bos install: Installs	bos install(8dfs)
The registry database override	file bos rmkey: Removes	bos rmkey(8dfs)
generated by the File Exporter	file bos uninstall: Uses	bos uninstall(8dfs)
cm getdevok: Shows whether device	file passwd_override:	passwd_override(5sec)
/Removes old binary and core	FileLog: Contains messages	FileLog(4dfs)
honored/ /Specifies whether device	files are honored by the Cache/	cm getdevok(8dfs)
filesets containing specified	files from dcelocal/bin and/	bos prune(8dfs)
intro: Introduction to DFS	files on specified filesets are	cm setdevok(8dfs)
in UNIX group and password	files or directories. /from	cm flushfileset(8dfs)
timestamps on versions of binary	files	intro(4dfs)
the DCE Security administrative	files /based on information	passwd_import(8sec)
Creates local password and group	files bos getdates: Lists	bos getdates(8dfs)
/Creates a read/write DCE LFS	files intro: Introduction to	intro(5sec)
/version of a read/write DCE LFS	files passwd_export:	passwd_export(8sec)
lsfdb: Shows information from	fileset and associated FLDB entry	fts create(8dfs)
fts crfldbentry: Creates a	fileset at the local site	fts release(8dfs)
fts lock: Locks a	fileset entries in the FLDB fts	fts lsfdb(8dfs)
Backup/ bak lsftfamilies: Lists	fileset entry in the FLDB	fts crfldbentry(8dfs)
level bak dump: Dumps a specific	fileset entry in the FLDB	fts lock(8dfs)
fileset/ bak rmftentry: Deletes a	fileset families defined in the	bak lsftfamilies(8dfs)
family bak addfentry: Defines a	fileset family at a specific dump	bak dump(8dfs)
bak rmftfamily: Deletes a	fileset family entry from a	bak rmftentry(8dfs)
Database /Creates a new (empty)	fileset family entry in a fileset	bak addfentry(8dfs)
a fileset family entry in a	fileset family from the Backup/	bak rmftfamily(8dfs)
	fileset family in the Backup	bak addftfamily(8dfs)
	fileset family /Defines	bak addfentry(8dfs)

a fileset family entry from a fileset family /Deletes bak rmftentry(8dfs)
 /file from a bytestream format to fileset format and places it in/ fts restore(8dfs)
 fts zap: Removes a DCE LFS fileset from its site without/ fts zap(8dfs)
 /fileset information from both the fileset header and the FLDB entry fts lsft(8dfs)
 site/ fts syncserv: Synchronizes fileset headers at a particular fts syncserv(8dfs)
 /Synchronizes FLDB entries with fileset headers at a particular/ fts syncfdb(8dfs)
 lsheader: Shows information from fileset headers fts fts lsheader(8dfs)
 fileset header/ fts lsft: Lists fileset information from both the fts lsft(8dfs)
 flserver: Initializes the Fileset Location Server (FL/ flserver(8dfs)
 the administrative list for the Fileset Location Server /Contains admin.fl(4dfs)
 messages generated by the Fileset Location Server /Contains FILog(4dfs)
 Displays the status of DCE LFS fileset replicas fts lsreplicas: fts lsreplicas(8dfs)
 flserver: Initializes the Fileset Server ftserver(8dfs)
 the administrative list for the Fileset Server /Contains admin.ft(4dfs)
 messages generated by the Fileset Server FtLog: Contains FtLog(4dfs)
 Reports on the activity of a Fileset Server fts statftserver: fts statftserver(8dfs)
 and place/ fts dump: Converts a fileset to a bytestream format fts dump(8dfs)
 move: Moves a read/write DCE LFS fileset to another site fts fts move(8dfs)
 fts rename: Renames a fileset fts rename(8dfs)
 or backup version of a DCE LFS fileset /a specified read/write fts delete(8dfs)
 site for a read/write DCE LFS fileset /Adds a replication fts addsite(8dfs)
 replicas of a read/write DCE LFS fileset /an immediate update of fts update(8dfs)
 version of a read/write DCE LFS fileset /clone: Creates backup fts clone(8dfs)
 site and read-only DCE LFS fileset /Removes a replication fts rmsite(8dfs)
 quota for a read/write DCE LFS fileset /Sets the maximum fts setquota(8dfs)
 for a read/write DCE LFS fileset /type and parameters fts setrepinfo(8dfs)
 the Backup Database about a fileset bak ftinfo: Queries bak ftinfo(8dfs)
 Creates a mount point for a fileset fts crmount: fts crmount(8dfs)
 /the Cache Manager to update fileset-related information cm checkfilesets(8dfs)
 mappings for filesets accessed/ FilesetItems: Records location FilesetItems(4dfs)
 /Records location mappings for filesets accessed by the Cache/ FilesetItems(4dfs)
 /whether device files on specified filesets are honored by the Cache/ cm setdevok(8dfs)
 points fts lsmount: Lists the filesets associated with mount fts lsmount(8dfs)
 /to discard data cached from filesets containing specified/ cm flushfileset(8dfs)
 bak restoreft: Restores filesets from tape bak restoreft(8dfs)
 Cache Manager/ cm lsstores: Lists filesets that contain data the cm lsstores(8dfs)
 setuid programs on specified filesets /Enables or disables cm setsetuid(8dfs)
 of setuid programs on specified filesets /Shows the status cm getsetuid(8dfs)
 all indicated read/write DCE LFS filesets. /backup versions of fts clonesys(8dfs)
 /after changing their status flags in the BosConfig file to/ bos stop(8dfs)
 /without changing their status flags in the BosConfig file bos shutdown(8dfs)
 /without changing their status flags in the BosConfig file bos startup(8dfs)
 /by changing their status flags to Run in memory without/ bos startup(8dfs)
 file/ /after setting their status flags to Run in the BosConfig bos start(8dfs)
 at a particular site with their FLDB entrie /fileset headers fts syncserv(8dfs)
 at a/ fts syncfdb: Synchronizes FLDB entries with fileset headers fts syncfdb(8dfs)
 DCE LFS fileset and associated FLDB entry /Creates a read/write fts create(8dfs)
 both the fileset header and the FLDB entry /information from fts lsft(8dfs)
 its site without updating the FLDB /a DCE LFS fileset from fts zap(8dfs)
 from fileset entries in the FLDB /lsfdb: Shows information fts lsfdb(8dfs)
 Creates a fileset entry in the FLDB fts crfdbentry: fts crfdbentry(8dfs)

a specified entry from the	FLDB	fts delfldbentry: Removes	fts delfldbentry(8dfs)
Deletes a server entry from the	FLDB	fts delserverentry:	fts delserverentry(8dfs)
Edits a server entry in the	FLDB	fts edserverentry:	fts edserverentry(8dfs)
Locks a fileset entry in the	FLDB	fts lock:	fts lock(8dfs)
Lists a server entry from the	FLDB	fts lserverentry:	fts lserverentry(8dfs)
specified locked entries in the	FLDB	fts unlockfdb: Unlocks all	fts unlockfdb(8dfs)
unlock: Unlocks an entry in the	FLDB	fts	fts unlock(8dfs)
Creates a server entry in the	FLDB.	fts crserverentry:	fts crserverentry(8dfs)
generated by the Fileset/ Location Server (FL Server)	FLLog:	Contains messages	FLLog(4dfs)
to discard data cached from a/ cm	flserver:	Initializes the Fileset	flserver(8dfs)
Manager to discard data/ cm	flush:	Forces the Cache Manager	cm flush(8dfs)
FMSLog: Lists the output of the	flushfileset:	Force the Cache	cm flushfileset(8dfs)
of file (EOF) mark size for a/ fms command	fms command	FMSLog(4dfs)
discard data/ cm flushfileset:	fms:	Determines tape size and end	fms(8dfs)
update/ cm checkfilesets:	FMSLog:	Lists the output of the	FMSLog(4dfs)
discard data cached/ cm flush:	Force the Cache Manager to	cm flushfileset(8dfs)
/a fileset to a bytestream	Forces the Cache Manager to	cm flush(8dfs)
/a bytestream format to fileset	format and place it in a file	fts dump(8dfs)
/Converts a file from a bytestream	format and places it in the file/	fts restore(8dfs)
bos uninstall: Uses the	format to fileset format and/	fts restore(8dfs)
Database about a fileset bak	former version of a binary file	bos uninstall(8dfs)
generated by the Fileset Server	ftinfo: Queries the Backup	bak ftinfo(8dfs)
site for a read/write DCE LFS/ information about aggregates and/ of a read/write DCE LFS fileset	FtLog: Contains messages	FtLog(4dfs)
versions of all indicated/ fts: Introduction to the	fts addsite: Adds a replication	fts addsite(8dfs)
DCE LFS fileset and associated/ fileset entry in the FLDB	fts aggrinfo: Displays disk space	fts aggrinfo(8dfs)
point for a fileset	fts clone: Creates backup version	fts clone(8dfs)
server entry in the FLDB.	fts clonesys: Creates backup	fts clonesys(8dfs)
read/write or backup version of/ specified entry from the FLDB	fts command suite	fts(8dfs)
point	fts create: Creates a read/write	fts create(8dfs)
server entry from the FLDB	fts crfldbentry: Creates a	fts crfldbentry(8dfs)
bytestream format and place it/ entry in the FLDB	fts crmount: Creates a mount	fts crmount(8dfs)
in the FLDB	fts crserverentry: Creates a	fts crserverentry(8dfs)
aggregates and partitions on a/ from fileset entries in the FLDB	fts delete: Removes a specified	fts delete(8dfs)
information from both the/ from fileset headers	fts delfldbentry: Removes a	fts delfldbentry(8dfs)
associated with mount points	fts delmount: Removes a mount	fts delmount(8dfs)
status of DCE LFS fileset/ entry from the FLDB	fts delserverentry: Deletes a	fts delserverentry(8dfs)
LFS fileset to another site	fts dump: Converts a fileset to a	fts dump(8dfs)
Replication by placing a/ a bytestream format to fileset/	fts edserverentry: Edits a server	fts edserverentry(8dfs)
	fts lock: Locks a fileset entry	fts lock(8dfs)
	fts lsaggr: Lists all exported	fts lsaggr(8dfs)
	fts lsfdb: Shows information	fts lsfdb(8dfs)
	fts lsft: Lists fileset	fts lsft(8dfs)
	fts lsheader: Shows information	fts lsheader(8dfs)
	fts lsmount: Lists the filesets	fts lsmount(8dfs)
	fts lsreplicas: Displays the	fts lsreplicas(8dfs)
	fts lserverentry: Lists a server	fts lserverentry(8dfs)
	fts move: Moves a read/write DCE	fts move(8dfs)
	fts release: Initiates Release	fts release(8dfs)
	fts rename: Renames a fileset	fts rename(8dfs)
	fts restore: Converts a file from	fts restore(8dfs)

site and read-only DCE LFS/ quota for a read/write DCE LFS/ replication type and parameters/ activity of a Fileset Server status of a Replication Server/ entries with fileset headers at/ fileset headers at a particular/ the FLDB specified locked entries in the/ update of replicas of a/ fileset from its site without/ command suite Server /specified bak commands or lists Exporter and start associated/ encryption keys from a key/ bos gdad: Starts the /Displays information on gdsstep: processes and processes using a cell entry in either DNS or program transport system interface/ on GDS daemon processes and/ administration program program administration program. BosLog: Contains messages (BOS)/ BosLog: Contains messages SalvageLog: Contains output FileLog: Contains messages Server FileLog: Contains messages FtLog: Contains messages Server RepLog: Contains messages of the/ UpLog: Contains messages it to a key file bos genkey: and adds it to a key file bos size of the cache, the amount/ cm versions of binary files bos files are honored by the/ cm a server process bos restart times for server/ bos setuid programs on specified/ cm /Introduction to commands for /Configures the system as a profile unadvertise: Removes the the specified server/ update: based on information in UNIX Creates local password and /Removes all group members and the

fts rmsite: Removes a replication fts rmsite(8dfs)
 fts setquota: Sets the maximum fts setquota(8dfs)
 fts setrepinfo: Sets or changes fts setrepinfo(8dfs)
 fts statftserver: Reports on the fts statftserver(8dfs)
 fts statrepserver: Displays the fts statrepserver(8dfs)
 fts syncflldb: Synchronizes FLDB fts syncflldb(8dfs)
 fts syncserv: Synchronizes fts syncserv(8dfs)
 fts unlock: Unlocks an entry in fts unlock(8dfs)
 fts unlockflldb: Unlocks all fts unlockflldb(8dfs)
 fts update: Requests an immediate fts update(8dfs)
 fts zap: Removes a DCE LFS fts zap(8dfs)
 fts: Introduction to the fts fts(8dfs)
 ftserver: Initializes the Fileset ftserver(8dfs)
 functional descriptions of all/ bak help(8dfs)
 fxd: Initializes the File fxd(8dfs)
 gckey: Removes obsolete server bos gckey(8dfs)
 GDA daemon gdad(8cdfs)
 gdad: Starts the GDA daemon gdad(8cdfs)
 GDS daemon processes and/ gdsdirinfo(8gdfs)
 GDS trace evaluation program gdsstep(8gdfs)
 GDS /information on GDS daemon gdsdirinfo(8gdfs)
 GDS /you need to create show cell(8cdfs)
 gdscheadm: Cache administration gdscheadm(8gdfs)
 gdscmxi: Controls and edits gdscmxi(8gdfs)
 gdsdirinfo: Displays information gdsdirinfo(8gdfs)
 gdsditadm: Directory database gdsditadm(8gdfs)
 gdsstep: GDS trace evaluation gdsstep(8gdfs)
 gdsysadm: Directory system gdsysadm(8gdfs)
 generated by the Backup Server BakLog(4dfs)
 generated by the Basic OverSeer BosLog(4dfs)
 generated by the DFS Salvager SalvageLog(4dfs)
 generated by the File Exporter FileLog(4dfs)
 generated by the Fileset Location FileLog(4dfs)
 generated by the Fileset Server FtLog(4dfs)
 generated by the Replication RepLog(4dfs)
 generated by the server portion UpLog(4dfs)
 Generates a random key and adds bos genkey(8dfs)
 genkey: Generates a random key bos genkey(8dfs)
 getcachesize: Shows the current cm getcachesize(8dfs)
 getdates: Lists timestamps on bos getdates(8dfs)
 getdevok: Shows whether device cm getdevok(8dfs)
 getlog: Examines the log file for bos getlog(8dfs)
 getrestart: Lists automatic bos getrestart(8dfs)
 getsetuid: Shows the status of cm getsetuid(8dfs)
 Global Directory Service (GDS)/ intro(8gdfs)
 global server by adding the/ advertise(8dts)
 global server entry from the cell unadvertise(8dts)
 Gradually adjusts the clock on update(8dts)
 group and password files /entries passwd_import(8sec)
 group files passwd_export: passwd_export(8sec)
 group from the specified name/ remove group(8rpc)

add member: Adds a member to a	group in a name service entry; if/	add member(8rpc)
the/ remove group: Removes all	group members and the group from	remove group(8rpc)
Removes a specified member from a	group remove member:	remove member(8rpc)
group: Shows the members of a	group show	show group(8rpc)
bos rmdadmin: Removes a user,	group, or server from an/	bos rmdadmin(8dfs)
bos addadmin: Adds a user,	group, or server to an/	bos addadmin(8dfs)
and the group from the/ remove	group: Removes all group members	remove group(8rpc)
group show	group: Shows the members of a	show group(8rpc)
bos lsadmin: Lists the users,	groups, and servers from an/	bos lsadmin(8dfs)
DCE LFS aggregate.	growaggr: Increases the size of a	growaggr(8dfs)
/information from both the fileset	header and the FLDB entry	fts lsft(8dfs)
/syncserv: Synchronizes fileset	headers at a particular site with/	fts syncserv(8dfs)
/FLDB entries with fileset	headers at a particular site	fts syncfdb(8dfs)
Shows information from fileset	headers fts lsheader:	fts lsheader(8dfs)
string bak apropos: Shows each	help entry containing a specified	bak apropos(8dfs)
commands. help: Displays	help information about DTSCP	help(8dts)
or the options of a specified/	help: Displays a list of commands	help(8rpc)
about DTSCP commands.	help: Displays help information	help(8dts)
bak commands or lists/ bak	help: Shows syntax of specified	bak help(8dfs)
bak lsdump: Displays the dump	hierarchy from the Backup/	bak lsdump(8dfs)
Defines a dump level in the dump	hierarchy bak adddump:	bak adddump(8dfs)
/Shows whether device files are	honored by the Cache Manager	cm setdevok(8dfs)
/files on specified filesets are	honored by the Cache Manager	cm setdevok(8dfs)
/information for an interface	identifier or object UUIDs or/	export(8rpc)
a/ /binding information, interface	identifiers, and object UUIDs in	show server(8rpc)
/binding information, interface	identifiers, and object UUIDs/	unexport(8rpc)
fts update: Requests an	immediate update of replicas of a/	fts update(8dfs)
Starts the skulk of a directory	immediately /directory to skulk:	set directory to skulk(8cds)
information and an object UUID/	import: Imports binding	import(8rpc)
an object UUID from a/ import:	Imports binding information and	import(8rpc)
aggregate. growaggr:	Increases the size of a DCE LFS	growaggr(8dfs)
/Creates backup versions of all	indicated read/write DCE LFS/	fts clonesys(8dfs)
checking is disabled NoAuth:	information that DFS authorization	NoAuth(4dfs)
/clearinghouse: Displays attribute	information about the specified/	show clearinghouse(8cds)
show link: Displays attribute	information about the specified/	show link(8cds)
fts agrinfo: Displays disk space	information about aggregates and/	fts agrinfo(8dfs)
help: Displays help	information about DTSCP commands.	help(8dts)
a disk cache CacheItems: Records	information about each V file in	CacheItems(4dfs)
backups bak dumpinfo: Lists	information about specified	bak dumpinfo(8dfs)
/clearinghouse: Displays current	information about the specified/	show cached clearinghouse(8cds)
show clerk: Displays attribute	information about the CDS clerk/	show clerk(8cds)
show: Displays current	information about the DTS entity	show(8dts)
show server: Displays attribute	information about the server/	show server(8cds)
show child: Displays attribute	information about the specified/	show child(8cds)
/directory: Displays attribute	information about the specified/	show directory(8cds)
show object: Displays attribute	information about the specified/	show object(8cds)
show replica: Displays attribute	information about the specified/	show replica(8cds)
from a/ import: Imports binding	information and an object UUID	import(8rpc)
export: Exports binding	information for an interface/	export(8rpc)
/Displays server encryption key	information from a key file	bos lskeys(8dfs)
bak scantape: Extracts dump set	information from a tape	bak scantape(8dfs)

header/	fts lsft: Lists fileset information from both the fileset	fts lsft(8dfs)
in the FLDB	fts lsflldb: Shows information from fileset entries	fts lsflldb(8dfs)
fts lsheader:	Shows information from fileset headers	fts lsheader(8dfs)
TL_device_name:	Lists execution information from the butc process	TL(4dfs)
/database entries based on	information in UNIX group and/	passwd_import(8sec)
/cached server:	Displays address information of a server in the/	show cached server(8cds)
processes/ gdsdirinfo:	Displays information on GDS daemon	gdsdirinfo(8gds)
cell/ show cell:	Displays the information you need to create a	show cell(8cds)
system interface library trace	information /and edits transport	gdscmx1(8gds)
Manager to update fileset-related	information /Forces the Cache	cm checkfilesets(8dfs)
and maintain user authentication	information /pathnames and obtain	dfsbind(8dfs)
show server:	Shows the binding information, interface/	show server(8rpc)
unexport:	Removes binding information, interface/	unexport(8rpc)
Cache/ CacheInfo:	Defines the initial configuration of the	CacheInfo(4dfs)
newaggr:	Initialize a DCE LFS aggregate.	newaggr(8dfs)
process butc:	Initializes a Tape Coordinator	butc(8dfs)
bakserver:	Initializes the Backup Server	bakserver(8dfs)
(BOS) Server process bossserver:	Initializes the Basic OverSeer	bossserver(8dfs)
start related daemons dfsd:	Initializes the Cache Manager and	dfsd(8dfs)
the Update Server upclient:	Initializes the client portion of	upclient(8dfs)
start associated kernel/ fxd:	Initializes the File Exporter and	fxd(8dfs)
Server (FL Server) flserver:	Initializes the Fileset Location	flserver(8dfs)
ftserver:	Initializes the Fileset Server	ftserver(8dfs)
Server process repserver:	Initializes the Replication	repserver(8dfs)
scout:	Initializes the Scout program	scout(8dfs)
the Update Server upserver:	Initializes the server portion of	upserver(8dfs)
placing a read-only/ fts release:	Initiates Release Replication by	fts release(8dfs)
of a binary file bos	install: Installs a new version	bos install(8dfs)
binary file bos install:	Installs a new version of a	bos install(8dfs)
up DCE dce_config:	Installs, configures, and starts	dce_config(8dce)
CDS server cdsclerk:	Manages the interface between clients and the	cdsclerk(8cds)
UUIDs/ /binding information for an	interface identifier or object	export(8rpc)
/Shows the binding information,	interface identifiers, and object/	show server(8rpc)
/Removes binding information,	interface identifiers, and object/	ftexport(8rpc)
and edits transport system	interface library trace/ /Controls	gdscmx1(8gds)
for Global Directory Service/	intro: Introduction to commands	intro(8gds)
daemon and RPC control program/	intro: Introduction to DCE RPC	intro(8rpc)
	intro: Introduction to DFS files	intro(4dfs)
commands	intro: Introduction to the CDS	intro(8cds)
Security administrative files	intro: Introduction to the DCE	intro(5sec)
DTS control program commands	Intro: Introduction to the DCE	intro(8dts)
Security administrative commands	intro: Introduction to the DCE	intro(8sec)
commands	intro: Introduction to the DFS	intro(8dfs)
Global Directory Service/ intro:	Introduction to commands for	intro(8gds)
and RPC control program/ intro:	Introduction to DCE RPC daemon	intro(8rpc)
intro:	Introduction to DFS files	intro(4dfs)
suite bak:	Introduction to the bak command	bak(8dfs)
suite bos:	Introduction to the bos command	bos(8dfs)
intro:	Introduction to the CDS commands	intro(8cds)
suite cm:	Introduction to the cm command	cm(8dfs)
control program commands Intro:	Introduction to the DCE DTS	intro(8dts)

administrative files intro:	Introduction to the DCE Security	intro(5sec)
administrative commands intro:	Introduction to the DCE Security	intro(8sec)
intro:	Introduction to the DFS commands	intro(8dfs)
suite fts:	Introduction to the fts command	fts(8dfs)
/confidence level of clerk calls	issued as a result of CDS control/	set cdscp confidence(8cds)
Exporter and start associated	kernel daemons /the File	fxd(8dfs)
/a string into a server encryption	key and adds it to a key file	bos addkey(8dfs)
bos genkey: Generates a random	key and adds it to a key file	bos genkey(8dfs)
encryption key and adds it to a	key file /a string into a server	bos addkey(8dfs)
encryption key information from a	key file /lskeys: Displays server	bos lskeys(8dfs)
server encryption keys from a	key file /Removes obsolete	bos gckeys(8dfs)
a random key and adds it to a	key file bos genkey: Generates	bos genkey(8dfs)
server encryption keys from a	key file bos rmkey: Removes	bos rmkey(8dfs)
/Displays server encryption	key information from a key file	bos lskeys(8dfs)
obsolete server encryption	keys from a key file /Removes	bos gckeys(8dfs)
rmkey: Removes server encryption	keys from a key file bos	bos rmkey(8dfs)
The server and machine	keytab file /krb5/v5srvtab:	v5srvtab(5sec)
define cached server: Creates	knowledge of a server in the/	define cached server(8cds)
had/ clear cached server: Removes	knowledge of a server that you	clear cached server(8cds)
clear clearinghouse: Removes	knowledge of the specified/	clear clearinghouse(8cds)
database server machines in cells	known to the Cache Manager /Shows	cm lscellinfo(8dfs)
bak labeltape: Creates the	label on a tape	bak labeltape(8dfs)
the name and size from a tape's	label bak readlabel: Displays	bak readlabel(8dfs)
tape bak	labeltape: Creates the label on a	bak labeltape(8dfs)
bak rmdump: Deletes a dump	level from the Backup Database	bak rmdump(8dfs)
bak adddump: Defines a dump	level in the dump hierarchy	bak adddump(8dfs)
/Displays the current confidence	level of clerk calls resulting/	show cdscp confidence(8cds)
/confidence: Sets the confidence	level of clerk calls issued as a/	set cdscp confidence(8cds)
date on an existing dump	level /Sets the expiration	bak setexp(8dfs)
fileset family at a specific dump	level bak dump: Dumps a specific	bak dump(8dfs)
newaggr: Initialize a DCE	LFS aggregate.	newaggr(8dfs)
Increases the size of a DCE	LFS aggregate. growaggr:	growaggr(8dfs)
partitions/ dfsatab: Lists DCE	LFS aggregates and non-LFS	dfsatab(4dfs)
partitions/ dfstab: Specifies DCE	LFS aggregates and non-LFS	dfstab(4dfs)
dfsexport: Exports DCE	LFS aggregates or non-LFS/	dfsexport(8dfs)
/create: Creates a read/write DCE	LFS fileset and associated FLDB/	fts create(8dfs)
/version of a read/write DCE	LFS fileset at the local site	fts release(8dfs)
updating/ fts zap: Removes a DCE	LFS fileset from its site without	fts zap(8dfs)
Displays the status of DCE	LFS fileset replicas /lsreplicas:	fts lsreplicas(8dfs)
fts move: Moves a read/write DCE	LFS fileset to another site	fts move(8dfs)
site and read-only DCE	LFS fileset /a replication	fts rmsite(8dfs)
site for a read/write DCE	LFS fileset /Adds a replication	fts addsite(8dfs)
of replicas of a read/write DCE	LFS fileset /an immediate update	fts update(8dfs)
version of a read/write DCE	LFS fileset /Creates backup	fts clone(8dfs)
parameters for a read/write DCE	LFS fileset /replication type and	fts setrepinfo(8dfs)
quota for a read/write DCE	LFS fileset /Sets the maximum	fts setquota(8dfs)
or backup version of a DCE	LFS fileset /specified read/write	fts delete(8dfs)
of all indicated read/write DCE	LFS filesets. /backup versions	fts clonesys(8dfs)
edits transport system interface	library trace information /and	gdscmxl(8gds)
create link: Creates a soft	link and optionally specifies an/	create link(8cds)
soft links whose names match the	link name that you specify /the	list link(8cds)

delete link: Deletes a soft link delete link(8cds)
 about the specified soft link /attribute information show link(8cds)
 single-valued attribute of a soft link /the value of a modifiable, set link(8cds)
 remove link: Removes a soft link's timeout value attribute remove link(8cds)
 modifiable, single-valued/ set link: Changes the value of a set link(8cds)
 optionally specifies an/ create link: Creates a soft link and create link(8cds)
 delete link: Deletes a soft link delete link(8cds)
 soft links whose names/ list link: Displays a list of all the list link(8cds)
 information about the/ show link: Displays attribute show link(8cds)
 timeout value attribute remove link: Removes a soft link's remove link(8cds)
 /Displays a list of all the soft links whose names match the link/ list link(8cds)
 all the child pointers whose/ list child: Displays a list of list child(8cds)
 list of all the clearinghouses/ list clearinghouse: Displays a list clearinghouse(8cds)
 of all the directories whose/ list directory: Displays a list list directory(8cds)
 /Contains the administrative list for the Backup Server admin.bak(4dfs)
 /Contains the administrative list for the Basic OverSeer (BOS)/ admin.bos(4dfs)
 /Contains the administrative list for the Fileset Location/ admin.fl(4dfs)
 /Contains the administrative list for the Fileset Server admin.ft(4dfs)
 /Contains the administrative list for the Update Server admin.up(4dfs)
 the soft links whose names match/ list link: Displays a list of all list link(8cds)
 object entries (including/ list object: Lists the specifies list object(8cds)
 whose/ list child: Displays a list of all the child pointers list child(8cds)
 list clearinghouse: Displays a list of all the clearinghouses/ list clearinghouse(8cds)
 names/ list directory: Displays a list of all the directories whose list directory(8cds)
 names/ list link: Displays a list of all the soft links whose list link(8cds)
 of a specified/ help: Displays a list of commands or the options help(8rpc)
 or server to an administrative list /Adds a user, group, bos addadmin(8dfs)
 server from an administrative list /Removes a user, group, bos rmadmin(8dfs)
 servers from an administrative list /the users, groups, and bos lsadmin(8dfs)
 FLDB fts lsserverentry: Lists a server entry from the fts lsserverentry(8dfs)
 partitions on a File/ fts lsaggr: Lists all exported aggregates and fts lsaggr(8dfs)
 server processes bos getrestart: Lists automatic restart times for bos getrestart(8dfs)
 non-LFS partitions that/ dfsatab: Lists DCE LFS aggregates and dfsatab(4dfs)
 butc process TE_device_name: Lists error messages from the TE(4dfs)
 the butc process TL_device_name: Lists execution information from TL(4dfs)
 the Backup/ bak lsftfamilies: Lists fileset families defined in bak lsftfamilies(8dfs)
 both the fileset/ fts lsft: Lists fileset information from fts lsft(8dfs)
 the Cache Manager/ cm lsstores: Lists filesets that contain data cm lsstores(8dfs)
 all/ /of specified bak commands or lists functional descriptions of bak help(8dfs)
 backups bak dumpinfo: Lists information about specified bak dumpinfo(8dfs)
 the Backup/ bak lshosts: Lists Tape Coordinator entries in bak lshosts(8dfs)
 with mount points fts lsmount: Lists the filesets associated fts lsmount(8dfs)
 command FMSLog: Lists the output of the fms FMSLog(4dfs)
 entries (including/ list object: Lists the specifies object list object(8cds)
 servers from an/ bos lsadmin: Lists the users, groups, and bos lsadmin(8dfs)
 binary files bos getdates: Lists timestamps on versions of bos getdates(8dfs)
 knowledge of a server in the local clerk's cache /Creates define cached server(8cds)
 information of a server in the local clerk's cache /address show cached server(8cds)
 had specifically defined from the local clerk's cache /that you clear cached server(8cds)
 the epoch number and time on the local node change: Alters change(8dts)
 /elements from either the local or a remote endpoint map remove mapping(8rpc)

/the elements of the either the	local or a remote endpoint map	show mapping(8rpc)
passwd_export: Creates	local password and group files	passwd_export(8sec)
/Creates a clearinghouse on the	local server system or makes an/	create clearinghouse(8cds)
specified clearinghouse from the	local server system /Deletes the	delete clearinghouse(8cds)
read/write DCE LFS fileset at the	local site /version of a	fts release(8dfs)
about the CDS clerk on the	local system /information	show clerk(8cds)
about the server running on the	local system /information	show server(8cds)
the CDS Browser utility on the	local system cdsbrowser: Starts	cdsbrowser(8cds)
clerk: Stops the clerk on the	local system disable	disable clerk(8cds)
server: Stops the server on the	local system disable	disable server(8cds)
accessed/ FilesetItems: Records	location mappings for filesets	FilesetItems(4dfs)
flserver: Initializes the Fileset	Location Server (FL Server)	flserver(8dfs)
list for the Fileset	Location Server /administrative	admin.fl(4dfs)
messages generated by the Fileset	Location Server FLog: Contains	FLog(4dfs)
the FLDB fts	lock: Locks a fileset entry in	fts lock(8dfs)
unlockfdb: Unlocks all specified	locked entries in the FLDB fts	fts unlockfdb(8dfs)
fts lock:	Locks a fileset entry in the FLDB	fts lock(8dfs)
bos getlog: Examines the	log file for a server process	bos getlog(8dfs)
and servers from an/ bos	lsadmin: Lists the users, groups,	bos lsadmin(8dfs)
aggregates and partitions on/ fts	lsaggr: Lists all exported	fts lsaggr(8dfs)
machines in cells known to/ cm	lscellinfo: Shows database server	cm lscellinfo(8dfs)
hierarchy from the Backup/ bak	lsdumps: Displays the dump	bak lsdumps(8dfs)
fileset entries in the FLDB fts	lsfdb: Shows information from	fts lsfdb(8dfs)
from both the fileset header/ fts	lsft: Lists fileset information	fts lsft(8dfs)
families defined in the/ bak	lsftfamilies: Lists fileset	bak lsftfamilies(8dfs)
fileset headers fts	lsheader: Shows information from	fts lsheader(8dfs)
entries in the Backup/ bak	lshosts: Lists Tape Coordinator	bak lshosts(8dfs)
encryption key information/ bos	lskeys: Displays server	bos lskeys(8dfs)
associated with mount points fts	lsmount: Lists the filesets	fts lsmount(8dfs)
of DCE LFS fileset replicas fts	lsreplicas: Displays the status	fts lsreplicas(8dfs)
entry from the FLDB fts	lsserverentry: Lists a server	fts lsserverentry(8dfs)
contain data the Cache/ cm	lsstores: Lists filesets that	cm lsstores(8dfs)
/krb5/v5Srvtab: The server and	machine keytab file	v5Srvtab(5sec)
and partitions on a File Server	machine /all exported aggregates	fts lsaggr(8dfs)
exported from a File Server	machine /and partitions	fts aggrinfo(8dfs)
for all DFS server processes on a	machine /authorization checking	bos setauth(8dfs)
of server processes on a server	machine /Displays the status	bos status(8dfs)
tape drives on a Tape Coordinator	machine /Displays the status	bos status(8dfs)
write back to a File Server	machine /parameters for	TapeConfig(4dfs)
Executes a command on a server	machine /the Cache Manager cannot	cm lsstores(8dfs)
Restarts processes on a server	machine bos exec:	bos exec(8dfs)
/lsccellinfo: Shows database server	machine bos restart:	bos restart(8dfs)
/CDS pathnames and obtain and	machines in cells known to the/	cm lscellinfo(8dfs)
/on the local server system or	maintain user authentication/	dfsbind(8dfs)
dfs: Initializes the Cache	makes an existing clearinghouse/	create clearinghouse(8cds)
File/ /that contain data the Cache	Manager and start related daemons	dfs(8dfs)
reports the status of the Cache	Manager cannot write back to a	cm lsstores(8dfs)
from/ cm flush: Forces the Cache	Manager debugging trace /or	cm debug(8dfs)
cm flushfileset: Force the Cache	Manager to discard data cached	cm flush(8dfs)
/checkfilesets: Forces the Cache	Manager to discard data cached/	cm flushfileset(8dfs)
	Manager to update fileset-related/	cm checkfilesets(8dfs)

/Cancels attempts by the Cache	Manager to write data back to a/	cm resetstores(8dfs)
in cells known to the Cache	Manager /database server machines	cm lscellinfo(8dfs)
configuration of the Cache	Manager /Defines the initial	CacheInfo(4dfs)
filesets are honored by the Cache	Manager /files on specified	cm setdevok(8dfs)
filesets accessed by the Cache	Manager /location mappings for	FilesetItem(4dfs)
files are honored by the Cache	Manager /Shows whether device	cm getdevok(8dfs)
clients and the CDS/ cdsclerk:	Manages the interface between	cdsclerk(8cds)
the local or a remote endpoint	map /elements from either	remove mapping(8rpc)
the local or a remote endpoint	map /the elements of the either	show mapping(8rpc)
elements from either the/ remove	mapping: Removes specified	remove mapping(8rpc)
the either the local or a/ show	mapping: Shows the elements of	show mapping(8rpc)
FilesetItem: Records location	mappings for filesets accessed by/	FilesetItem(4dfs)
/tape size and end of file (EOF)	mark size for a tape drive	fms(8dfs)
/Creates a child pointer at the	master replica of the parent/	create child(8cds)
/allowing you to designate a new	master replica or to exclude a/	set directory to new epoch(8cds)
/of all the soft links whose names	match the link name that you/	list link(8cds)
/the child pointers whose names	match the specified child name	list child(8cds)
/the clearinghouses whose names	match the specified clearinghouse/	list clearinghouse(8cds)
all the directories whose names	match the specified directory/ /of	list directory(8cds)
DCE LFS/ fts setquota: Sets the	maximum quota for a read/write	fts setquota(8dfs)
member: Removes a specified	member from a group remove	remove member(8rpc)
service/ add member: Adds a	member to a group in a name	add member(8rpc)
in a name service entry; if/ add	member: Adds a member to a group	add member(8rpc)
member from a group remove	member: Removes a specified	remove member(8rpc)
remove group: Removes all group	members and the group from the/	remove group(8rpc)
show group: Shows the	members of a group	show group(8rpc)
/their status flags to Run in	memory without changing their/	bos startup(8dfs)
clearinghouse from the server's	memory /of the specified	clear clearinghouse(8cds)
Run in the BosConfig file and in	memory /their status flags to	bos start(8dfs)
TE_device_name: Lists error	messages from the butc process	TE(4dfs)
Server BakLog: Contains	messages generated by the Backup	BakLog(4dfs)
OverSeer (BOS)/ BosLog: Contains	messages generated by the Basic	BosLog(4dfs)
Exporter FileLog: Contains	messages generated by the File	FileLog(4dfs)
Location Server FileLog: Contains	messages generated by the Fileset	FileLog(4dfs)
Server FtLog: Contains	messages generated by the Fileset	FtLog(4dfs)
portion of the/ UpLog: Contains	messages generated by the server	UpLog(4dfs)
Replication/ RepLog: Contains	messages generated by the	RepLog(4dfs)
add directory: Adds a value to a	modifiable, set-valued attribute/	add directory(8cds)
add object: Adds a value to a	modifiable, set:valued attribute/	add object(8cds)
set link: Changes the value of a	modifiable, single-valued/	set link(8cds)
directory: Changes the value of a	modifiable, single-valued/ set	set directory(8cds)
object: Changes the value of a	modifiable, single:valued/ set	set object(8cds)
DTS entity set:	Modifies characteristics for the	set(8dts)
/Defines server processes to be	monitored by the Basic OverSeer/	BosConfig(4dfs)
fts crmount: Creates a	mount point for a fileset	fts crmount(8dfs)
fts delmount: Removes a	mount point	fts delmount(8dfs)
the filesets associated with	mount points fts lsmount: Lists	fts lsmount(8dfs)
fileset to another site fts	move: Moves a read/write DCE LFS	fts move(8dfs)
fileset to another/ fts move:	Moves a read/write DCE LFS	fts move(8dfs)
bak readlabel: Displays the	name and size from a tape's label	bak readlabel(8dfs)
a name service entry from the	name service database /Removes	remove entry(8rpc)

Adds a name service entry to the service/	name service database add entry:	add entry(8rpc)
remove entry: Removes a service/	name service entry from the name	remove entry(8rpc)
add entry: Adds a profile from the specified and the group from the specified	name service entry to the name	add entry(8rpc)
Shows the NSI attributes of a /Adds an element to a profile in a	name service entry /elements and	remove profile(8rpc)
/Adds a member to a group in a links whose names match the link	name service entry /group members	remove group(8rpc)
match the specified clearinghouse names match the specified child	name service entry show entry:	show entry(8rpc)
match the specified directory list of all the soft links whose	name service entry; if the/	add element(8rpc)
/of all the child pointers whose of all the clearinghouses whose	name service entry; if the/	add member(8rpc)
list of all the directories whose or non-LFS partitions to the DCE	name that you specify /the soft	list link(8cds)
Deletes a child pointer from the name /clearinghouses whose names	name /the child pointers whose	list clearinghouse(8cds)
/Displays the information you aggregate.	name /the directories whose names	list child(8cds)
authorization checking is/	names match the link name that/ /a	list link(8cds)
/the clock on the specified server number and time on the local	names match the specified child/	list child(8cds)
DCE DTS entity on the specified DCE DTS entity on the specified	names match the specified/ /a list	list clearinghouse(8cds)
the DTS entity on the specified the DTS entity on the specified	names match the specified/ /a	list directory(8cds)
/Specifies DCE LFS aggregates and namespace /DCE LFS aggregates	namespace delete child:	delete child(8cds)
/Lists DCE LFS aggregates and /Exports DCE LFS aggregates or	need to create a cell entry in/	show cell(8cds)
flags in the BosConfig file to entry show entry: Shows the	newaggr: Initialize a DCE LFS	newaggr(8dfs)
change: Alters the epoch	NoAuth: Indicates that DFS	NoAuth(4dfs)
list object: Lists the specifies	node to the time specified by the/	update(8dts)
entries (including clearinghouse	node change: Alters the epoch	change(8dts)
create object: Creates an	node create: Creates the	create(8dts)
delete object: Deletes an	node disable: Stops the	disable(8dts)
attributes) of an	node. enable: Starts	enable(8dts)
information about the specified	non-LFS partitions that can be/	dfstab(4dfs)
single:valued attribute of an	non-LFS partitions that have been/	dfsatab(4dfs)
/binding information and an	non-LFS partitions to the DCE/	dfsexport(8dfs)
/interface identifiers, and	NotRun /changing their status	bos stop(8dfs)
/interface identifiers, and	NSI attributes of a name service	show entry(8rpc)
/for an interface identifier or	number and time on the local node	change(8dts)
modifiable, set:valued/ add	object entries (including/	list object(8cds)
modifiable, single:valued/ set	object entries) /specifies object	list object(8cds)
create	object entry	create object(8cds)
delete	object entry	delete object(8cds)
information about the/ show	object entry /application-defined	add object(8cds)
object entries (including/ list	object entry /application:defined	remove object(8cds)
set-valued or/ remove	object entry /Displays attribute	show object(8cds)
	object entry /of a modifiable,	set object(8cds)
	object UUID from a server entry	import(8rpc)
	object UUIDs from a server entry	unexport(8rpc)
	object UUIDs in a server entry	show server(8rpc)
	object UUIDs or both to a server/	export(8rpc)
	object: Adds a value to a	add object(8cds)
	object: Changes the value of a	set object(8cds)
	object: Creates an object entry	create object(8cds)
	object: Deletes an object entry	delete object(8cds)
	object: Displays attribute	show object(8cds)
	object: Lists the specifies	list object(8cds)
	object: Removes a value from a	remove object(8cds)

from a key/ bos gckey: Removes obsolete server encryption keys bos gckey(8dfs)
 /Resolves CDS pathnames and obtain and maintain user/ dfsbind(8dfs)
 is/ bak status: Reports on the operation that a Tape Coordinator bak status(8dfs)
 link: Creates a soft link and optionally specifies an/ create create link(8cds)
 /a list of commands or the options of a specified command help(8rpc)
 DFSLog: Contains debugging output from the cm debug command DFSLog(4dfs)
 Salvager SalvageLog: Contains output generated by the DFS SalvageLog(4dfs)
 FMSLog: Lists the output of the fms command FMSLog(4dfs)
 The registry database override file passwd_override: passwd_override(5sec)
 bosserv: Initializes the Basic OverSeer (BOS) Server process bosserv(8dfs)
 messages generated by the Basic OverSeer (BOS) Server /Contains BosLog(4dfs)
 to be monitored by the Basic OverSeer (BOS) Server /processes BosConfig(4dfs)
 administrative list for the Basic OverSeer (BOS) Server /the admin.bos(4dfs)
 /or changes replication type and parameters for a read/write DCE/ fts setrepinfo(8dfs)
 TapeConfig: Defines configuration parameters for tape drives on a/ TapeConfig(4dfs)
 at the master replica of the parent directory /child pointer create child(8cds)
 /Synchronizes fileset headers at a particular site with their FLDB/ fts syncserv(8dfs)
 entries with fileset headers at a particular site /FLDB fts syncfldb(8dfs)
 /information about aggregates and partitions exported from a File/ fts agrinfo(8dfs)
 /Lists all exported aggregates and partitions on a File Server/ fts lsaggr(8dfs)
 /DCE LFS aggregates and non-LFS partitions that can be exported dfstab(4dfs)
 DCE LFS aggregates and non-LFS partitions that have been/ /Lists dfsatab(4dfs)
 /DCE LFS aggregates or non-LFS partitions to the DCE namespace dfsexport(8dfs)
 password and group files passwd_export: Creates local passwd_export(8sec)
 database entries based on/ passwd_import: Creates registry passwd_import(8sec)
 database override file passwd_override: The registry passwd_override(5sec)
 passwd_export: Creates local password and group files passwd_export(8sec)
 information in UNIX group and password files /entries based passwd_import(8sec)
 user/ dfsbind: Resolves CDS pathnames and obtain and maintain dfsbind(8dfs)
 to a bytestream format and place it in a file /a fileset fts dump(8dfs)
 /format to fileset format and places it in the file system fts restore(8dfs)
 /Initiates Release Replication by placing a read-only version of a/ fts release(8dfs)
 fts crmount: Creates a mount point for a fileset fts crmount(8dfs)
 fts delmount: Removes a mount point fts delmount(8dfs)
 create child: Creates a child pointer at the master replica of/ create child(8cds)
 delete child: Deletes a child pointer from the namespace delete child(8cds)
 about the specified child pointer /attribute information show child(8cds)
 /Displays a list of all the child pointers whose names match the/ list child(8cds)
 filesets associated with mount points fts lsmount: Lists the fts lsmount(8dfs)
 upclient: Initializes the client portion of the Update Server upclient(8dfs)
 /messages generated by the server portion of the Update Server UpLog(4dfs)
 upserver: Initializes the server portion of the Update Server upserver(8dfs)
 /clearinghouse: Displays the preferred clearinghouse for/ show cdscp preferred clearinghouse(8cds)
 for/ /clearinghouse: Specifies a preferred clearinghouse to use set cdscp preferred clearinghouse(8cds)
 the preferred/ show cdscp preferred clearinghouse: Displays show cdscp preferred clearinghouse(8cds)
 Specifies a preferred/ set cdscp preferred clearinghouse: set cdscp preferred clearinghouse(8cds)
 rpcd: DCE Remote Procedure Call daemon rpcd(8rpc)
 start/ bos create: Creates a new process in the BosConfig file and bos create(8dfs)
 status of a Replication Server process /Displays the fts statrepserver(8dfs)
 information from the butc process /Lists execution TL(4dfs)
 the log file for a server process bos getlog: Examines bos getlog(8dfs)

the Basic OverSeer (BOS) Server	process bossserver: Initializes	bossserver(8dts)
Initializes a Tape Coordinator	process butc:	butc(8dfs)
the Replication Server	process repserver: Initializes	repserver(8dfs)
error messages from the butc	process TE_device_name: Lists	TE(4dfs)
status flags in/ bos stop: Stops	processes after changing their	bos stop(8dfs)
status flags/ bos start: Starts	processes after setting their	bos start(8dfs)
/information on GDS daemon	processes and processes using GDS	gdsdirinfo(8gds)
status flags/ bos startup: Starts	processes by changing their	bos startup(8dfs)
bos delete: Deletes server	processes from the BosConfig file	bos delete(8dfs)
/checking for all DFS server	processes on a machine	bos setauth(8dfs)
bos restart: Restarts	processes on a server machine	bos restart(8dfs)
/Displays the status of server	processes on a server machine	bos status(8dfs)
/Displays the status of server	processes on a server machine	bos status(8dfs)
/which the BOS Server restarts all	processes or only those with new/	bos setrestart(8dfs)
Basic/ BosConfig: Defines server	processes to be monitored by the	BosConfig(4dfs)
on GDS daemon processes and	processes using GDS /information	gdsdirinfo(8gds)
status flags/ bos shutdown: Stops	processes without changing their	bos shutdown(8dfs)
restart times for server	processes /Lists automatic	bos getrestart(8dfs)
from/ remove profile: Removes all	profile elements and the profile	remove profile(8rpc)
/all profile elements and the	profile from the specified name/	remove profile(8rpc)
add element: Adds an element to a	profile in a name service entry;/	add element(8rpc)
the server's entry to the cell	profile /global server by adding	advertise(8dts)
Removes selected elements from a	profile remove element:	remove element(8rpc)
profile: Shows the elements of a	profile show	show profile(8rpc)
global server entry from the cell	profile unadvertise: Removes the	unadvertise(8dts)
elements and the profile/ remove	profile: Removes all profile	remove profile(8rpc)
profile show	profile: Shows the elements of a	show profile(8rpc)
to the DCE DTS control	program commands /Introduction	intro(8dts)
to DCE RPC daemon and RPC control	program commands /Introduction	intro(8rpc)
calls resulting from CDS control	program commands /level of clerk	show cdscp confidence(8cds)
issued as a result of CDS control	program commands /of clerk calls	set cdscp confidence(8cds)
that result from CDS control	program commands /read requests	set cdscp preferred clearinghouse(8cds)
that result from CDS control	program commands /read requests	show cdscp preferred clearinghouse(8cds)
cdscp: Starts the CDS control	program	cdscp(8cds)
dtscp: DTS control	program	dtscp(8dts)
gdscaheadm: Cache administration	program	gdscaheadm(8gds)
gdsstep: GDS trace evaluation	program	gdsstep(8gds)
rpccp: Starts the RPC control	program	rpccp(8rpc)
scout: Initializes the Scout	program	scout(8dfs)
Directory database administration	program gdsditadm:	gdsditadm(8gds)
Directory system administration	program. gdsysadm:	gdsysadm(8gds)
/Shows the status of setuid	programs on specified filesets	cm getsetuid(8dfs)
/Enables or disables setuid	programs on specified filesets	cm setsetuid(8dfs)
core files from dcelocal/bin/ bos	prune: Removes old binary and	bos prune(8dfs)
a fileset bak ftinfo:	Queries the Backup Database about	bak ftinfo(8dfs)
complete execution	quit: Causes the DTSCP to	quit(8dts)
fts setquota: Sets the maximum	quota for a read/write DCE LFS/	fts setquota(8dfs)
file bos genkey: Generates a	random key and adds it to a key	bos genkey(8dfs)
CDS/ /to use for satisfying	read requests that result from	set cdscp preferred clearinghouse(8cds)
CDS/ /clearinghouse for satisfying	read requests that result from	show cdscp preferred clearinghouse(8cds)
/Removes a replication site and	read-only DCE LFS fileset	fts rmsite(8dfs)

from *a/* delete replica: Deletes a read-only replica of a directory delete replica(8cds)
/Release Replication by placing a read-only version of a read/write/ fts release(8dfs)
 associated/ fts create: Creates a read/write DCE LFS fileset and fts create(8dfs)
/placing a read-only version of a read/write DCE LFS fileset at the/ fts release(8dfs)
 another site fts move: Moves a read/write DCE LFS fileset to fts move(8dfs)
/Adds a replication site for a read/write DCE LFS fileset fts addsite(8dfs)
/Creates backup version of a read/write DCE LFS fileset fts clone(8dfs)
/Sets the maximum quota for a read/write DCE LFS fileset fts setquota(8dfs)
/type and parameters for a read/write DCE LFS fileset fts setrepinfo(8dfs)
 immediate update of replicas of a read/write DCE LFS fileset */an* fts update(8dfs)
/backup versions of all indicated read/write DCE LFS filesets. fts clonesys(8dfs)
 fts delete: Removes a specified read/write or backup version of *a/* fts delete(8dfs)
 size from a tape's label bak readlabel: Displays the name and bak readlabel(8dfs)
 set directory to new epoch: Reconstructs a *directory's/* set directory to new epoch(8cds)
 bak deletedump: Deletes the record of a dump set from *the/* bak deletedump(8dfs)
 file in a disk cache CacheItems: Records information about each *V* CacheItems(4dfs)
 filesets accessed/ FilesetItems: Records location mappings for FilesetItems(4dfs)
/Uses the DFS Salvager to verify, recover, or repair the structure/ salvage(8dfs)
 utility sec_create_db: Registry database creation sec_create_db(8sec)
 on/ passwd_import: Creates registry database entries based passwd_import(8sec)
 passwd_override: The registry database override file passwd_override(5sec)
 rgy_edit: Edits the registry database rgy_edit(8sec)
 the Cache Manager and start related daemons */Initializes* dfsd(8dfs)
read-only/ fts release: Initiates Release Replication by placing a fts release(8dfs)
 Replication by placing *a/* fts release: Initiates Release fts release(8dfs)
 from either the local or a remote endpoint map */elements* remove mapping(8rpc)
 of the either the local or a remote endpoint map */the elements* show mapping(8rpc)
 rpcd: DCE Remote Procedure Call daemon rpcd(8rpc)
 from a set-valued or/ remove directory: Removes a value remove directory(8cds)
 elements from a profile remove element: Removes selected remove element(8rpc)
 service entry from the name/ remove entry: Removes a name remove entry(8rpc)
 members and the group from the/ remove group: Removes all group remove group(8rpc)
 link's timeout value attribute remove link: Removes a soft remove link(8cds)
 elements from either the local/ remove mapping: Removes specified remove mapping(8rpc)
 specified member from a group remove member: Removes a remove member(8rpc)
 from a set-valued or/ remove object: Removes a value remove object(8cds)
 profile elements and the profile/ remove profile: Removes all remove profile(8rpc)
 its site without/ fts zap: Removes a DCE LFS fileset from fts zap(8dfs)
 fts delmount: Removes a mount point fts delmount(8dfs)
 the name service/ remove entry: Removes a name service entry from remove entry(8rpc)
 read-only DCE LFS/ fts rmsite: Removes a replication site and fts rmsite(8dfs)
 value attribute remove link: Removes a soft link's timeout remove link(8cds)
 the FLDB fts delfldbentry: Removes a specified entry from fts delfldbentry(8dfs)
 group remove member: Removes a specified member from a remove member(8rpc)
 backup version of *a/* fts delete: Removes a specified read/write or fts delete(8dfs)
 from the Backup/ bak rmhost: Removes a Tape Coordinator entry bak rmhost(8dfs)
 from *an/ bos* rmadmadmin: Removes a user, group, or server bos rmadmadmin(8dfs)
 or/ remove directory: Removes a value from a set-valued remove directory(8cds)
 or single-valued/ remove object: Removes a value from a set-valued remove object(8cds)
 group from the/ remove group: Removes all group members and the remove group(8rpc)
 the profile from/ remove profile: Removes all profile elements and remove profile(8rpc)

interface identifiers/ unexport:	Removes binding information,	unexport(8rpc)
that you/ clear cached server:	Removes knowledge of a server	clear cached server(8cds)
specified/ clear clearinghouse:	Removes knowledge of the	clear clearinghouse(8cds)
encryption keys from/ bos gkeys:	Removes obsolete server	bos gkeys(8dfs)
from dcelocal/bin and/ bos prune:	Removes old binary and core files	bos prune(8dfs)
profile remove element:	Removes selected elements from a	remove element(8rpc)
from a key file bos rmkey:	Removes server encryption keys	bos rmkey(8dfs)
either the local/ remove mapping:	Removes specified elements from	remove mapping(8rpc)
from the cell/ unadvertise:	Removes the global server entry	unadvertise(8dts)
fts	rename: Renames a fileset	fts rename(8dfs)
fts rename:	Renames a fileset	fts rename(8dfs)
Salvager to verify, recover, or	repair the structure of an/ /DFS	salvage(8dfs)
/replica: Deletes a read-only	replica of a directory from a/	delete replica(8cds)
in/ create replica: Creates a	replica of an existing directory	create replica(8cds)
/a child pointer at the master	replica of the parent directory	create child(8cds)
/you to designate a new master	replica or to exclude a replica	set directory to new epoch(8cds)
/Reconstructs a directory's	replica set, allowing you to/	set directory to new epoch(8cds)
information about the specified	replica /Displays attribute	show replica(8cds)
master replica or to exclude a	replica /you to designate a new	set directory to new epoch(8cds)
existing directory in/ create	replica: Creates a replica of an	create replica(8cds)
replica of a directory/ delete	replica: Deletes a read-only	delete replica(8cds)
information about the/ show	replica: Displays attribute	show replica(8cds)
/Requests an immediate update of	replicas of a read/write DCE LFS/	fts update(8dfs)
the status of DCE LFS fileset	replicas /lsreplicas: Displays	fts lsreplicas(8dfs)
fts release: Initiates Release	Replication by placing a/	fts release(8dfs)
/Displays the status of a	Replication Server process	fts statrepserver(8dfs)
repserver: Initializes the	Replication Server process	repserver(8dfs)
messages generated by the	Replication Server /Contains	RepLog(4dfs)
DCE LFS/ fts rmsite: Removes a	replication site and read-only	fts rmsite(8dfs)
DCE LFS/ fts addsite: Adds a	replication site for a read/write	fts addsite(8dfs)
fts setrepinfo: Sets or changes	replication type and parameters/	fts setrepinfo(8dfs)
generated by the Replication/	RepLog: Contains messages	RepLog(4dfs)
Fileset Server fts statftserver:	Reports on the activity of a	fts statftserver(8dfs)
Tape Coordinator is/ bak status:	Reports on the operation that a	bak status(8dfs)
cm sysname:	Reports or set the CPU/OS type	cm sysname(8dfs)
cm debug: Enables, disables, or	reports the status of the Cache/	cm debug(8dfs)
Replication Server process	repserver: Initializes the	repserver(8dfs)
replicas of a/ fts update:	Requests an immediate update of	fts update(8dfs)
/to use for satisfying read	requests that result from CDS/	set cdscp preferred clearinghouse(8cds)
/clearinghouse for satisfying read	requests that result from CDS/	show cdscp preferred clearinghouse(8cds)
the Cache Manager to write/ cm	resetstores: Cancels attempts by	cm resetstores(8dfs)
and maintain user/ dfsbind:	Resolves CDS pathnames and obtain	dfsbind(8dfs)
bos getrestart: Lists automatic	restart times for server/	bos getrestart(8dfs)
server machine bos	restart: Restarts processes on a	bos restart(8dfs)
/and time at which the BOS Server	restarts all processes or only/	bos setrestart(8dfs)
machine bos restart:	Restarts processes on a server	bos restart(8dfs)
cdsd:	Restarts the CDS server	cdsd(8cds)
dtstd:	Restarts the DTS daemon	dtstd(8dts)
bytestream format to fileset/ fts	restore: Converts a file from a	fts restore(8dfs)
of the Backup Database. bak	restoredb: Restores a backup copy	bak restoredb(8dfs)
contents of an aggregate bak	restoredisk: Restores the entire	bak restoredisk(8dfs)

tape bak restoreft: Restores filesets from bak restoreft(8dfs)
 Backup Database. bak restoredb: Restores a backup copy of the bak restoredb(8dfs)
 bak restoreft: Restores filesets from tape bak restoreft(8dfs)
 an aggregate bak restoredisk: Restores the entire contents of bak restoredisk(8dfs)
 /for satisfying read requests that result from CDS control program/ set cdscp preferred clearinghouse(8cds)
 /for satisfying read requests that result from CDS control program/ show cdscp preferred clearinghouse(8cds)
 /level of clerk calls issued as a result of CDS control program/ set cdscp confidence(8cds)
 /confidence level of clerk calls resulting from CDS control/ show cdscp confidence(8cds)
 database rgy_edit: Edits the registry rgy_edit(8sec)
 or server from an/ bos rmdadmin: Removes a user, group, bos rmdadmin(8dfs)
 the Backup Database bak rmdump: Deletes a dump level from bak rmdump(8dfs)
 family entry from a fileset/ bak rmftentry: Deletes a fileset bak rmftentry(8dfs)
 family from the Backup/ bak rmftfamily: Deletes a fileset bak rmftfamily(8dfs)
 Coordinator entry from the/ bak rmhost: Removes a Tape bak rmhost(8dfs)
 keys from a key file bos rmkey: Removes server encryption bos rmkey(8dfs)
 site and read-only DCE LFS/ fts rmsite: Removes a replication fts rmsite(8dfs)
 /to DCE RPC daemon and RPC control program commands intro(8rpc)
 rpccp: Starts the RPC control program rpccp(8rpc)
 intro: Introduction to DCE RPC daemon and RPC control/ intro(8rpc)
 program rpccp: Starts the RPC control rpccp(8rpc)
 daemon rpcd: DCE Remote Procedure Call rpcd(8rpc)
 /by changing their status flags to Run in memory without changing/ bos startup(8dfs)
 /setting their status flags to Run in the BosConfig file and in/ bos start(8dfs)
 /information about the server running on the local system show server(8cds)
 verify, recover, or repair the/ salvage: Uses the DFS Salvager to salvage(8dfs)
 generated by the DFS Salvager SalvageLog: Contains output SalvageLog(4dfs)
 repair the/ salvage: Uses the DFS Salvager to verify, recover, or salvage(8dfs)
 output generated by the DFS Salvager SalvageLog: Contains SalvageLog(4dfs)
 result/ /clearinghouse to use for satisfying read requests that set cdscp preferred clearinghouse(8cds)
 /the preferred clearinghouse for satisfying read requests that/ show cdscp preferred clearinghouse(8cds)
 the Backup Database bak savedb: Creates a backup copy of bak savedb(8dfs)
 information from a tape bak scantape: Extracts dump set bak scantape(8dfs)
 scout: Initializes the Scout program scout(8dfs)
 program scout: Initializes the Scout scout(8dfs)
 secd: The DCE Security Server secd(8sec)
 intro: Introduction to the DCE Security administrative commands intro(8sec)
 intro: Introduction to the DCE Security administrative files intro(5sec)
 sec_admin: Security administrative tool sec_admin(8sec)
 sec_clientd: Security client daemon sec_clientd(8sec)
 secd: The DCE Security Server secd(8sec)
 administrative tool sec_admin: Security sec_admin(8sec)
 daemon sec_clientd: Security client sec_clientd(8sec)
 creation utility sec_create_db: Registry database sec_create_db(8sec)
 remove element: Removes selected elements from a profile remove element(8rpc)
 /attribute information about the server running on the local/ show server(8cds)
 Initializes the Fileset Location Server (FL Server) flserver: flserver(8dfs)
 /to write data back to a File Server and discard the unstored/ cm resetstores(8dfs)
 /krb5/v5srvtab: The server and machine keytab file v5srvtab(5sec)
 /Configures the system as a global server by adding the server's/ advertise(8dts)
 /addkey: Converts a string into a server encryption key and adds it/ bos addkey(8dfs)
 from a key/ bos lskeys: Displays server encryption key information bos lskeys(8dfs)

file bos rmkey: Removes	server encryption keys from a key	bos rmkey(8dfs)
bos gkeys: Removes obsolete	server encryption keys from a key/	bos gkeys(8dfs)
unadvertise: Removes the global	server entry from the cell/	unadvertise(8dts)
fts delsrverentry: Deletes a	server entry from the FLDB	fts delsrverentry(8dfs)
fts lssrverentry: Lists a	server entry from the FLDB	fts lssrverentry(8dfs)
fts edsrverentry: Edits a	server entry in the FLDB	fts edsrverentry(8dfs)
fts crsrverentry: Creates a	server entry in the FLDB.	fts crsrverentry(8dfs)
and an object UUID from a	server entry /binding information	import(8rpc)
and object UUIDs in a	server entry /identifiers,	show server(8rpc)
and object UUIDs from a	server entry /identifiers,	unexport(8rpc)
/or object UUIDs or both to a	server entry; if the specified/	export(8rpc)
list /Removes a user, group, or	server from an administrative	bos radmin(8dfs)
/server: Creates knowledge of a	server in the local clerk's/	define cached server(8cds)
/Displays address information of a	server in the local clerk's cache	show cached server(8cds)
bos exec: Executes a command on	server machine	bos exec(8dfs)
partitions exported from a File	Server machine /aggregates and	fts aggrinfo(8dfs)
and partitions on a File	Server machine /aggregates	fts lsaggr(8dfs)
status of server processes on a	server machine /Displays the	bos status(8dfs)
status of server processes on a	server machine /Displays the	bos status(8dfs)
cannot write back to a File	Server machine /the Cache Manager	cm lstores(8dfs)
restart: Restarts processes on a	server machine bos	bos restart(8dfs)
cm lscellinfo: Shows database	server machines in cells known to/	cm lscellinfo(8dfs)
by/ /the clock on the specified	server node to the time specified	update(8dts)
disable server: Stops the	server on the local system	disable server(8cds)
Server /messages generated by the	server portion of the Update	UpLog(4dfs)
Server upserver: Initializes the	server portion of the Update	upserver(8dfs)
status of a Replication	Server process /Displays	fts statrepsrver(8dfs)
the Basic OverSeer (BOS)	Server process /Initializes	bossrver(8dfs)
Examines the log file for a	server process bos getlog:	bos getlog(8dfs)
Initializes the Replication	Server process repserver:	repserver(8dfs)
BosConfig/ bos delete: Deletes	server processes from the	bos delete(8dfs)
/checking for all DFS	server processes on a machine	bos setauth(8dfs)
status: Displays the status of	server processes on a server/ bos	bos status(8dfs)
status: Displays the status of	server processes on a server/ bos	bos status(8dfs)
by the Basic/ BosConfig: Defines	server processes to be monitored	BosConfig(4dfs)
Lists automatic restart times for	server processes bos getrestart:	bos getrestart(8dfs)
/date and time at which the BOS	Server restarts all processes or/	bos setrestart(8dfs)
/a clearinghouse on the local	server system or makes an/	create clearinghouse(8cds)
clearinghouse from the local	server system /the specified	delete clearinghouse(8cds)
/server: Removes knowledge of a	server that you had specifically/	clear cached server(8cds)
/addadmin: Adds a user, group, or	server to an administrative list	bos addadmin(8dfs)
bakserver: Initializes the Backup	Server	bakserver(8dfs)
cdsd: Restarts the CDS	server	cdsd(8cds)
fts server: Initializes the Fileset	Server	fts server(8dfs)
secd: The DCE Security	Server	secd(8sec)
between clients and the CDS	server /Manages the interface	cdsclerk(8cds)
the server portion of the Update	Server /messages generated by	UpLog(4dfs)
by the Basic OverSeer (BOS)	Server /messages generated	BosLog(4dfs)
by the Basic OverSeer (BOS)	Server /processes to be monitored	BosConfig(4dfs)
list for the Backup	Server /the administrative	admin.bak(4dfs)
list for the Basic OverSeer (BOS)	Server /the administrative	admin.bos(4dfs)

list for the Fileset Location	Server /the administrative	admin.fl(4dfs)
list for the Fileset	Server /the administrative	admin.ft(4dfs)
list for the Update	Server /the administrative	admin.up(4dfs)
messages generated by the Backup	Server BakLog: Contains	BakLog(4dfs)
generated by the Fileset Location	Server FILog: Contains messages	FILog(4dfs)
messages generated by the Fileset	Server FtLog: Contains	FtLog(4dfs)
on the activity of a Fileset	Server fts statftserver: Reports	fts statftserver(8dfs)
generated by the Replication	Server RepLog: Contains messages	RepLog(4dfs)
the client portion of the Update	Server upclient: Initializes	upclient(8dfs)
server portion of the Update	Server upserver: Initializes	upserver(8dfs)
/as a global server by adding the	server's entry to the cell/	advertise(8dts)
specified clearinghouse from the	server's memory /knowledge of the	clear clearinghouse(8cds)
the Fileset Location Server (FL	Server) flserver: Initializes	flserver(8dfs)
server in the/ define cached	server: Creates knowledge of a	define cached server(8cds)
information of a/ show cached	server: Displays address	show cached server(8cds)
information about the/ show	server: Displays attribute	show server(8cds)
server that you/ clear cached	server: Removes knowledge of a	clear cached server(8cds)
information, interface/ show	server: Shows the binding	show server(8rpc)
local system disable	server: Stops the server on the	disable server(8cds)
/Lists the users, groups, and	servers from an administrative/	bos lsadmin(8dfs)
/to commands for Global Directory	Service (GDS) administration	intro(8gds)
name service entry from the name	service database /Removes a	remove entry(8rpc)
a name service entry to the name	service database add entry: Adds	add entry(8rpc)
remove entry: Removes a name	service entry from the name/	remove entry(8rpc)
database add entry: Adds a name	service entry to the name service	add entry(8rpc)
profile from the specified name	service entry /elements and the	remove profile(8rpc)
the group from the specified name	service entry /group members and	remove group(8rpc)
the NSI attributes of a name	service entry show entry: Shows	show entry(8rpc)
/an element to a profile in a name	service entry; if the specified/	add element(8rpc)
/a member to a group in a name	service entry; if the specified/	add member(8rpc)
confidence level of clerk calls/	set cdscp confidence: Sets the	set cdscp confidence(8cds)
clearinghouse: Specifies a/	set cdscp preferred	set cdscp preferred clearinghouse(8cds)
Reconstructs a directory's/	set directory to new epoch:	set directory to new epoch(8cds)
the skulk of a directory/	set directory to skulk: Starts	set directory to skulk(8cds)
of a modifiable, single-valued/	set directory: Changes the value	set directory(8cds)
/Deletes the record of a dump	set from the Backup Database	bak deletedump(8dfs)
bak scantape: Extracts dump	set information from a tape	bak scantape(8dfs)
modifiable, single-valued/	set link: Changes the value of a	set link(8cds)
a modifiable, single-valued/	set object: Changes the value of	set object(8cds)
cm sysname: Reports or	set the CPU/OS type	cm sysname(8dfs)
new/ /a directory's replica	set, allowing you to designate a	set directory to new epoch(8cds)
/Adds a value to a modifiable,	set-valued attribute (including/	add directory(8cds)
/directory: Removes a value from a	set-valued or single-valued/	remove directory(8cds)
/object: Removes a value from a	set-valued or single-valued/	remove object(8cds)
the DTS entity	set: Modifies characteristics for	set(8dts)
/Adds a value to a modifiable,	set:valued attribute (including/	add object(8cds)
authorization checking for/ bos	setauth: Enables or disables DFS	bos setauth(8dfs)
disk cache cm	setcachesize: Sets the size of a	cm setcachesize(8dfs)
device files on specified/ cm	setdevok: Specifies whether	cm setdevok(8dfs)
on an existing dump level bak	setexp: Sets the expiration date	bak setexp(8dfs)
for a read/write DCE LFS/ fts	setquota: Sets the maximum quota	fts setquota(8dfs)

replication type and/	fts	setrepinfo: Sets or changes	fts setrepinfo(8dfs)
time at which the BOS Server/ bos		setrestart: Sets the date and	bos setrestart(8dfs)
and parameters/ fts setrepinfo:		Sets or changes replication type	fts setrepinfo(8dfs)
clerk/ set cdscp confidence:		Sets the confidence level of	set cdscp confidence(8cds)
the BOS Server/ bos setrestart:		Sets the date and time at which	bos setrestart(8dfs)
existing dump level bak setexp:		Sets the expiration date on an	bak setexp(8dfs)
read/write DCE LFS/ fts setquota:		Sets the maximum quota for a	fts setquota(8dfs)
cm setcachesize:		Sets the size of a disk cache	cm setcachesize(8dfs)
setuid programs on specified/ cm		setsetuid: Enables or disables	cm setsetuid(8dfs)
bos start: Starts processes after		setting their status flags to Run/	bos start(8dfs)
cm getsetuid: Shows the status of		setuid programs on specified/	cm getsetuid(8dfs)
cm setsetuid: Enables or disables		setuid programs on specified/	cm setsetuid(8dfs)
Displays current information/		show cached clearinghouse:	show cached clearinghouse(8cds)
address information of a server/		show cached server: Displays	show cached server(8cds)
the current confidence level of/		show cdscp confidence: Displays	show cdscp confidence(8cds)
clearinghouse: Displays the/		show cdscp preferred	show cdscp preferred clearinghouse(8cds)
information you need to create a/		show cell: Displays the	show cell(8cds)
information about the specified/		show child: Displays attribute	show child(8cds)
attribute information about the/		show clearinghouse: Displays	show clearinghouse(8cds)
information about the CDS clerk/		show clerk: Displays attribute	show clerk(8cds)
attribute information about the/		show directory: Displays	show directory(8cds)
attributes of a name service/		show entry: Shows the NSI	show entry(8rpc)
a group		show group: Shows the members of	show group(8rpc)
information about the specified/		show link: Displays attribute	show link(8cds)
of the either the local or a/		show mapping: Shows the elements	show mapping(8rpc)
information about the specified/		show object: Displays attribute	show object(8cds)
of a profile		show profile: Shows the elements	show profile(8rpc)
information about the specified/		show replica: Displays attribute	show replica(8cds)
information about the server/		show server: Displays attribute	show server(8cds)
information, interface/		show server: Shows the binding	show server(8rpc)
information about the DTS entity		show: Displays current	show(8dts)
cells known to/ cm lscellinfo:		Shows database server machines in	cm lscellinfo(8dfs)
a specified string bak apropos:		Shows each help entry containing	bak apropos(8dfs)
entries in the FLDB fts lsfldb:		Shows information from fileset	fts lsfldb(8dfs)
headers fts lsheader:		Shows information from fileset	fts lsheader(8dfs)
commands or lists/ bak help:		Shows syntax of specified bak	bak help(8dfs)
interface/ show server:		Shows the binding information,	show server(8rpc)
cache, the/ cm getcachesize:		Shows the current size of the	cm getcachesize(8dfs)
show profile:		Shows the elements of a profile	show profile(8rpc)
the local or a/ show mapping:		Shows the elements of the either	show mapping(8rpc)
show group:		Shows the members of a group	show group(8rpc)
name service entry show entry:		Shows the NSI attributes of a	show entry(8rpc)
programs on/ cm getsetuid:		Shows the status of setuid	cm getsetuid(8dfs)
honored by the/ cm getdevok:		Shows whether device files are	cm getdevok(8dfs)
changing their status flags/ bos		shutdown: Stops processes without	bos shutdown(8dfs)
/the value of a modifiable,		single-valued attribute of a/	set directory(8cds)
a value from a set-valued or		single-valued attribute/ /Removes	remove directory(8cds)
link /the value of a modifiable,		single-valued attribute of a soft	set link(8cds)
a value from a set-valued or		single-valued attribute/ /Removes	remove object(8cds)
/the value of a modifiable,		single-valued attribute of an/	set object(8cds)
fts rmsite: Removes a replication		site and read-only DCE LFS/	fts rmsite(8dfs)

fts addsite: Adds a replication site for a read/write DCE LFS/ fts addsite(8dfs)
 /fileset headers at a particular site with their FLDB entry fts syncserv(8dfs)
 /a DCE LFS fileset from its site without updating the FLDB fts zap(8dfs)
 fileset headers at a particular site /FLDB entries with fts syncfldb(8dfs)
 DCE LFS fileset to another site /move: Moves a read/write fts move(8dfs)
 DCE LFS fileset at the local site /version of a read/write fts release(8dfs)
 size for a/ fms: Determines tape size and end of file (EOF) mark fms(8dfs)
 size and end of file (EOF) mark size for a tape drive /tape fms(8dfs)
 readlabel: Displays the name and size from a tape's label bak bak readlabel(8dfs)
 growaggr: Increases the size of a DCE LFS aggregate. growaggr(8dfs)
 cm setcachesize: Sets the size of a disk cache cm setcachesize(8dfs)
 /getcachesize: Shows the current size of the cache, the amount of/ cm getcachesize(8dfs)
 /directory to skulk: Starts the skulk of a directory immediately set directory to skulk(8cds)
 directory/ set directory to skulk: Starts the skulk of a set directory to skulk(8cds)
 speciefies/ create link: Creates a soft link and optionally create link(8cds)
 delete link: Deletes a soft link delete link(8cds)
 information about the specified soft link /Displays attribute show link(8cds)
 single-valued attribute of a soft link /value of a modifiable, set link(8cds)
 attribute remove link: Removes a soft link's timeout value remove link(8cds)
 /link: Displays a list of all the soft links whose names match the/ list link(8cds)
 Starts the advertisement daemon cdsadv: cdsadv(8cds)
 fts aggrinfo: Displays disk space information about/ fts aggrinfo(8dfs)
 a specific fileset family at a specific dump level /dump: Dumps bak dump(8dfs)
 specific dump/ bak dump: Dumps a specific fileset family at a bak dump(8dfs)
 local/ /of a server that you had specifically defined from the clear cached server(8cds)
 attribute information about the specified replica /Displays show replica(8cds)
 dumpinfo: Lists information about specified backups bak bak dumpinfo(8dfs)
 bak help: Shows syntax of specified bak commands or lists/ bak help(8dfs)
 specified server node to the time specified by the argument /on the update(8dts)
 /current information about the specified cached clearinghouse show cached clearinghouse(8cds)
 pointers whose names match the specified child name /the child list child(8cds)
 attribute information about the specified child pointer /Displays show child(8cds)
 /Removes knowledge of the specified clearinghouse from the/ clear clearinghouse(8cds)
 delete clearinghouse: Deletes the specified clearinghouse from the/ delete clearinghouse(8cds)
 /whose names match the specified clearinghouse name list clearinghouse(8cds)
 attribute information about the specified clearinghouse /Displays show clearinghouse(8cds)
 of an existing directory in the specified clearinghouse /replica create replica(8cds)
 the of commands or the options of a specified command /a list help(8rpc)
 directories whose names match the specified directory name /all the list directory(8cds)
 attribute information about the specified directory /Displays show directory(8cds)
 the/ remove mapping: Removes specified elements from either remove mapping(8rpc)
 /in a name service entry; if the specified entry does not exist,/ add element(8rpc)
 /in a name service entry; if the specified entry does not exist,/ add member(8rpc)
 /or both to a server entry; if the specified entry does not exist,/ export(8rpc)
 fts delfldbentry: Removes a specified entry from the FLDB fts delfldbentry(8dfs)
 /cached from filesets containing specified files or directories. cm flushfileset(8dfs)
 /Specifies whether device files on specified filesets are honored by/ cm setdevok(8dfs)
 or disables setuid programs on specified filesets /Enables cm setsetuid(8dfs)
 the status of setuid programs on specified filesets /Shows cm getsetuid(8dfs)
 FLDB fts unlockfldb: Unlocks all specified locked entries in the fts unlockfldb(8dfs)
 remove member: Removes a specified member from a group remove member(8rpc)

/members and the group from the	specified name service entry	remove group(8rpc)
/elements and the profile from the	specified name service entry	remove profile(8rpc)
Creates the DCE DTS entity on the	specified node create:	create(8dts)
Stops the DCE DTS entity on the	specified node disable:	disable(8dts)
Starts the DTS entity on the	specified node. enable:	enable(8dts)
attribute information about the	specified object entry /Displays	show object(8cds)
version of/ fts delete: Removes a	specified read/write or backup	fts delete(8dfs)
/adjusts the clock on the	specified server node to the time/	update(8dts)
attribute information about the	specified soft link /Displays	show link(8cds)
each help entry containing a	specified string /apropos: Shows	bak apropos(8dfs)
cdscp preferred clearinghouse:	Specifies a preferred/ set	set cdscp preferred clearinghouse(8cds)
an/ /a soft link and optionally	specifies an expiration time and	create link(8cds)
non-LFS partitions that/ dfstab:	Specifies DCE LFS aggregates and	dfstab(4dfs)
list object: Lists the	specified object entries/	list object(8cds)
specified filesets/ cm setdevok:	Specifies whether device files on	cm setdevok(8dfs)
match the link name that you	specify /soft links whose names	list link(8cds)
/Initializes the File Exporter and	start associated kernel daemons	fxd(8dfs)
process in the BosConfig file and	start it /create: Creates a new	bos create(8dfs)
Initializes the Cache Manager and	start related daemons dfds:	dfds(8dfs)
setting their status flags/ bos	start: Starts processes after	bos start(8dfs)
their status flags to/ bos start:	Starts processes after setting	bos start(8dfs)
their status flags/ bos startup:	Starts processes by changing	bos startup(8dfs)
solicitation daemon cdsadv:	Starts the advertisement and	cdsadv(8cds)
the local system cdsbrowser:	Starts the CDS Browser utility on	cdsbrowser(8cds)
cdscp:	Starts the CDS control program	cdscp(8cds)
specified node. enable:	Starts the DTS entity on the	enable(8dts)
gdad:	Starts the GDA daemon	gdad(8cds)
rpccp:	Starts the RPC control program	rpccp(8rpc)
set directory to skulk:	Starts the skulk of a directory/	set directory to skulk(8cds)
Installs, configures, and	starts up DCE dce_config:	dce_config(8dce)
changing their status flags/ bos	startup: Starts processes by	bos startup(8dfs)
activity of a Fileset Server fts	statfsrserver: Reports on the	fts statfsrserver(8dfs)
status of a Replication/ fts	statrepserver: Displays the	fts statrepserver(8dfs)
/processes without changing their	status flags in the BosConfig/	bos shutdown(8dfs)
/in memory without changing their	status flags in the BosConfig/	bos startup(8dfs)
/processes after changing their	status flags in the BosConfig/	bos stop(8dfs)
/processes by changing their	status flags to Run in memory/	bos startup(8dfs)
/processes after setting their	status flags to Run in the/	bos start(8dfs)
fts statrepserver: Displays the	status of a Replication Server/	fts statrepserver(8dfs)
fts lsreplicas: Displays the	status of DCE LFS fileset/	fts lsreplicas(8dfs)
server/ bos status: Displays the	status of server processes on a	bos status(8dfs)
server/ bos status: Displays the	status of server processes on a	bos status(8dfs)
cm getsetuid: Shows the	status of setuid programs on/	cm getsetuid(8dfs)
bak verifydb: Checks the	status of the Backup Database	bak verifydb(8dfs)
/Enables, disables, or reports the	status of the Cache Manager/	cm debug(8dfs)
server processes on a server/ bos	status: Displays the status of	bos status(8dfs)
server processes on a server/ bos	status: Displays the status of	bos status(8dfs)
that a Tape Coordinator is/ bak	status: Reports on the operation	bak status(8dfs)
changing their status flags/ bos	stop: Stops processes after	bos stop(8dfs)
their status flags in/ bos stop:	Stops processes after changing	bos stop(8dfs)
their status flags/ bos shutdown:	Stops processes without changing	bos shutdown(8dfs)

system disable clerk: Stops the clerk on the local disable clerk(8cds)
 specified node disable: Stops the DCE DTS entity on the disable(8dts)
 system disable server: Stops the server on the local disable server(8cds)
 key and/ bos addkey: Converts a string into a server encryption bos addkey(8dfs)
 help entry containing a specified string bak apropos: Shows each bak apropos(8dfs)
 /to verify, recover, or repair the structure of an aggregate salvage(8dfs)
 Introduction to the bak command suite bak: bak(8dfs)
 Introduction to the bos command suite bos: bos(8dfs)
 Introduction to the cm command suite cm: cm(8dfs)
 Introduction to the fts command suite fts: fts(8dfs)
 entries with fileset headers/ fts syncfdb: Synchronizes FLDB fts syncfdb(8dfs)
 system/ /Causes the DTS entity to synchronize the clock in the synchronize(8dts)
 entity to synchronize the clock/ synchronize: Causes the DTS synchronize(8dts)
 particular site/ fts syncserv: Synchronizes fileset headers at a fts syncserv(8dfs)
 fileset headers at/ fts syncfdb: Synchronizes FLDB entries with fts syncfdb(8dfs)
 headers at a particular site/ fts syncserv: Synchronizes fileset fts syncserv(8dfs)
 or lists/ bak help: Shows syntax of specified bak commands bak help(8dfs)
 CPU/OS type cm sysname: Reports or set the cm sysname(8dfs)
 gdsysadm: Directory system administration program. gdsysadm(8gds)
 adding/ advertise: Configures the system as a global server by advertise(8dts)
 /Controls and edits transport system interface library trace/ gdscml(8gds)
 /on the local server system or makes an existing/ create clearinghouse(8cds)
 /to synchronize the clock in the system where the command is/ synchronize(8dts)
 about the CDS clerk on the local system /attribute information show clerk(8cds)
 format and places it in the file system /format to fileset fts restore(8dfs)
 the server running on the local system /information about show server(8cds)
 from the local server system /specified clearinghouse delete clearinghouse(8cds)
 CDS Browser utility on the local system cdsbrowser: Starts the cdsbrowser(8cds)
 Stops the clerk on the local system disable clerk: disable clerk(8cds)
 Stops the server on the local system disable server: disable server(8cds)
 Backup/ bak lshosts: Lists Tape Coordinator entries in the bak lshosts(8dfs)
 Backup/ bak rmhost: Removes a Tape Coordinator entry from the bak rmhost(8dfs)
 Backup/ bak addhost: Adds a Tape Coordinator entry to the bak addhost(8dfs)
 /Reports on the operation that a Tape Coordinator is executing bak status(8dfs)
 /parameters for tape drives on a Tape Coordinator machine TapeConfig(4dfs)
 butc: Initializes a Tape Coordinator process butc(8dfs)
 end of file (EOF) mark size for a tape drive /tape size and fms(8dfs)
 /configuration parameters for tape drives on a Tape Coordinator/ TapeConfig(4dfs)
 mark size for a/ fms: Determines tape size and end of file (EOF) fms(8dfs)
 dump set information from a tape bak scantape: Extracts bak scantape(8dfs)
 labeltape: Creates the label on a tape bak bak labeltape(8dfs)
 restoreft: Restores filesets from tape bak bak restoreft(8dfs)
 Displays the name and size from a tape's label bak readlabel: bak readlabel(8dfs)
 parameters for tape drives on a/ TapeConfig: Defines configuration TapeConfig(4dfs)
 messages from the butc process TE_device_name: Lists error TE(4dfs)
 link: Removes a soft link's timeout value attribute remove remove link(8cds)
 /Lists automatic restart times for server processes bos getrestart(8dfs)
 files bos getdates: Lists timestamps on versions of binary bos getdates(8dfs)
 information from the butc/ TL_device_name: Lists execution TL(4dfs)
 Security administrative tool sec_admin: sec_admin(8sec)
 gdsstep: GDS trace evaluation program gdsstep(8gds)

system interface library	trace information /transport	gdscmxl(8gds)
of the Cache Manager debugging	trace /or reports the status	cm debug(8dfs)
gdscmxl: Controls and edits	transport system interface/	gdscmxl(8gds)
/Sets or changes replication	type and parameters for a/	fts setrepinfo(8dfs)
amount of cache in use, and the	type of cache /of the cache, the	cm getcachesize(8dfs)
Reports or set the CPU/OS	type cm sysname:	cm sysname(8dfs)
server entry from the cell/	unadvertise: Removes the global	unadvertise(8dts)
information, interface/	unexport: Removes binding	unexport(8rpc)
version of a binary file bos	uninstall: Uses the former	bos uninstall(8dfs)
/entries based on information in	UNIX group and password files	passwd_import(8sec)
FLDB fts	unlock: Unlocks an entry in the	fts unlock(8dfs)
locked entries in the FLDB fts	unlockfldb: Unlocks all specified	fts unlockfldb(8dfs)
entries in the/ fts unlockfldb:	Unlocks all specified locked	fts unlockfldb(8dfs)
fts unlock:	Unlocks an entry in the FLDB	fts unlock(8dfs)
to a File Server and discard the	unstored data /to write data back	cm resetstores(8dfs)
portion of the Update Server	upclient: Initializes the client	upclient(8dfs)
/Forces the Cache Manager to	update fileset-related/	cm checkfilesets(8dfs)
update: Requests an immediate	update of replicas of a/	fts update(8dfs)
the client portion of the	Update Server /Initializes	upclient(8dfs)
the server portion of the	Update Server /Initializes	upserver(8dfs)
by the server portion of the	Update Server /messages generated	UpLog(4dfs)
administrative list for the	Update Server admin.up: Contains	admin.up(4dfs)
clock on the specified server/	update: Gradually adjusts the	update(8dts)
update of replicas of a/ fts	update: Requests an immediate	fts update(8dfs)
LFS fileset from its site without	updating the FLDB /Removes a DCE	fts zap(8dfs)
generated by the server portion/	UpLog: Contains messages	UpLog(4dfs)
portion of the Update Server	upserver: Initializes the server	upserver(8dfs)
/a preferred clearinghouse to	use for satisfying read requests/	set cdscp preferred clearinghouse(8cds)
the cache, the amount of cache in	use, and the type of cache /of	cm getcachesize(8dfs)
/pathnames and obtain and maintain	user authentication information	dfsbind(8dfs)
bos rmdadmin: Removes a	user, group, or server from an/	bos rmdadmin(8dfs)
bos addadmin: Adds a	user, group, or server to an/	bos addadmin(8dfs)
an/ bos lsadmin: Lists the	users, groups, and servers from	bos lsadmin(8dfs)
recover, or repair the/ salvage:	Uses the DFS Salvager to verify,	salvage(8dfs)
binary file bos uninstall:	Uses the former version of a	bos uninstall(8dfs)
daemon processes and processes	using GDS /information on GDS	gdsdirinfo(8gds)
/Starts the CDS Browser	utility on the local system	cdsbrowser(8cds)
Registry database creation	utility sec_create_db:	sec_create_db(8sec)
binding information and an object	UUID from a server entry /Imports	import(8rpc)
/interface identifiers, and object	UUIDs from a server entry	unexport(8rpc)
/interface identifiers, and object	UUIDs in a server entry	show server(8rpc)
/an interface identifier or object	UUIDs or both to a server entry;/	export(8rpc)
/Records information about each	V file in a disk cache	CacheItems(4dfs)
Removes a soft link's timeout	value attribute remove link:	remove link(8cds)
remove directory: Removes a	value from a set-valued or/	remove directory(8cds)
remove object: Removes a	value from a set-valued or/	remove object(8cds)
set directory: Changes the	value of a modifiable,/	set directory(8cds)
set link: Changes the	value of a modifiable,/	set link(8cds)
set object: Changes the	value of a modifiable,/	set object(8cds)
attribute/ add directory: Adds a	value to a modifiable, set-valued	add directory(8cds)
attribute/ add object: Adds a	value to a modifiable, set:valued	add object(8cds)

salvage: Uses the DFS Salvager to verify, recover, or repair the/ salvage(8dfs)
 the Backup Database bak verifydb: Checks the status of bak verifydb(8dfs)
 bos install: Installs a new version of a binary file bos install(8dfs)
 bos uninstall: Uses the former version of a binary file bos uninstall(8dfs)
 /a specified read/write or backup version of a DCE LFS fileset fts delete(8dfs)
 fileset/ /by placing a read-only version of a read/write DCE LFS fts release(8dfs)
 fts clone: Creates backup version of a read/write DCE LFS/ fts clone(8dfs)
 fts clonesys: Creates backup versions of all indicated/ fts clonesys(8dfs)
 bos getdates: Lists timestamps on versions of binary files bos getdates(8dfs)
 cached in a disk cache Vn: Contains a chunk of data Vn(4dfs)
 by the Cache/ cm getdevok: Shows whether device files are honored cm getdevok(8dfs)
 filesets/ cm setdevok: Specifies whether device files on specified cm setdevok(8dfs)
 /a list of all the soft links whose names match the link name/ list link(8cds)
 /a list of all the child pointers whose names match the specified/ list child(8cds)
 /a list of all the clearinghouses whose names match the specified/ list clearinghouse(8cds)
 /a list of all the directories whose names match the specified/ list directory(8cds)
 bos shutdown: Stops processes without changing their status/ bos shutdown(8dfs)
 /status flags to Run in memory without changing their status/ bos startup(8dfs)
 /a DCE LFS fileset from its site without updating the FLDB fts zap(8dfs)
 /data the Cache Manager cannot write back to a File Server/ cm lsstores(8dfs)
 /attempts by the Cache Manager write data back to a File Server/ cm resetstores(8dfs)
 to from its site without/ fts zap: Removes a DCE LFS fileset fts zap(8dfs)

Chapter 1

Remote Procedure Call Commands

intro

Purpose Introduction to DCE RPC daemon and RPC control program commands

Description

DCE RPC provides two administrative facilities, the RPC daemon and the RPC control program:

RPC daemon

Provides the **Endpoint Map Service**, which maintains the local endpoint map for local RPC servers and looks up endpoints for RPC clients. An **endpoint** is the address of a specific instance of a server executing in a particular address space on a given system (a **server instance**). Each endpoint can be used on a system by only one server at a time.

endpoint map

A database where servers register their binding information, including endpoints, for each of their RPC interfaces and the associated RPC objects. Each combination of binding information, interface identifier, and object UUID uses a distinct element in the local endpoint map.

The **rpcd** command starts the RPC daemon.

RPC control program

A set of commands for accessing the operations of the RPC **name service interface (NSI)**. For managing endpoint maps, the control program supports showing endpoint map elements and removing any set of map elements from the local endpoint map or from any remote endpoint map.

The **rpccp** command starts the RPC control program (RPCCP).

Exit Values

The RPC control program reports DCE error messages on the command line. If the command executes successfully, the internal value returned is 0 (zero); otherwise, the value is -1 (negative one).

Related Information

Commands: **rpcd(8rpc)**, **rpccp(8rpc)**

Books: *OSF DCE Administration Guide*, *OSF DCE Application Development Guide*, *OSF DCE Application Development Reference*

rpccp

Purpose Starts the RPC control program

Synopsis `rpccp [rpccp-command]`

Arguments

rpccp-command

Specifies one of the following control program commands:

add element Adds an element to a profile in a name service entry; if the specified entry does not exist, creates the entry.

add entry Adds an entry to the name service database.

add member Adds a member to a group in a name service entry; if the specified entry does not exist, creates the entry.

exit Leaves the RPC control program.

export Exports binding information for an interface identifier, object UUIDs, or both to a server entry; if the specified entry does not exist, creates the entry.

help Displays a list of commands or the possible options of a specified command.

import Imports binding information and an object UUID from a server entry.

quit Leaves the RPC control program.

remove element
Removes selected elements from a profile.

remove entry
Removes an entry from the name service database.

remove group
Removes all group members and the group from the specified entry.

remove mapping

Removes specified elements from the local endpoint map or from the endpoint map of a specified remote host.

remove member

Removes a selected member from a group.

remove profile

Removes all profile elements and the profile from the specified entry.

show entry Shows the NSI attributes of an entry.

show group Shows the members of a group.

show mapping

Shows the elements of the local endpoint map.

show profile Shows the elements of a profile.

show server Shows the binding information, interface identifier, and object UUIDs in a server entry.

unexport Removes binding information, interface identifiers, and object UUIDs from a server entry.

Notes

A server entry equates to an NSI binding attribute and, optionally, an object attribute; a group equates to an NSI group attribute; and a profile equates to an NSI profile attribute. Typically, each server's entries, groups, and profiles reside in distinct name service entries.

Description

The RPC control program (RPCCP) provides a set of commands for managing name service use for RPC applications and for managing the endpoint map.

You can use control program commands from within the control program or from the system prompt (represented here as a \$).

To use the control program commands from inside the control program, Start and enter the control program using the **rpccp** command alone, without any argument. The control program then displays the control program prompt (**rpccp>**), as follows:

```
$ rpccp
rpccp>
```


rpccp(8rpc)

You can then enter any control program command, for example:

```
rpccp> show entry /LandS/anthro/pr_server_node3
```

You leave the control program and return to the system prompt using the **exit** or **quit** command.

If you enter invalid input, the control program displays the valid commands.

To use the control program commands from the system prompt, enter the **rpccp** command with an internal command of the control program as the first argument. You can do this either interactively or in a command procedure. For example, you can enter the **show entry** command as follows:

```
$ rpccp show entry /LandS/anthro/pr_server_node3
```

Arguments and Options

Except for the **exit** and **quit** commands, **rpccp** commands have one or more options. Each option is identified by a - (dash) followed by a letter; for example, **-s**. Some options require arguments.

Commands that access NSI operations also require the name of a name service entry as an argument. The order of arguments and the entry-name option is arbitrary; for example, the following placements of arguments and options are equivalent:

```
rpccp> add entry /LandS/anthro/mis_node_2 \  
> -i EC1EEB60-5943-11C9-A309-08002B102989,1.0
```

```
rpccp> add entry -i EC1EEB60-5943-11C9-A309-08002B102989,1.0 \  
> /LandS/anthro/mis_node_2
```

Environmental Influences on Command Syntax

There are variations in the action of the control program, depending on whether commands are entered from the system prompt or from within the control program. For example, entering the annotation field of profile elements from the system prompt allows you to include internal spaces in an annotation.

Function	At System Prompt	Inside Control Program
Strings within quotation marks	Supported ¹	Not required
Wildcard substitution	Supported	Unsupported

¹ Some UNIX systems require that you place an escape symbol (\) before string binding delimiters such as brackets ([]) or that you place the delimiters within quotation marks (' ' or " ") at the system prompt.

The following table describes the scope of the RPC control program commands.

Scope	Command
All entries	add entry remove entry show entry
Server entry	export import show server unexport
Group	add member remove group remove member show group
Profile	add element remove element remove profile show profile
Endpoint map	remove mapping show mapping

Environment Variables

The control program supports environment variables. Using environment variables facilitates interactive use of the control program.

To distinguish environment variables, **rpccp*(8rpc)** reference pages follow the convention of using all uppercase letters for examples of environment variables. Note that UNIX environment variables are case sensitive.

User-defined environment variables

You can set an environment variable to represent values to **rpccp**. Using an environment variable is helpful for specifying a long string such as the following:

- A string representation of binding information (binding string)
- A string representation of an object or interface UUID (string UUID)

rpccp(8rpc)

- An interface identifier (the interface UUID and version numbers)
- The name of a name service entry

For example, in the following C-Shell example, the environment variable **JANE_CAL** represents an object UUID; the target name service entry, **/Lands/anthro/Cal_host_2**, is in the local cell:

```
$ setenv JANE_CAL 47F40D10-E2E0-11C9-BB29-08002B0F4528
```

```
$ rpccp
```

```
rpccp> export -o JANE_CAL /Lands/anthro/Cal_host_2
```

DCE RPC environment variables

NLSPATH The environment variable **NLSPATH** must point to the location of **dcerpc.cat** and **dcedcs.cat**. Otherwise, any runtime status codes returned by the control program will be hexadecimal, rather than textual. form. The value of this variable must include both the pathname of the directory where the **.cat** files reside and the string **%N**.

RPC_DEFAULT_ENTRY_SYNTAX

The **dce** name syntax is the only syntax currently supported by the DCE Cell Directory Service (CDS). However, the Name Service Interface (NSI) is independent of any specific name service and, in the future, may support name services that use other name syntaxes. When alternative name syntaxes are supported, you can override the standard default with a process-specific default by setting the **RPC_DEFAULT_ENTRY_SYNTAX** environment variable. When this variable is set for a process, the control program uses it to find out the default syntax for the process. You can override this default in any NSI command of the control program by using the **-s** option to specify an alternative entry syntax. Setting **RPC_DEFAULT_ENTRY_SYNTAX** requires specifying the integer 3 to indicate the **dce** syntax. To set **RPC_DEFAULT_ENTRY_SYNTAX** in the C shell, use the **setenv** command. The following command specifies **dce** as the default name syntax in a login command file:

```
# .login command file
# setting dce as default name syntax,
setenv RPC_DEFAULT_ENTRY_SYNTAX 3
```

For the Bourne or Korn Shell, the equivalent command is

RPC_DEFAULT_ENTRY_SYNTAX=3

RPC_DEFAULT_ENTRY

For the **import** command, you can use this environment variable to indicate the entry where the search operation starts. Usually, the starting entry is a profile.

The Name Service Interface

The remainder of this description contains information to help you use commands that call the name service interface to access name service entries (NSI commands).

The DCE RPC name service interface (NSI) is independent of any particular name service. CDS, however, is the only name service available for DCE RPC Version 1.0 applications. For more details on the name service interface, see the *OSF DCE Application Development Guide*. For a description of the DCE Cell Directory Service, see the *OSF DCE Administration Guide*.

Name Service Entries

To store information about RPC servers, interfaces, and objects, the NSI defines the following name service entries:

- server entry** Stores binding information, interface identifiers, and object UUIDs for an RPC server
- group** Corresponds to one or more RPC servers that offer a common RPC interface, type of RPC object, or both
- profile** Defines search paths for looking in a name service database for a server that offers a particular RPC interface and object

Note that when the NSI is used with the Cell Directory Service, the name service entries are CDS object entries

Structure of Entry Names

Each entry in a name service database is identified by a unique global name made up of a cell name and a cell-relative name.

A cell is a group of users, systems, and resources that share common DCE services. A cell configuration includes at least one cell directory server, one security server, and one time server. A cell's size can range from one system to thousands of systems. For information on cells, see the CDS reference pages.

The following is an example of a global name:

/.../C=US/O=uw/OU=MadCity/LandS/anthro/Stats_host_2

The parts of a global name are as follows:

Cell name (using X.500 name syntax)

For example:

```
/. . . /C=US/O=uw/OU=MadCity
```

The symbol */. . .* begins a cell name. The letters before the equal signs (=) are abbreviations for country (C), organization (O), and organization unit (OU).

For entries in the local cell, the cell name can be represented by a */.:* prefix, in place of the actual cell name; for example,

```
./:/LandS/anthro/Stats_host_2
```

For NSI operations on entries in the local cell you can omit the cell name.

Cell-relative name

Each name service entry requires a cell-relative name, which contains a directory pathname and a leaf name.

directory pathname

Follows the cell name and indicates the hierarchical relationship of the entry to the cell root.

The directory pathname is the middle portion of the global name. The cell name is to the left of the directory pathname, and the leaf name is to the right, as follows:

cell-name + directory-pathname + leaf-name

The directory pathname contains the names of any subdirectories in the path; each subdirectory name begins with a slash (/), as follows:

/sub-dir-a-name/sub-dir-b-name/sub-dir-c-name

Directory paths are created by name service administrators. If an appropriate directory path does not exist, ask your name service administrator to extend an existing path or create a new path. In a directory path, the name of a subdirectory should reflect its relationship to its parent directory (the directory that contains the subdirectory).

leaf name Identifies the specific entry. The leaf name is the right-hand part of global name beginning with the rightmost slash.

In the following example, `/.../C=US/O=uw/OU=MadCity` is the cell name, `/LandS/anthro` is the directory pathname, and `/Cal_host_4` is the leaf name.

`/.../C=US/O=uw/OU=MadCity/LandS/anthro/Cal_host_4`

If a name service entry is located at the cell root, the leaf name directly follows the cell name; for example, `./:cell_profile`.

Note that when the NSI is used with CDS, the cell-relative name is a CDS name.

Guidelines for Constructing Names of Name Service Entries

A global name includes both a cell name and a cell-relative name composed of a directory pathname and a leaf name. The cell name is assigned to a cell root at its creation. When you specify only a cell-relative name to an NSI command, the NSI automatically expands the name into a global name by inserting the local cell name. When returning the name of a name service entry, a group member, or member in a profile element, NSI operations return global names.

The directory pathname and leaf name uniquely identify a name service entry. The leaf name should somehow describe the entry; for example, by identifying its owner or its contents. The remainder of this section contains guidelines for choosing leaf names. Note that directory pathnames and leaf names are case sensitive.

Naming a Server Entry

For a server entry that advertises an RPC interface or service offered by a server, the leaf name must distinguish the entry from the equivalent entries of other servers. When a single server instance runs on a host, you can ensure a unique name by combining the name of the service, interface (from the interface definition), or the system name for the server's host system.

For example, consider two servers, one offering a calendar service on host JULES and one, on host VERNE.

The server on JULES uses the following leaf name:

calendar_JULES

The server on VERNE uses the following leaf name:

calendar_VERNE

For servers that perform tasks on or for a specific system, an alternative approach is to create server entries in a system-specific host directory within the name service database. Each host directory takes the name of the host to which it corresponds. Because the directory name identifies the system, the leaf name of the server entry name need not include the host name, for example:

./:/LandS/host_1/Process_control

To construct names for the server entries used by distinctive server instances on a single host, you can construct unique server entry names by combining the following information: the name of the server's service, interface, or object; the system name of the server's host system, and a reusable instance identifier, such as an integer.

For example, the following leaf names distinguish two instances of a calendar service on the JULES system:

calendar_JULES_01

calendar_JULES_02

Avoid automatically generating entry names for the server entries of server instances, for example, by using unique data such as a time stamp (**calendar_verne_15OCT91_21:25:32**) or a process identifier (**calendar_jules_208004D6**). When a server incorporates such unique data into its server entry names, each server instance creates a separate server entry, causing many server entries. When a server instance stops running, it leaves an obsolete server entry that is not reused. The creation of a new entry whenever a server instance starts may impair performance.

A server can use multiple server entries to advertise different combinations of interfaces and objects. For example, a server can create a separate server entry for a specific object (and the associated interfaces). The name of such a server entry should correspond to a well-known name for the object. For example, consider a server that offers a horticulture bulletin board known to users as **horticulture_bb**. The server exports the **horticulture_bb** object, binding information, and the associated bulletin-board interface to a server entry whose leaf name identifies the object, as follows:

horticulture_bb

Note that an RPC server that uses RPC authentication can choose identical names for its principal name and its server entry. Use of

identical names permits a client that calls the **rpc_binding_set_auth_info** routine to automatically determine a server's principal name (the client will assume the principal name to be the same as the server's entry name). If a server uses different principal and server entry names, users must explicitly supply the principal name. For an explanation of principal names, see the DCE Security Service part of the *OSF DCE Application Development Guide*.

Naming a Group

The leaf name of a group should indicate the interface, service, or object that determines membership in the group. For example, for a group whose members are selected because they advertise an interface named **Statistics**, the following is an effective leaf name:

Statistics

For a group whose members advertise laser-printer print queues as objects, the following is an effective leaf name:

laser-printer

Naming a Profile

The leaf name of a profile should indicate the profile users; for example, for a profile that serves the members of an accounting department, the following is an effective leaf name:

accounting_profile

Privilege Required

To use the NSI commands to access entries in a CDS database, you need access control list (ACL) permissions. Depending on the NSI operation, you need ACL permissions to the parent directory or the CDS object entry (the name service entry) or both. The ACL permissions are as follows:

- To create an entry, you need insert permission to the parent directory.
- To read an entry, you need read permission to the CDS object entry.
- To write to an entry, you need write permission to the CDS object entry.
- To delete an entry, you need delete permission either to the CDS object entry or to the parent directory.

Note that write permission does not imply read permission.

ACL permissions for the NSI commands of the control program are described in the reference pages.

rpccp(8rpc)

Examples The following command starts the RPC control program:

```
$ rpccp
rpccp>
```

The following command at the system prompt removes the entry **/LandS/anthro/Cal_host_2**:

```
$ rpccp remove entry \  
> /LandS/anthro/Cal_host_2
```

Related Information

Commands: **add element(8rpc)**, **add entry(8rpc)**, **add member(8rpc)**, **export(8rpc)**, **import(8rpc)**, **remove element(8rpc)**, **remove entry(8rpc)**, **remove group(8rpc)**, **remove mapping(8rpc)**, **remove member(8rpc)**, **remove profile(8rpc)**, **show entry(8rpc)**, **show group(8rpc)**, **show mapping(8rpc)**, **show profile(8rpc)**, **show server(8rpc)**, **unexport(8rpc)**

add element

Purpose Adds an element to a profile in a name service entry; if the specified entry does not exist, creates the entry

Synopsis `rpcpc add element profile-entry-name -m member {-d | -i if-id [-p priority]} [-a annotation] [-s syntax]`

Options

- m** Defines a member name for the profile element to be added (required).
- d** Performs the **add element** operation on the default profile element. With the **-d** option, the **-i** and **-p** options are ignored.
- i** Defines an interface identifier for the profile element to be added. Only one interface can be added in a single operation. An interface identifier is required, unless the default profile element is being added. With the **-d** option, the **-i** option is ignored.
The value has the following form:
interface-uuid.major-version.minor-version
The UUID is a hexadecimal string and the version numbers are a decimal string, for example:
`-i EC1EEB60-5943-11C9-A309-08002B102989,3.11`
Leading zeros in version numbers are ignored.
- p** Defines a search priority for the new profile element. The priority value is in the range 0 to 7, with zero having the highest priority. When a default element is added (with the **-d** option), the **-p** option is ignored. By default, a nondefault element is assigned a priority value of zero.
- a** Defines an annotation string for the profile element.
Note that the shell supports quotation marks around the annotation field of profile elements, which allows you to include internal spaces in an annotation; the control program does not. To specify or refer to annotations from within the control program, limit each annotation to an unbroken alphanumeric string; for example, **CalendarGroup**. To refer to annotations from the system prompt, do not incorporate quotation marks into any annotation.

add element(8rpc)

- s Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

Arguments

profile-entry-name

Specifies the entry name of the target profile. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **add element** command adds an element to a profile in a name service entry. The name of the entry containing the profile and the entry name of the profile member in the new element are required. The entry of a profile may have been created previously (by either the **add entry** or **add element** command). But, if the specified entry does not exist, the **add element** command tries to create the entry.

A profile element is a database record containing the following fields:

Interface identifier

This is the primary search key. The interface identifier consists of the interface UUID and the interface version numbers.

Member name

The entry name of one of the following kinds of name service entries:

- A server entry for a server offering the requested RPC interface and object
- A group corresponding to the requested RPC interface
- A profile

Priority value

The priority value (0 (zero) is the highest priority; 7 is the lowest) is designated by the creator of a profile element to help determine the order for using the element. NSI search operations select among like priority elements at random. For the **rpccp add element** command, the default is 0.

Annotation string

The annotation string enables you to identify the purpose of the

profile element. The annotation can be any textual information, for example, an interface name associated with the interface identifier or a description of a service or resource associated with a group. The annotation string is not a search key for the import or lookup operations.

Privilege Required

You need both read permission and write permission to the CDS object entry (the target profile entry). If the entry does not exist, you also need insert permission to the parent directory.

Examples The following command adds an element to the cell profile, `/cell_profile`, in the local cell:

```
$ rpccp
rpccp> add element \
> -i EC1EEB60-5943-11C9-A309-08002B102989,1.1 \
> -m /Calendar_profile \
> -a RefersToCalendarGroups \
> /cell_profile
```

The following control program commands start the control program, set up a user profile associated with the cell profile as its default element, and add a user-specific element for the Calendar V1.1 interface, as follows:

```
$ rpccp
rpccp> add element /LandS/anthro/molly_o_profile \
> -d -m /cell_profile
rpccp>
rpccp> add element /LandS/anthro/molly_o_profile \
> -m /LandS/anthro/Calendar_group \
> -i EC1EEB60-5943-11C9-A309-08002B102989,1.1 \
> -a Calendar_Version 1.1_Interface
rpccp>
```

The added profile element contains the global name of the member (specified using its cell-relative name, `/LandS/anthro/Calendar_group`) and the RPC interface identifier for the Calendar Version 1.1 interface.

Related Information

Commands: `remove element(8rpc)`, `remove profile(8rpc)`, `show profile(8rpc)`

add entry(8rpc)

add entry

Purpose Adds a name service entry to the name service database

Synopsis `rpcpp add entry entry-name [-s syntax]`

Options `-s` Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the `-s` option is unnecessary.

Arguments

entry-name Specifies the name of the target name service entry. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **add entry** command adds an unspecialized entry to the name service database. The name of the entry is required.

The new entry initially contains no NSI attributes. This command creates a general name service entry for an application or user. The application or user can later use the **export**, **add element**, and **add member** commands to make the generic entry into a server entry, a group, or a profile (or a combination), as follows:

- For a server entry, specify the new entry as the target entry for the **rpcpp export** command.
- For a group, specify the new entry as the target group for the **rpcpp add member** command.
- For a profile, specify the new entry as the target profile for the **rpcpp add element** command.

The **add entry** command enables administrators to add entries for users who lack the required permissions. If you have the permissions required by the **add entry**

command, you can also add an entry using an **export**, **add member**, or **add element** command; if the entry you specify does not exist, the command creates the entry.

Privilege Required

To add an entry, you need insert permission to the parent directory and both read permission and write permission to the CDS object entry (the target name service entry).

Examples The following commands start RPCCP and add an unspecialized entry to the name service database:

```
$ rpccp
rpccp> add entry \
> /LandS/anthro/Cal_host_2
```

The following command operates from the system prompt to add an unspecialized entry to the name service database:

```
$ rpccp add entry \
> /LandS/anthro/Cal_host_3
```

Related Information

Commands: **remove entry(8rpc)**, **show entry(8rpc)**

add member(8rpc)

add member

Purpose Adds a member to a group in a name service entry; if the specified entry does not exist, creates the entry

Synopsis `rpcpp add member group-entry-name -m member [-s syntax]`

Options

- m** Declares the name of a member to be added to the specified group entry (required).
You can add only one member at a time.
- s** Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

Arguments

group-entry-name

Specifies the name of the target group. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **add member** command adds a member to a group in a name service entry. The name of the entry containing the group and the name of the new group member are required. The entry of a group may have been created previously (by either the **add entry** or **add member** command). If the specified entry does not exist, the **add member** command tries to create the entry.

Privilege Required

You need both read permission and write permission to the CDS object entry (the target group entry). If the entry does not exist, you also need insert permission to the parent directory.

add member(8rpc)

Examples The following commands run RPCCP and add the member `/LandS/anthro/Cal_host_3` to the group `/LandS/anthro/Calendar_group`:

```
$ rpccp
rpccp> add member \
> -m /LandS/anthro/Cal_host_3 \
> /LandS/anthro/Calendar_group
```

Related Information

Commands: `remove group(8rpc)`, `remove member(8rpc)`, `show group(8rpc)`

export(8rpc)**export**

Purpose Exports binding information for an interface identifier or object UUIDs or both to a server entry; if the specified entry does not exist, creates the entry

Synopsis **rpccp export** *entry-name*
 { **-i** *if-id* **-b** *string-binding* [**-b** *string-binding...*] **-o** *object-uuid* [**-o** *object-uuid...*] |
-i *if-id* **-b** *string-binding* [**-b** *string-binding...*] | **-o** *object-uuid* [**-o** *object-uuid...*] }
 [**-s** *syntax*]

Options **-i** Declares the interface identifier of an RPC interface. The **export** command operates on only one **-i** option; if you enter more than one, the command ignores all but the last interface identifier. If you specify an interface identifier, you must specify at least one **-b** option. The **-i** and **-o** options can occur together or separately, but one of them is necessary.

The interface identifier takes the following form:

interface-uuid,major-version.minor-version

The version numbers are optional, but if you omit a version number, the value defaults to 0. The UUID is a hexadecimal string and the version numbers are decimal strings, for example:

-i EC1EEB60-5943-11C9-A309-08002B102989,3.11

Leading zeros in version numbers are ignored.

-b Declares a string binding (optional). To use this option, you must also specify an interface identifier (using the **-i** option). Each command accepts up to 32 **-b** options.

The value has the form of an RPC string binding, without an object UUID. The binding information contains an RPC protocol sequence, a network address, and sometimes an endpoint within brackets (*rpc-prot-seq:network-addr[endpoint]*). For a well-known endpoint, include the endpoint in the string binding, for example:

-b ncadg_ip_udp:63.0.2.17[5347]

For a dynamic endpoint, omit the endpoint from the string binding, for example:

-b ncacln_ip_tcp:16.20.15.25

Note that depending on your system, string binding delimiters such as brackets ([]) may need to be preceded by an escape symbol (\) or placed within quotation marks (' ' or " "). Requirements vary from system to system, and you must conform to the usage rules of a system.

- o Declares the UUID of an object. Each **export** command accepts up to 32 **-o** options. The **-i** and **-o** options can occur together or separately, but one of them is necessary.

The UUID is a hexadecimal string, for example:

-o 3C6B8F60-5945-11C9-A236-08002B102989

- s Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

Arguments

- entry-name* Specifies the name of the target name service entry. Usually, the target is a server entry. However, objects also can be exported (without an interface identifier or any binding information) to a group or a profile.

For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **export** command places binding information and an interface identifier, object UUIDs, or both into a server entry, or the command object UUIDs into a group's entry. The **export** command searches the name service database for the entry with the specified entry name. If the entry exists, the command uses it; otherwise, the command tries to create a new name service entry using the specified entry name.

Minimally, the command requires the name of the entry and either an identifier and binding string or an object.

If the specified entry does not exist, the **export** command tries to create the entry.

export(8rpc)**Privilege Required**

You need both read permission and write permission to the CDS object entry (the target name service entry). If the entry does not exist, you also need insert permission to the parent directory.

Examples This example shows a control program **export** command that is stored in a file for later execution from the system prompt. The command exports two objects and an interface with two string bindings to the server entry **/LandS/anthro/Cal_host_3** in the local cell:

```
# file to export Calendar 1.1 at installation time
rpccp export \
-i EC1EEB60-5943-11C9-A309-08002B102989,1.1 \
-b ncacn_ip_tcp:16.20.15.25 \
-b ncadg_ip_udp:63.0.2.17 \
-o 30DBEEA0-FB6C-11C9-8EEA-08002B0F4528 \
-o 16977538-E257-11C9-8DC0-08002B0F4528 \
/LandS/anthro/Cal_host_3
```

The following example shows the use of a user-defined environment variable as an interface identifier, to facilitate entering an export command interactively (in this case, from inside the control program).

The C shell **setenv** command sets up an environment variable **Calendar_1_1**, which represents the interface identifier of an RPC interface. The **rpccp** command then starts the control program, and the **export** command exports the Calendar interface and two string bindings to the server entry **/LandS/anthro/Cal_host_2** in the local cell, as follows:

```
$ setenv Calendar_1_1 EC1EEB60-5943-11C9-A309-08002B102989,1.1
$ rpccp
rpccp> export -i Calendar_1_1 \
> -b ncacn_ip_tcp:16.20.15.25 \
> -b ncadg_ip_udp:63.0.2.17 \
> /LandS/anthro/Cal_host_2
```

The following example shows the use of user-defined environment variables for object UUIDs to facilitate entering an export command interactively (in this case, from inside the control program).

The C shell **setenv** command sets up the environment variables **LUKE_CAL** and **JOSH_CAL**, which represent personal calendars that are accessible as objects to an RPC server. The **rpccp** command then starts the control program, and the

export(8rpc)

export command exports the two objects to the server's entry **/LandS/anthro/Cal_host_2** in the local cell:

```
$ setenv LUKE_CAL 30DBEEA0-FB6C-11C9-8EEA-08002B0F4528
$ setenv JOSH_CAL 16977538-E257-11C9-8DC0-08002B0F4528
$ rpccp
rpccp> export -o LUKE_CAL -o JOSH_CAL \
> /LandS/anthro/Cal_host_2
```

Related Information

Commands: **import(8rpc)**, **show server(8rpc)**, **unexport(8rpc)**

help(8rpc)

help

Purpose Displays a list of commands or the options of a specified command

Synopsis `rpccp help [rpccp-command]`

Arguments

rpccp-command

Specifies one of the following control commands:

- **add element**
- **add entry**
- **add member**
- **exit**
- **export**
- **import**
- **quit**
- **remove element**
- **remove entry**
- **remove group**
- **remove mapping**
- **remove member**
- **remove profile**
- **show entry**
- **show group**
- **show mapping**

- **show profile**
- **show server**
- **unexport**

Description

The **help** command displays information about the RPCCP command set or the options and argument associated with a specific command.

Examples The following command operates from the system prompt to display the internal commands of the control program:

```
$ rpccp help
```

The following commands start the control program and display the syntax of the **remove entry** command:

```
$ rpccp  
rpccp> help remove entry
```

Related Information

Commands: **add element(8rpc)**, **add entry(8rpc)**, **add member(8rpc)**, **export(8rpc)**, **import(8rpc)**, **remove element(8rpc)**, **remove entry(8rpc)**, **remove group(8rpc)**, **remove mapping(8rpc)**, **remove member(8rpc)**, **remove profile(8rpc)**, **rpccp(8rpc)**, **show entry(8rpc)**, **show group(8rpc)**, **show mapping(8rpc)**, **show profile(8rpc)**, **show server(8rpc)**, **unexport(8rpc)**

import(8rpc)**import**

Purpose Imports binding information and an object UUID from a server entry

Synopsis `rpcpp import starting-entry-name -i if-id [-v versions] [-e] [-n [integer]] [-o object-uuid] [-s syntax] [-u]`

Options **-i** Defines an interface identifier to be imported (required). You can import only one interface at a time.

The value has the following form:

interface-uuid,major-version.minor-version

The UUID is a hexadecimal string and the version numbers are decimal strings, for example:

-i EC1EEB60-5943-11C9-A309-08002B102989,1.1

Leading zeros in version numbers are ignored.

-v Indicates how a specified interface version is used (optional). If it is used without the **-i** option, the **-v** option is ignored. The possible combinations of versions for the **-v** option and their actions are described in the following table.

Versions	Action
all	The interface version is ignored.
exact	Both the major and minor versions must match the specified versions.
compatible	The major version must match the specified version, and the minor version must be greater than or equal to the specified version.
major_only	The major version must match the specified version; the minor version is ignored.
upto	The major and minor versions must be less than or equal to those specified.

If the **-v** option is absent, the command shows compatible version numbers.

-e Shows the name of the entry where the binding is found (optional).

- n** Declares that the import operation is to continue until no more potential bindings are found (optional). Providing a numeric value to this option restricts the number of imported bindings. If you omit the number, only one binding is imported. If repeated, this operation may return the same binding.

For example, **-n** imports all available bindings, and **-n 5** imports up to five bindings. Note that the imported bindings are displayed as string bindings.
- o** Declares the UUID of an object to be imported (optional). Only one UUID can occur in a single operation.

If an object is specified, the import operation limits its search to server entries that contain both the specified interface identifier and object UUID when searching for a potential binding. Without the **-o** option, the import operation ignores object UUIDs.

The UUID is a hexadecimal string, for example:

-o 3C6B8F60-5945-11C9-A236-08002B102989
- s** Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.
- u** Updates the local copy of name service data (optional). Name service data that is requested by applications is sometimes stored locally. If a local copy of name service data satisfies an NSI command, the RPC control program uses the local copy. Local copies of name service data are not automatically updated. Specify the **-u** option to display the current contents of an entry has changed recently.

Arguments

starting-entry-name

Indicates the name of the server entry where the import operation starts. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

import(8rpc)

Description

The **import** command imports binding information and an RPC object UUID for a specific RPC interface from a server entry. The name of the entry and the interface identifier are required. The entry name can refer to a server entry, a group, or a profile.

Privilege Required

You need read permission to the specified CDS object entry (the starting name service entry) and to any CDS object entry in the resulting search path.

Examples The following commands run RPCCP and import an interface and object:

```
$ rpccp
rpccp> import -i EC1EEB60-5943-11C9-A309-08002B102989,1.1\
> -o 30DBEEA0-FB6C-11C9-8EEA-08002B0F4528 \
> /LandS/anthro/Cal_host_3
```

Related Information

Commands: **export(8rpc)**, **show server(8rpc)**, **unexport(8rpc)**

remove element

Purpose Removes selected elements from a profile

Synopsis `rpcpp remove element profile-entry-name {-d | -i if-id -m member | -a annotation} [-s syntax]`

Options

- d** Removes the default profile element. With the **-d** option, the **-a**, **-i**, and **-m** options are ignored.
- i** Defines an interface identifier for the profile element to be removed for a member specified with the **-m** option. Only one interface and member pair can be removed in a single operation. If you supply multiple instances of the **-i** option, the command uses the final instance.

The **-i** and **-m** options take precedence over the **-a** option. However, if the default profile element is specified (by the **-d** option), the **-i** and **-m** options are ignored.

The interface identifier value has the following form:
interface-uuid,major-version.minor-version

The UUID is a hexadecimal string and the version numbers are decimal strings, for example:
-i EC1EEB60-5943-11C9-A309-08002B102989,1.1

Leading zeros in version numbers are ignored.
- m** Defines a member name for the profile element to be removed. This option is required if the interface identifier is specified. Only one interface and member can be removed in a single operation. If you supply multiple instances of the **-m** option, the command uses the final instance.
- a** Removes all elements whose annotation fields match the specified annotation; in the presence of **-d** option or **-i** and **-m** options, the **-a** option is ignored.

Note that the shell supports quotation marks around the annotation field of profile elements, which allows you to include internal spaces in an annotation; the control program does not. To specify or refer to annotations from within the control program, limit each annotation to an unbroken alphanumeric string; for example,

remove element(8rpc)

CalendarGroup. To refer to annotations from the system prompt, do not incorporate quotation marks into any annotation.

- s Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

Arguments

profile-entry-name

Indicates the name of the target profile. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **remove element** command removes an element from a profile in the name service database. For a description of the fields in a profile element, see **add entry(8rpc)**.

The **remove element** command requires the entry name of the profile. The command also requires one of the following options:

- d The default profile option takes precedence over the other two options.
- i *interface-id* -m *member-name* An interface and member pair takes precedence over the **-a** option.
- a *annotation-string*
The annotation option takes effect only if neither the **-d** or **-i** option is specified.

Privilege Required

You need read permission and write permission to the CDS object entry (the target profile entry).

Examples The C-Shell **setenv** command sets up an environment variable **Calendar_1_1**, which represents the interface identifier of an RPC interface. The control program commands set up an environment variable for the interface identifier of the Calendar Version 1.1 RPC interface, run RPCCP, and remove an element from a profile, as follows:

```
$ setenv Calendar_1_1 EC1EEB60-5943-11C9-A309-08002B102989,1.1
$
$ rpccp
rpccp> remove element -i Calendar_1_1 \
> -m /LandS/anthro/Calendar_group \
> /LandS/anthro/molly_o_profile
```

Related Information

Commands: **add element(8rpc)**, **remove profile(8rpc)**, **show profile(8rpc)**

remove entry(8rpc)

remove entry

Purpose Removes a name service entry from the name service database

Synopsis `rpccp remove entry entry-name [-s syntax]`

Options `-s` Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the `-s` option is unnecessary.

Arguments

entry-name Indicates the name of the target name service entry. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **remove entry** command removes an entry from the name service database. The name of the entry is required.

Privilege Required

You need read permission to the CDS object entry (the target name service entry). You also need delete permission to the CDS object entry or to the parent directory.

Examples The following commands run RPCCP and remove the entry `/LandS/anthro/Cal_host_2` from the local cell of the name service database:

```
$ rpccp
rpccp> remove entry /LandS/anthro/Cal_host_2
```

Related Information

Commands: **add entry(8rpc)**, **show entry(8rpc)**

remove group

Purpose Removes all group members and the group from the specified name service entry

Synopsis `rpcpp remove group group-entry-name [-s syntax]`

Options **-s** Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

Arguments

group-entry-name

Indicates the name of the target group. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **remove group** command removes a group from the name service database. The group need not be empty. The entry name of the group is required.

Privilege Required

You need write permission to the CDS object entry (the target group entry).

Examples The following commands run RPCCP and remove the group from the name service entry `/LandS/anthro/Calendar_group`:

```
$ rpcpp  
rpcpp> remove group /LandS/anthro/Calendar_group
```

Related Information

Commands: **add member(8rpc)**, **remove member(8rpc)**, **show group(8rpc)**

remove mapping(8rpc)

remove mapping

Purpose Removes specified elements from either the local or a remote endpoint map

Synopsis `rpcpp remove mapping [host-address] -b string-binding -i if-id [-v versions] [-o object-uuid]`

Options **-b** Declares a string binding (required). You must also specify an interface identifier (using the **-i** option). Each command accepts up to 32 **-b** options.

The value has the form of an RPC string binding, without an object UUID, for example:

-b ncadg_ip_udp:63.0.2.17[5347]

Note that depending on your system, string binding delimiters such as brackets ([]) may need to be preceded by an escape symbol (\) or placed within quotation marks (' ' or " "). Requirements vary from system to system, and you must conform to the usage rules of a system.

-i Declares an interface identifier (required). Only one interface can be removed in a single operation. The interface identifier has the following form:

interface-uuid,major-version.minor-version

The UUID is a hexadecimal string and the version numbers are decimal strings, for example:

-i EC1EEB60-5943-11C9-A309-08002B102989,1.1

Leading zeros in version numbers are ignored.

-v Indicates how a specified interface version is used (optional). If it is used without the **-i** option, the **-v** option is ignored. The possible combinations of versions for the **-v** option and their actions are as follows:

remove mapping(8rpc)

Versions	Action
all	The interface version is ignored.
exact	Both the major and minor versions must match the specified versions.
compatible	The major version must match the specified version, and the minor version must be greater than or equal to the specified version.
major_only	The major version must match the specified version; the minor version is ignored.
upto	The major and minor versions must be less than or equal to those specified.

If the **-v** option is absent, the command shows compatible version numbers.

- o** Defines an object UUID that further determines the endpoint map elements that are removed (optional). Each **remove mapping** command accepts up to 32 **-o** options.

The UUID is a hexadecimal string, for example:

-o 3C6B8F60-5945-11C9-A236-08002B102989

Arguments

host-address The **host-address** argument is a string binding that indicates where to find the target endpoint map. When accessing the local endpoint map, you can specify what protocol sequence to use (optional); for example,

ncadg_ip_udp:

When accessing a remote endpoint map, you must specify both a protocol sequence and a network address for the remote system (required); for example,

ncadg_ip_udp:16.20.16.44

An endpoint is unnecessary in local or remote host addresses, and the **remove mapping** command ignores any endpoint specified as part of a host address.

remove mapping(8rpc)**Description**

The **remove mapping** command removes one or more elements from an endpoint map. The target endpoint map can be either the local endpoint map or the endpoint map of a specified remote host. Each map element corresponds to an object UUID, interface identifier, annotation (optional), and binding information. The binding information contains an RPC protocol sequence, a network address, and an endpoint within brackets (*rpc-prot-seq:network-addr[endpoint]*).

If entered without a remote host address as an argument, the command operates on the local endpoint map. This command requires one interface identifier (the **-i** option); at least one string binding (the **-b** option); and optionally, one or more object UUIDs (the **-o** option). Each instance of the command accepts from 1 to 32 **-b** options and from 0 to 32 **-o** options. The options work together to delimit the elements to be removed from the target endpoint map. The command removes any map element that contains the specified interface identifier, a specified string binding, and a specified object UUID (if any).

Examples The following command operates from the system prompt to remove a map element from the local endpoint map. The command removes only the map element that contains the specified interface identifier, server address (specified as a string binding), and object UUID.

```
$ rpccp remove mapping \  
> -b ncdg_ip_udp:16.20.16.64[3424] \  
> -i EC1EEB60-5943-11C9-A309-08002B102989,1.1 \  
> -o 30DBEEA0-FB6C-11C9-8EEA-08002B0F4528  
$
```

The following commands start the control program and remove an element from a remote endpoint map. The **remove mapping** command operates on the endpoint map of the remote host specified by the host address (**ncdg_ip_udp:16.20.16.44**) and removes the map element that contains the specified interface identifier, server address (specified as a string binding), and object UUID:

```
$ rpccp  
rpccp> remove mapping \  
> -b ncdg_ip_udp:16.20.16.64[3424] \  
> -i EC1EEB60-5943-11C9-A309-08002B102989,1.1 \  
> -o 30DBEEA0-FB6C-11C9-8EEA-08002B0F4528 \  
> ncdg_ip_udp:16.20.16.44  
rpccp>
```

remove mapping(8rpc)

Related Information

Commands: **show mapping(8rpc)**, **show server(8rpc)**

remove member(8rpc)

remove member

Purpose Removes a specified member from a group

Synopsis `rpccp remove member group-entry-name -m member [-s syntax]`

Options

-m	Declares the entry name of the group member to be removed (required).
-s	Indicates the name syntax of the entry name (optional). The only value for this option is the dce name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the -s option is unnecessary.

Arguments

group-entry-name

Indicates the name of the target group. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **remove member** command removes a specified member from a specified group.

Privilege Required

You need read permission and write permission to the CDS object entry (the target group entry).

Examples The following commands run RPCCP and remove the member `/LandS/anthro/Cal_host_2` from the group `/LandS/dept/Calendar_group`:

```
$ rpccp
rpccp> remove member \
> -m /LandS/anthro/Cal_host_2 \
> /LandS/anthro/Calendar_group
```

remove member(8rpc)

The following command removes the member `/LandS/anthro/Cal_host_3` from the group `/LandS/anthro/Calendar_group`:

```
$ rpecp remove member \  
> -m /LandS/anthro/Cal_host_3 \  
> /LandS/anthro/Calendar_group
```

Related Information

Commands: `add member(8rpc)`, `remove group(8rpc)`, `show group(8rpc)`

remove profile(8rpc)

remove profile

Purpose Removes all profile elements and the profile from the specified name service entry

Synopsis `rpccp remove profile profile-entry-name [-s syntax]`

Options `-s` Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the `-s` option is unnecessary.

Arguments

profile-entry-name

Indicates the name of the target profile. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **remove profile** command removes a profile (and all of its elements) from the name service database. The entry name of the profile is required.

Privilege Required

You need write permission to the CDS object entry (the target profile entry).

Examples The following commands run RPCCP and remove the profile named `/LandS/anthro/molly_o_profile`:

```
$ rpccp
rpccp> remove profile /LandS/anthro/molly_o_profile
```

Related Information

Commands: **add element(8rpc)**, **remove element(8rpc)**, **show profile(8rpc)**

show entry

Purpose Shows the NSI attributes of a name service entry

Synopsis `rpcce show entry entry-name [-i if-id] [-s syntax] [-u]`

Options **-i** Selects a specified interface identifier (optional). Only elements containing that identifier are shown.

The interface identifier value has the following form:

interface-uuid,major-version.minor-version

The UUID is a hexadecimal string and the version numbers are decimal strings, for example:

-i EC1EEB60-5943-11C9-A309-08002B102989,1.1

Leading zeros in version numbers are ignored.

-s Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

-u Updates the local copy of name service data (optional). Name service data that is requested by applications is sometimes stored locally. If a local copy of name service data satisfies an NSI command, the RPC control program uses the local copy. Local copies of name service data are not automatically updated. To display the current contents of an entry that has changed recently, specify the **-u** option.

Arguments

entry-name Indicates the name of the target name service entry. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **show entry** command shows the NSI attributes of a name service entry. The name of the entry is required.

show entry(8rpc)

Note that this operation shows all of the compatible bindings for a given interface.

The **show entry** command shows the same list of string bindings as the **import** operation returns for the specified entry. This list includes all string bindings that refer to a major version that matches the specified version and a minor version that is equal to or greater than the specified version. The list may include string bindings exported for other versions of the interface that are upwardly compatible, rather than for this particular version of the interface.

Privilege Required

You need read permission to the CDS object entry (the target name service entry).

Examples The following command operates from the system prompt to show the name service entry **/LandS/anthro/calendar_mgr_node_3**.

```
$ rpscc show entry /LandS/anthro/Cal_host_3
```

The following commands run the control program and show the name service entry **/LandS/anthro/Calendar_group**:

```
$ rpscc  
rpscc> show entry \  
> /LandS/anthro/Calendar_group
```

Related Information

Commands: **add entry(8rpc)**, **remove entry(8rpc)**

show group

Purpose Shows the members of a group

Synopsis **rpcpp show group** *group-entry-name* [-m *member*] [-r [*integer*]]
[-s *syntax*] [-u]

Options

- m** Declares the name of a single group member.
- r** Indicates that the **show group** operation recurses. If any members of a group are also groups, their entries are shown. By default, the **-r** option causes the **show group** operation to recurse until all nested groups are expanded. For example, **-r** shows the members of the specified group and all nested groups.

You can limit recursion to one or more levels by specifying a decimal integer as part of the **-r** option. For example, **-r 1** shows the members of the specified group and, for members that are groups, the command also shows their members; then recursion stops.

Without the **-r** option, only the members of the specified group are shown.
- s** Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.
- u** Updates the local copy of name service data (optional). Name service data that is requested by applications is sometimes stored locally. If a local copy of name service data satisfies an NSI command, the RPC control program uses the local copy. Local copies of name service data are not automatically updated. To display the current contents of an entry that has changed recently, specify the **-u** option.

Arguments

group-entry-name

Indicates the name of the target group. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

show group(8rpc)**Description**

The **show group** command shows the members of a group in the name service database. The entry name of the group is required. Unless it is limited to a specific member (by the **-m** option), the **show group** command shows all members. The command shows only the members in the specified group; the **-r** option enables you to show members of nested groups.

Privilege Required

You need read permission to the CDS object entry (the target group entry). If you use the **-r** option, you also need read permission to any nested groups.

Examples The following example shows all the members of the group **/LandS/anthro/Calendar_group**, in the order in which they were added to the group:

```
$ rpccp
rpccp> show group /LandS/anthro/Calendar_group
```

The following command operates from the system prompt to show a specific member of the group **/LandS/dept/Calendar_group**:

```
$ rpccp show group \  
> -m /LandS/anthro/Cal_host_2 \  
> /LandS/anthro/Calendar_group
```

Related Information

Commands: **add member(8rpc)**, **remove group(8rpc)**, **remove member(8rpc)**

show mapping

Purpose Shows the elements of the either the local or a remote endpoint map

Synopsis `rpccep show mapping [host-address] [-i if-id [-v versions]]`
`[-o object-uuid [-o object-uuid...]]`

Options **-i** Defines an interface identifier to be shown (optional). Only one interface can be shown in a single operation. If specified, only elements containing this interface identifier are shown. The **-i** option can be qualified by the **-v** option. The value has the following form:

interface-uuid,major-version.minor-version

The UUID is a hexadecimal string and the version numbers are decimal strings, for example:

-i EC1EEB60-5943-11C9-A309-08002B102989,1.1

Leading zeros in version numbers are ignored.

-v Indicates how a specified interface version is used (optional). If it is used without the **-i** option, the **-v** option is ignored. The possible combinations of versions for the **-v** option and their actions are described in the following table.

show mapping(8rpc)

Versions	Action
all	The interface version is ignored.
exact	Both the major and minor versions must match the specified versions.
compatible	The major version must match the specified version, and the minor version must be greater than or equal to the specified version.
major_only	The major version must match the specified version; the minor version is ignored.
upto	The major and minor versions must be less than or equal to those specified.

If the **-v** option is absent, the command shows compatible version numbers.

-o Defines an object to be shown (optional). Each **show mapping** command accepts up to 32 **-o** options.

The UUID is a hexadecimal string, for example:

-o 3C6B8F60-5945-11C9-A236-08002B102989

Arguments

host-address The **host-address** argument is a string binding that indicates where to find the target endpoint map. When accessing the local endpoint map, you can specify which protocol sequence to use (optional); for example,

ncadg_ip_udp:

When accessing a remote endpoint map, you must specify both a protocol sequence and a network address for the remote system (required); for example,

ncadg_ip_udp:16.20.16.44

An endpoint is unnecessary in local or remote host addresses, and the **remove mapping** command ignores any endpoint specified as part of a host address.

Description

The **show mapping** command shows elements of an endpoint map. Each element corresponds to an object UUID, interface identifier, annotation, and binding

information. The binding information contains an RPC protocol sequence, a network address, and an endpoint within square brackets (*rpc-prot-seq:network-addr[endpoint]*).

The endpoint map can be either the local endpoint map or the endpoint map of a specified remote host. If entered without a remote host address, the command accesses the local endpoint map. For the local endpoint map, a **show mapping** command without any options displays all the map elements. For a remote endpoint map, map elements are accessible only for protocol sequences that are supported on both your system and the remote system.

The options list a selected subset of map elements. The **-i** option selects a specific interface, and the **-v** option qualifies the **-i** option. The **-o** object selects a specific object. You can use from 0 to 32 **-o** options per command. The options work together to specify the subset of elements for the target protocol sequence(s).

Notes

To ensure that you can remotely display all map elements from every remote endpoint map, run the RPC control program on a system that supports all of the protocol sequences available in your network environment.

Examples

The following commands start the control program and show the map elements in the local endpoint map that contain the specified interface identifier:

```
$ rpscp
rpscp> show mapping -i EC1EEB60-5943-11C9-A309-08002B102989,1.1
```

The following **rpscp show mapping** command operates from the system prompt. The command accesses the endpoint map of the remote host specified by the host address (*ncadg_ip_udp:16.20.16.44*) and displays the one map element that contains both the specified interface identifier and the specified object UUID.

```
$ rpscp show mapping \
> -i EC1EEB60-5943-11C9-A309-08002B102989,1.1 \
> -o 30DBEEA0-FB6C-11C9-8EEA-08002B0F4528 \
> ncadg_ip_udp:16.20.16.44
```

Related Information

Commands: **remove mapping(8rpc)**, **show server(8rpc)**

show profile(8rpc)

show profile

Purpose Shows the elements of a profile

Synopsis `rpccp show profile profile-entry-name`
{`-d` | `-a annotation` | `-i if-id` [`-v versions`] `-m member`}
[`-r [integer]`] [`-s syntax`] [`-u`]

Options

- d** Selects the default profile element. With the `-d` option, the `-a`, `-i`, and `-m` options are ignored.
Note that the `-a` option works with the `-d` option, but do not use them together.
- a** Declares a single annotation field (optional). The `-a` option selects only elements containing the specified annotation. The option is case sensitive.
The `-a` option works alone or in combination with the `-i` or `-m` options or both; only elements containing all the specified values are displayed.
Note that the shell supports quotation marks around the annotation field of profile elements, allowing you to include internal spaces in an annotation; the control program does not. To specify or refer to annotations from within the control program, limit each annotation to an unbroken alphanumeric string; for example, `CalendarGroup`. To refer to annotations from the system prompt, do not incorporate quotation marks into any annotation.
- i** Selects a specified interface identifier (optional). Only elements containing that interface identifier are shown.
The interface identifier value has the following form:
interface-uuid,major-version.minor-version
The UUID is a hexadecimal string and the version numbers are decimal strings, for example:
-i EC1EEB60-5943-11C9-A309-08002B102989,1.1
Leading zeros in version numbers are ignored.
The `-i` option works alone or in combination with the `-a` or `-m` options or both; only elements containing all the specified values are displayed. When the `-d` option is specified, the `-i` option is ignored.

- v** Indicates how a specified interface version is used (optional). If it is used without the **-i** option, the **-v** option is ignored. The possible combinations of versions for the **-v** option and their actions are described in the following table.

Versions	Action
all	The interface version is ignored.
exact	Both the major and minor versions must match the specified versions.
compatible	The major version must match the specified version, and the minor version must be greater than or equal to the specified version.
major_only	The major version must match the specified version; the minor version is ignored.
upto	The major and minor versions must be less than or equal to those specified.

If the **-v** option is absent, the command shows compatible version numbers.

- m** Declares a single member name (optional). Only elements containing that member name are shown.

The **-m** option works alone or in combination with the **-a** or **-i** options or both; only elements containing all the specified values are displayed. When the **-d** option is specified, the **-m** option is ignored.

- r** Indicates that the **show profile** operation recurses. If the member of any element of a profile is also a profile, its elements are shown. By default, the **-r** option causes the show profile operation to recurse until all nested profiles are expanded; for example, **-r** shows the elements of the specified profile and of all nested profiles.

You can limit recursion to one or more levels by specifying a decimal integer as part of the **-r** option. For example, **-r 1** shows the elements of the specified profile and, for element members that are profiles, the command also shows their elements; then recursion stops.

Without the **-r** option, only the profile elements in the specified entry are shown.

- s** Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

show profile(8rpc)

- u** Updates the local copy of name service data (optional). Name service data that is requested by applications is sometimes stored locally. If a local copy of name service data satisfies an NSI command, the RPC control program uses the local copy. Local copies of name service data are not automatically updated. Specify the **-u** option to display the current contents of an entry that has changed recently.

Arguments*profile-entry-name*

Indicates the name of the target profile. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **show profile** command shows the elements of a profile in the name service database. The entry name of the profile is required.

By default, all elements in the profile are shown. You can select a subset of the elements by specifying the **-a**, **-i**, or **-m** options. The **-r** option enables you to show nested profiles.

Privilege Required

You need read permission to the CDS object entry (the target profile entry). If you use the **-r** option, you also need read permission to any nested profiles.

Examples The following command operates from the system prompt to show the cell **profile /cell_profile** in the local cell:

```
$ rpccp show profile ./cell-profile
```

The C-Shell **setenv** command sets up an environment variable **MOLLY_O_PROFILE**, which represents the user profile **/LandS/anthro/molly_o_profile**. The control program commands start the control program and show the user profile associated with the **MOLLY_O_PROFILE** environment variable, as follows:

```
$ setenv MOLLY_O_PROFILE \  
> /LandS/anthro/molly_o_profile  
$  
$ rpccp  
rpccp> show profile MOLLY_O_PROFILE
```

Related Information

Commands: **add element(8rpc)**, **remove element(8rpc)**, **remove profile(8rpc)**

show server(8rpc)

show server

Purpose Shows the binding information, interface identifiers, and object UUIDs in a server entry

Synopsis **rpcpp show server** *server-entry-name* [-i [*if-id*]] [-o [*object-uuid*]] [-s *syntax*] [-u]

Options **-i** Shows interface identifiers from binding information found in the entry (optional). Without the **-i** option, the command displays all interface identifiers.

To display a specific interface, supply its identifier as the value. The value has the following form:

interface-uuid,major-version.minor-version

The UUID is a hexadecimal string and the version numbers are decimal strings, for example:

-i EC1EEB60-5943-11C9-A309-08002B102989,1.1

Leading zeros in version numbers are ignored.

-o Shows object UUIDs found in the entry (optional). Without the **-o** option, the command displays all object UUIDs. To display a specific object UUID, supply its string representation as the value, for example:

-o 3C6B8F60-5945-11C9-A236-08002B102989

-s Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

-u Updates the local copy of name service data (optional). Name service data that is requested by applications is sometimes stored locally. If a local copy of name service data satisfies an NSI command, the RPC control program uses the local copy. Local copies of name service data are not automatically updated. Specify the **-u** option to display the current contents of an entry that has changed recently.

Arguments

server-entry-name

Indicates the name of the target server. For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

Description

The **show server** command shows the RPC binding information, interface identifiers, and object UUIDs in a server entry. The entry name of the server entry is required.

This operation shows all of the potential bindings for an interface. By default, this command displays bindings for the specified version of the interface and for upwardly compatible versions of the interface. The **-v** option controls which versions are targeted by this command.

Privilege Required

You need read permission to the CDS object entry (the target server entry).

Examples The following commands start the control program and show the server entry **/LandS/anthro/Cal_host_2** in the local cell:

```
$ rpccp
rpccp> show server /LandS/anthro/Cal_host_2
```

The following command operates from the system prompt to display a specific object and interface from the server entry **/LandS/anthro/Cal_host_2** in the local cell:

```
$ rpccp show server \  
> /LandS/anthro/Cal_host_2 \  
> -o 16977538-E257-11C9-8DC0-08002B0F4528 \  
> -i EC1EEB60-5943-11C9-A309-08002B102989,1.1
```

Related Information

Commands: **export(8rpc)**, **import(8rpc)**, **unexport(8rpc)**

unexport(8rpc)

unexport

Purpose Removes binding information, interface identifiers, and object UUIDs from a server entry

Synopsis `rpccp unexport entry-name [[-i if-id [-v versions]] | [-o object-uuid]] [-s syntax]`

Options **-i** Defines an interface identifier to be unexported (optional). Only one interface can be unexported in a single operation. If specified, binding information for this interface is removed from the entry. The **-i** option can be qualified by the **-v** option. The value has the following form:

interface-uuid,major-version.minor-version

The UUID is a hexadecimal string and the version numbers are decimal strings, for example:

-i EC1EEB60-5943-11C9-A309-08002B102989,1.1

Leading zeros in version numbers are ignored.

-v Indicates how a specified interface version is used (optional). If it is used without the **-i** option, the **-v** option is ignored. The possible combinations of versions for the **-v** option and their actions are described in the following table.

Versions	Action
all	The interface version is ignored.
exact	Both the major and minor versions must match the specified versions.
compatible	The major version must match the specified version, and the minor version must be greater than or equal to the specified version.
major_only	The major version must match the specified version; the minor version is ignored.
upto	The major and minor versions must be less than or equal to those specified.

If the **-v** option is absent, the command shows compatible version numbers.

-o Defines an object to be unexported (optional). Each **unexport** command accepts up to 32 **-o** options.

The UUID is a hexadecimal string, for example:

-o 3C6B8F60-5945-11C9-A236-08002B102989

-s Indicates the name syntax of the entry name (optional). The only value for this option is the **dce** name syntax, which is the default name syntax. Until an alternative name syntax becomes available, specifying the **-s** option is unnecessary.

-u Updates the local copy of name service data (optional). Name service data that is requested by applications is sometimes stored locally. If a local copy of name service data satisfies an NSI command, the RPC control program uses the local copy. Local copies of name service data are not automatically updated. If an entry has changed recently, to display its current contents, specify the **-u** option.

Arguments

entry-name Indicates the name of the target name service entry. Usually, the target is a server entry. However, objects also can be exported (without an interface identifier or binding information) to a group or a profile.

For an entry in the local cell, you can omit the cell name and specify only the cell-relative name.

unexport(8rpc)

Description

The **unexport** command removes binding information and an interface identifier, object UUIDs, or both from a server entry, or it removes object UUIDs from a group's entry. The command requires the entry name and either the interface identifier or one or more object UUIDs.

By default, the **unexport** operation removes **compatible** interface versions.

Privilege Required

You need both read permission and write permission to the CDS object entry (the target name service entry).

Examples The C-Shell **setenv** command sets up an environment variable **Calendar_1_1**, which represents the interface identifier of an RPC interface. The control program commands start the control program and remove (unexport) the Calendar Version 1.1 interface from the server entry **/LandS/anthro/Cal_host_2** in the local cell, as follows:

```
$ setenv Calendar_1_1 EC1EEB60-5943-11C9-A309-08002B102989,1.1

$ rpccp
rpccp> unexport \
> -i Calendar_1_1 \
> /LandS/anthro/Cal_host_2
rpccp>
```

Related Information

Commands: **export(8rpc)**, **import(8rpc)**, **show server(8rpc)**

rpcd

Purpose DCE Remote Procedure Call daemon

Synopsis `rpcd [-vuf] [protseq...]`

Options

- v** Prints the **rpcd** version and exits.
- u** Prints the **rpcd** usage and exits.
- f** Starts the RPC daemon process in the foreground. The default is for RPC daemon to run in the background — that is, to fork and have the parent exit.

Arguments

protseq An RPC protocol sequence string. Possible values include **ncadg_ip_udp** (for a datagram protocol) and **ncacn_ip_tcp** (for a connection-based protocol). A complete list of the protocol sequences recognized by the Endpoint Map Service can be found in **dce/ep.idl**.

Description

The DCE Remote Procedure Call daemon is a process that provides the Endpoint Map Service for the local host. The Endpoint Map Service is provided on a per host basis. On each host, it maintains a local endpoint map, which is a database that maps interfaces, object UUIDs, and protocol sequence registrations to servers (endpoints). The Endpoint Map Service on each host uses the local endpoint map to locate a compatible server (endpoint) for clients that do not know the endpoint of a compatible server. Servers register their bindings (endpoints) with their local Endpoint Map Service. The RPC daemon provides the Endpoint Map Service for the local host.

Any host that runs RPC-based servers must run the RPC daemon. The RPC daemon must be running before any RPC servers are started. Typically, the RPC daemon starts each time a host boots. Usually, a *dcelocal/etc/dce.rc* file is created with a link from */etc/dce.rc* and with */etc/rc* or */etc/rc.local* invoking */etc/dce.rc*. The *dcelocal/etc/dce.rc* file is responsible for deleting the database file(s) and starting the RPC daemon.

rpcd(8rpc)

The endpoint map is retained in persistent storage (on disk in *dcelocal/var/rpc/rpcdep.dat*) so that the RPC daemon can be stopped and restarted without requiring servers to re-register with the RPC daemon. After a system reboot, RPC-based servers restart and re-register with the Endpoint Map Service, so the database file needs to be deleted before the RPC daemon starts. On most systems, **rpcdep.dat** is deleted at system boot time.

By default, the RPC daemon listens on one well-known port for each RPC protocol sequence (that is, each combination of an RPC protocol and a transport protocol) supported by the host on which it is running. The optional *protseq* argument lets you limit the protocol sequences that the RPC daemon listens on to only those protocol sequences that you specify.

For backward compatibility, the RPC daemon also provides the services previously supplied by the NCS 1.5.1 Local Location Broker (LLB) daemon, **llbd**. For the most part, the LLB and Endpoint Map components of the RPC daemon are independent. They are separate logical databases and in fact have separate physical database files (names) and administrative utilities. RPCCP administers only the endpoint map component; the NCS 1.5.1 **lb_admin** program administers only the LLB component.

The only time that the two components are used together is when the RPC daemon is performing the Endpoint Map mapping function or the LLB forwarding function. This is to allow 1.5.1 clients to use compatible DCE servers and DCE clients to use compatible 1.5.1 servers. The RPC daemon first searches the Endpoint Map for a compatible server and, if none is found, it then searches the LLB database.

Along with the endpoint map database (**rpcdep.dat**), the RPC daemon creates the NCS 1.5.1 LLB file, which resides on disk in *dcelocal/var/rpc/rpcdllb.dat*. Like **rpcdep.dat**, **rpcdllb.dat** needs to be deleted at system boot time.

Only a single RPC daemon can run on a host. A host cannot run the RPC daemon and the NCS 1.5.1 **llbd** simultaneously. The RPC daemon process must run with superuser privileges since it typically listens on privileged or reserved network ports.

Note that you can kill and restart the RPC daemon after the system has booted. The restarted RPC daemon will **not** lose any previously registered server bindings. It simply reloads its internal databases from the **rpcdep.dat** and **rpcdllb.dat** files.

Files

dcelocal/var/rpc/rpcdep.dat

dcelocal/etc/dce.rc

/etc/dce.rc

/etc/rc

/etc/rc.local

Related Information

Library calls: **rpc_ep_register(3rpc)**, **rpc_ep_unregister(3rpc)**

Books: *OSF DCE Administration Guide*

Chapter 2

Directory Service Commands

intro

Purpose Introduction to the CDS commands

Description

The DCE Cell Directory Service provides the following management commands:

cadsbrowser Starts the CDS Browser utility. This utility is based on the OSF/Motif™ graphical user interface. The Browser can display an overall directory structure as well as show the contents of directories.

cads Starts the CDS control program. Use this command line interface to manage the CDS components and the contents of your namespace.

The following commands are typically started automatically by scripts that execute as part of normal system start-up procedures. See the reference pages for these commands before you try to use them.

cadsadv Starts the advertisement and solicitation daemon on the local system and then starts clerks as needed by applications. Use this command only when troubleshooting because it creates and automatically starts the CDS clerk whenever the host system is rebooted.

cadsd Restarts the CDS server. Use this command only when troubleshooting because it starts the CDS server process automatically whenever the host system is rebooted.

gdad Starts the GDA (Global Directory Agent) daemon. The GDA enables intercell communication, serving as a connection to other cells through the global naming environment. The GDA is typically started automatically by scripts that execute as part of normal system start-up and shutdown procedures.

Related Information

Book: *OSF DCE Administration Guide*

Commands: **cdsadv(8cds)**, **cdsbrowser(8cds)**, **cdsep(8cds)**, **cdsd(8cds)**,
gdad(8cds), **rpcd(8rpc)**

add directory(8cdfs)

add directory

Purpose Adds a value to a modifiable, set-valued attribute (including application-defined attributes) of a directory

Synopsis `cdscp add directory directory-name attribute-name = attribute-value`

Arguments

directory-name

The full name of the directory.

attribute-name

The name of a particular attribute. Specify only one attribute at a time. See the **cds_attributes** file for the list of attributes and corresponding data types that your application uses.

attribute-value

The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute.

Description

The **add directory** command adds a value to a modifiable, set-valued attribute (including application-defined attributes) of a directory. If the attribute does not exist, this command creates it. Usually, this task is performed through the client application. See the *OSF DCE Administration Guide* for more information about attributes.

Privilege Required

You must have write permission to the directory.

Examples To add the value **ontario** to the attribute **myname** of a directory named **./:/sales**, read the **cds_attributes** file to verify that the attribute shown in the following display exists:

OID	LABEL	SYNTAX
1.3.22.1.3.91	myname	char

Enter the following command to assign the value **ontario** to the attribute **myname**:

```
cdscp> add directory ./:/sales myname = ontario
```

Related Information

Commands: **remove directory(8cds)**, **show directory(8cds)**

Books: *OSF DCE Administration Guide*

add object(8cads)

add object

Purpose Adds a value to a modifiable, set-valued attribute (including application-defined attributes) of an object entry

Synopsis `cdscp add object object-name attribute-name = attribute-value`

Arguments

object-name The full name of the object entry.

attribute-name

The name of a particular attribute. Specify only one attribute at a time. See the `cads_attributes` file for the list of attributes and corresponding data types that your application uses.

attribute-value

The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute.

Description

The `add object` command adds a value to a modifiable, set-valued attribute (including application-defined attributes) of an object entry. If the attribute does not exist, this command creates it. Usually, this task is performed through the client application. See the *OSF DCE Administration Guide* for more information about attributes.

Privilege Required

You must have write permission to the object entry.

Examples To add the value `ps` to the attribute `printcap` of an object entry named `./:./subsys/deskprinter`, read the `cads_attributes` file to verify that the attribute shown in the following display exists:

OID	LABEL	SYNTAX
1.3.22.1.3.70	printcap	char

Enter the following command to assign the value `ps` to the attribute `printcap`:

```
cdscp> add object ./:./subsys/deskprinter printcap = ps
```

Related Information

Commands: **create object(8cds)**, **delete object(8cds)**, **list object(8cds)**, **remove object(8cds)**, **set object(8cds)**, **show object(8cds)**

Books: *OSF DCE Administration Guide*

cdsadv

- Purpose** Starts the advertisement and solicitation daemon
- Synopsis** `cdsadv [-d] [-e*] [-s] [-c] [-v]`
- Options**
- d** For debugging use only. Causes the **cdsadv** process not to fork and turns on tracing.
 - e** Prints error messages. Use a \ (backslash) character to escape the shell. Use an * (asterisk) character to indicate full error messages (-e*) or use a fully qualified filename.
 - s** Causes the server not to send or receive advertisements. This argument can be used for diagnostic work involving multiple servers on the same local area network to limit access to those servers identified with the **define cached server** command.
 - c** Specifies cache size in kilobytes.
 - v** Prints initialization progress messages; these verify that initialization successfully completed.

Description

The **cdsadv** command starts the advertisement and solicitation daemon on the local system and then starts clerks as needed by applications. This command creates and automatically starts the CDS clerk whenever the host system is rebooted.

Privilege Required

You must log in as superuser (root).

- Notes** This command is ordinarily executed by the CDS startup script on the system where the CDS server is running. You should use this command interactively only when a server fails to start automatically upon reboot, or if you want to restart a clerk or server that you disabled to perform a backup or to do diagnostic work on the host system.

Examples To restart a clerk, follow these steps:

1. Log in to the clerk system as superuser (root).
2. Log in to DCE as the machine principal of the local host. Enter the principal name in the format **hosts/hostname/self** as shown in the following example command for a host named **orion** whose password is **smith**:

```
# dce_login hosts/orion/self smith
```
3. Enter the following command to see if the **rpcd** process is already running:

```
# ps
```

If the **rpcd** process appears on the list of active processes, proceed to step 4. If the **rpcd** process does not appear on the list of active processes, enter the following command to start the process:

```
# rpcd
```
4. Enter the following command to start the **cadsadv** process:

```
# cadsadv
```

Related Information

Books: *OSF DCE Administration Guide*

cdsbrowser

Purpose Starts the CDS Browser utility on the local system

Synopsis `cdsbrowser`

Description

The `cdsbrowser` command starts the CDS Browser utility on the local system. This utility runs on workstations with windowing software based on the OSF/Motif™ graphical user interface. Using a mouse to manipulate pull-down menus, you can view the directory structure of a namespace, view child directories of a particular directory, view the object entries and soft links in a directory, and set a filter to display only object entries of a particular class. (For users who do not have windowing software, similar functions are available with the control program.)

Examples The following command starts the CDS Browser utility on the local system:

```
$ cdsbrowser
```

Related Information

Books: *OSF DCE Administration Guide*

cdsclerk

Purpose Manages the interface between clients and the CDS server

Synopsis `cdsclerk [-d] [-e*]`

Options

- d** For debugging use only. Causes the CDS clerk process not to fork and turns on tracing.
- e** Prints error messages. Use a \ (backslash) to escape the shell. Use a * (asterisk) to indicate full error messages (-e*) or use a fully qualified filename.

Description

The `cdsclerk` command manages the interface between clients and the CDS server.

Privilege Required

You must log in as superuser (root).

Notes This command is used by the advertiser on the system where the CDS clerk is running. You should use this command interactively only to do diagnostic work on the host system.

Examples Before you run the following process, make sure that other clerks are not running. To start the `cdsclerk` process, follow these steps:

1. Make sure that a CDS server is already running somewhere within the cell.
2. Log in to the system as superuser (root).
3. Log in to DCE as the machine principal of the local host. Enter the principal name in the format `/hosts/hostname/self` as shown in the following example command for a host named **orion** whose password is **smith**:

```
# dce_login hosts/orion/self smith
```
4. Enter the following command to see if the **rpcd** process is already running:

```
# ps
```

cdsclerk(8cds)

If the **rpcd** process appears on the list of active processes, proceed to step 5. If the **rpcd** process does not appear on the list of active processes, enter the following command to start the process:

```
# rpcd
```

5. Enter the following command to start the **cdsadv** process:

```
# cdsadv
```

6. Enter the following command with the appropriate arguments to start the **cdsclerk** process:

```
# cdsclerk
```

Related Information

Books: *OSF DCE Administration Guide*

cdscp

Purpose Starts the CDS control program

Synopsis `cdscp` [*cdscp-command*]

Arguments

cdscp-command

Optionally, specifies one of the following control commands:

add directory

Adds a value to a modifiable, set-valued attribute (including application-defined attributes) of a directory

add object

Adds a value to a modifiable, set-valued attribute (including application-defined attributes) of an object entry

clear cached server

Removes knowledge of a server that you had specifically defined from the local clerk's cache

clear clearinghouse

Removes knowledge of the specified clearinghouse from the server's memory

create child

Creates a child pointer at the master replica of the parent directory

create clearinghouse

Creates a clearinghouse on the local server system or makes an existing clearinghouse available

create directory

Creates a directory

create link

Creates a soft link and optionally specifies an expiration time and an extension time

create object

Creates a new object entry

cdscp(8c ds)

create replica

Creates a replica of an existing directory in the specified clearinghouse

define cached server

Creates knowledge of a server in the local clerk's cache

delete child Deletes a child pointer from the namespace

delete clearinghouse

Deletes the specified clearinghouse from the local server system

delete directory

Deletes a directory

delete link Deletes a soft link

delete object Deletes an object entry

delete replica

Deletes a read-only replica of a directory from a clearinghouse

disable clerk Stops the clerk on the local system

disable server

Stops the server on the local system

dump clerk cache

Displays the contents of the clerk cache

help

Displays a list of the CDS control program commands

list child

Displays a list of all the child pointers whose names match the specified child name

list clearinghouse

Displays a list of all the clearinghouses whose names match the specified clearinghouse name

list directory

Displays a list of all the directories whose names match the specified directory name

list link

Displays a list of all the soft links whose names match the specified link name

- list object** Displays a list of all the object entries (including clearinghouse object entries) whose names match the specified object entry name
- remove directory**
Removes a value from a set-valued or single-valued attribute (including application-defined attributes) of a directory
- remove link** Removes a soft link's timeout value attribute
- remove object**
Removes a value from a set-valued or single-valued attribute (including application-defined attributes) of an object entry
- set cdscp confidence**
Sets the confidence level of clerk calls issued as a result of CDS control program commands
- set cdscp preferred clearinghouse**
Specifies a preferred clearinghouse to use for satisfying read requests that result from CDS control program commands
- set directory** Changes the value of a modifiable, single-valued attribute of a directory
- set directory to new epoch**
Reconstructs a directory's replica set, allowing you to designate a new master replica or to exclude a replica
- set directory to skulk**
Starts the skulk of a directory immediately
- set link** Changes the value of a modifiable, single-valued attribute of a soft link
- set object** Changes the value of a modifiable, single-valued attribute of an object entry
- show cached clearinghouse**
Displays current information about the specified cached clearinghouse

cdscp(8c ds)

- show cached server** Displays address information of a server in the local clerk's cache
- show cdscp confidence** Displays the current confidence level of clerk calls resulting from CDS control program commands
- show cdscp preferred clearinghouse** Displays the preferred clearinghouse for satisfying read requests that result from CDS control program commands
- show cell** Displays the information you need to create a cell entry in either DNS or GDS
- show child** Displays attribute information about the specified child pointer
- show clearinghouse** Displays attribute information about the specified clearinghouse
- show clerk** Displays attribute information about the CDS clerk on the local system
- show directory** Displays attribute information about the specified directory
- show link** Displays attribute information about the specified soft link
- show object** Displays attribute information about the specified object entry
- show replica** Displays attribute information about the specified replica
- show server** Displays attribute information about the server running on the local system

Description

The Cell Directory Service (CDS) control program is a command line interface for managing the components of the Cell Directory Service and the contents of the namespace.

You can use the control program commands from within the control program or from the system prompt. To use the control program commands from inside the control program, start the control program by using the **cdscp** command alone, without any argument. This enters the control program, which displays the control program prompt (**cdscp>**):

```
$ cdscp
cdscp>
```

At this prompt, you can enter any control program command. For example:

```
cdscp> show server
```

Use the command **do filename** from inside the control program to read a file of commands.

To leave the control program and return to the system prompt, use the **quit** command. To use the control program commands from the system prompt, enter the **cdscp** command with an internal command of the CDS control program as the first argument. The control program executes the command immediately, without displaying the control program prompt. For example, you can enter the **show server** command as follows:

```
$ cdscp show server
```

Elements of a CDS Command

All CDS control program commands must include a verb, an entity name, and all required arguments. Depending on the command, you can also specify optional arguments and attributes. A space must separate more than one attribute or argument. A space must precede and follow an = (equals sign).

Verbs

The following is a list of the definitions of verbs used in control program commands:

add	Adds a value to a modifiable, set-valued attribute
clear	Removes knowledge of a cached clearinghouse or cached server from memory
create	Creates an entity
define	Creates knowledge of a locally cached server
delete	Deletes an entity
disable	Stops operation of a clerk or server

dump	Displays the contents of a clerk cache
list	Displays a list of specified entity names
remove	Removes a value from a set-valued or single-valued attribute
set	Changes the value of a modifiable, single-valued attribute
show	Displays attribute information

Entity Names

Any individually manageable piece of CDS is called an entity. A set of commands exists for each entity. The following is a list of the entities and a description of what each entity represents:

Cached Clearinghouse

A cached clearinghouse is a clearinghouse that a clerk has discovered and cached. A clerk can learn about clearinghouses as a result of configuration information, advertisements received on a LAN, or during the process of finding a name.

Cached Server

A cached server is a server that a clerk has cached as a result of manual configuration through the control program.

Child

A child pointer connects a parent and child directory in a hierarchical namespace. The child pointer is stored in the parent directory and has the same name as the child directory.

Clearinghouse

A clearinghouse is a database containing a collection of directory replicas at a particular server.

Clerk

The clerk is the interface between client applications and servers.

Directory

A directory contains child, object, and link entries that are logically stored under one name (the directory name).

Link

A soft link is a pointer providing an alternate name for an object entry, directory, or other soft link.

Object

An object entry represents a resource (for example, an application) that is named in the namespace.

Replica

A replica is a copy of a directory. Each copy, including the original or master, is referred to as a replica.

Server A server handles lookup requests from clerks and maintains the contents of the clearinghouse or clearinghouses at its node.

Attributes

Every CDS entity has attributes, which are pieces or sets of data associated with that entity. Attributes can reflect or affect the operational behavior of an entity, record the number of times a particular event or problem occurred since the entity was last enabled, and uniquely distinguish an entity from any other entity. Some attributes have a single value; others contain a set of values.

CDS attributes are identified by ISO object identifiers (OIDs). Every CDS attribute name maps to an OID and a corresponding data type. Usually, client applications define the name of an attribute and its data type. Application programmers should never need to modify (except for the purpose of foreign language translation) the existing CDS labels associated with the unique OIDs in the `cds_attributes` file. However, programmers can obtain new OIDs from the appropriate allocation authority, create new attributes for their own object entries, and then append them to the existing list. The OID and data type of each attribute are stored in the file `/opt/dcelocal/etc/cds_attributes`. Descriptions of the CDS data types that applications can use are in the `cdsclerk.h` file.

All entities have **show** commands that you can use to display the names and values of specific attributes or all attributes. For more information about CDS attributes, see the *OSF DCE Administration Guide*.

Editing the Commands

You can abbreviate commands, continue a command beyond one line, or redirect output to a file within the control program.

To abbreviate any command name, type only the first four characters. You can abbreviate a command name to fewer than four characters as long as the abbreviated name remains unique among all command names in the control program. For example, the following commands are equivalent:

```
cdscp> show directory ./:sales
cdscp> sh dir ./:sales
```

To continue a long command line onto the next line, type a space and then a \ (backslash) at the end of the first line, for example:

```
cdscp> set link ./:sales CDS_LinkTimeout \
> (1991-12-31-12:00:00 090-00:00:00)
```

To add a comment, use the # (number sign). Everything following the # character on a line is ignored.

cdscp(8cnds)

To redirect output to a file, most UNIX shell users can type `>filename` at the shell prompt. To redirect output of error text to a file, most UNIX shell users can type `>&filename` at the shell prompt.

For example, the following command redirects the display produced by the **show directory** command to a new text file named **directory_names**:

```
$ cdscp show directory ./:* >directory_names
```

Using Wildcard Characters

When entering a name in **show** and **list** commands, you can use wildcard characters in the rightmost simple name (the name to the right of the last slash (/) in the full pathname). The asterisk (*) matches 0 or more characters in a simple name. The question mark (?) matches exactly one character in a simple name.

When you use an asterisk or a question mark as a normal character in the rightmost simple name of a **show** or **list** command, escape it with a backslash (* or \?). Otherwise, the character is interpreted as a wildcard.

You cannot use wildcard characters in **show clerk** and **show server** commands.

Privilege Required

CDS supports the following DCE permissions: read (**r**), write (**w**), insert (**i**), delete (**d**), test (**t**), control (**c**), and administer (**a**). Each permission has a slightly different meaning, depending on the kind of CDS name with which it is associated. In general, the permissions are defined as follows:

Read	Allows a principal to look up a name and view the attribute values associated with it.
Write	Permission allows a principal to change the modifiable attributes associated with a name, except the name's access control list (ACL) entries.
Insert	Permission (for use with directory entries only) allows a principal to create new names in a directory.
Delete	Permission allows a principal to delete a name from the namespace.
Test	Permission allows a principal to test whether an attribute of a name has a particular value without being able to actually see any of the values (that is, without having read permission to the name). Test permission provides application programs a more efficient way to verify a CDS attribute value. Rather than reading an entire set of values, an application can test for the presence of a particular value.

Control Permission allows a principal to modify the ACL entries associated with a name. Control permission is automatically granted to the creator of a CDS name.

Administer Permission (for use with directory entries only) allows a principal to issue CDS control program commands that control the replication of directories.

The creator of a name is automatically granted all permissions appropriate for the type of name created. For example, a principal creating an object entry is granted read, write, delete, test, and control permission to the object entry. A principal creating a directory is granted read, write, insert, delete, test, control, and administer permission to the directory.

Examples The following command starts the CDS control program:

```
$ cdsep  
cdscp>
```

The following command operates from the system prompt to display the attributes of the CDS clerk on the local system:

```
$ cdscp show clerk
```

Related Information

Books: *OSF DCE Administration Guide*

cdsd

Purpose Restarts the CDS server

Synopsis `cdsd [-d] [-e*] [-a] [-v]`

Arguments

- d** For debugging use only. Causes the **cdsd** process not to fork and turns on tracing.
- e** Prints error messages. Use the \ (backslash) character to escape the shell. Use an * (asterisk) to indicate full error messages (**-e***) or use a fully qualified filename.
- a** Creates a new namespace if there is not an existing namespace. This flag is meaningful only when the cell is first configured (that is, the initial creation of the namespace).
- v** Prints initialization progress messages; these verify that initialization successfully completed.

Description

The **cdsd** command restarts the CDS server. This command starts the CDS server process automatically whenever the host system is rebooted.

Privilege Required

You must log in as superuser (root).

Notes

This command is ordinarily executed by the CDS startup script on the system where the CDS server is running. You should use this command interactively only when a server fails to start automatically upon reboot, or if you want to restart a server that you disabled to perform a backup or to do diagnostic work on the host system.

Examples To restart a server, follow these steps:

1. Log in to the server system as superuser (root).
2. Log into DCE as the machine principal of the local host. Enter the principal name in the format **hosts/hostname/self** as shown in the following example command for a host named **mystic** whose password is **smith**:

```
# dce_login hosts/mystic/self smith
```

3. Enter the following command to see if the **rpcd** process is already running:

```
# ps
```

If the **rpcd** process appears on the list of active processes, proceed to step 4.

If the **rpcd** process does not appear on the list of active processes, enter the following command to start the process:

```
# rpcd
```

4. Enter the following command to see if the **cdsadv** process is already running:

```
# ps
```

If the **cdsadv** process appears on the list of active processes, proceed to step 5.

If the **cdsadv** process does not appear on the list of active processes, enter the following command to start the process:

```
# cdsadv
```

5. Enter the following command to restart the server:

```
# cdsd
```

When the server process starts, it starts all clearinghouses on the system.

Related Information

Books: *OSF DCE Administration Guide*

clear cached server(8cds)

clear cached server

Purpose Removes knowledge of a server that you had specifically defined from the local clerk's cache

Synopsis `cdscp clear cached server name`

Arguments

name The simple name given to the cached server when it is created.

Description

The **clear cached server** command removes knowledge of a server from the local clerk's cache. You can only clear servers that you have specifically created with the **define cached server** command.

Privilege Required

You must have write permission to the clerk.

Examples The following command removes knowledge of the server **nrl** from the clerk cache:

```
cdscp> clear cached server nrl
```

Related Information

Commands: **define cached server(8cds)**, **dump clerk cache**, **show cached server(8cds)**

clear clearinghouse

Purpose Removes knowledge of the specified clearinghouse from the server's memory

Synopsis `cdscp clear clearinghouse clearinghouse-name`

Arguments

clearinghouse-name

The full name of the clearinghouse.

Description

The **clear clearinghouse** command removes knowledge of the specified clearinghouse from the server's memory. The clearinghouse files are not deleted. This ensures that the clearinghouse is not automatically enabled on server restarts. If you issue a **list clearinghouse** command, the clearinghouse will be listed. You must use the **delete clearinghouse** command to remove the clearinghouse.

This command is part of the process of relocating a clearinghouse. See the *OSF DCE Administration Guide* for more information.

Privilege Required

You must have write permission to the server on which the clearinghouse resides.

Examples The following command clears the clearinghouse `./:/Paris2_CH` before moving it to another server:

```
cdscp> clear clearinghouse ./:/Paris2_CH
```

Related Information

Books: *OSF DCE Administration Guide*

Commands: **create clearinghouse(8cds)**, **delete clearinghouse(8cds)**, **list clearinghouse(8cds)**, **set cdscp preferred clearinghouse(8cds)**, **show cdscp preferred clearinghouse(8cds)**, **show clearinghouse(8cds)**

create child(8cds)

create child

Purpose Creates a child pointer at the master replica of the parent directory

Synopsis `cdscp create child child-name clearinghouse clearinghouse-name`

Arguments

child-name The full name of the child pointer.

clearinghouse-name

The full name of a clearinghouse that contains a replica of the child directory.

Description

The **create child** command creates a child pointer at the master replica of the parent directory. When CDS looks up a name in the namespace, it uses child pointers to locate directory replicas.

Privilege Required

You must have insert permission to the parent directory.

Notes Use the **create child** command only to re-create a child pointer that is accidentally deleted. This command is designed only for troubleshooting.

Examples The following command creates the child pointer in the parent directory `./:/subsys`. It uses the replica located at the `./:/subsys/NY_CH` clearinghouse to fill in its replica set.

```
cdscp> create child ./:/subsys clearinghouse ./:/subsys/NY_CH
```

Related Information

Commands: **delete child(8cds)**, **list child(8cds)**, **show child(8cds)**

create clearinghouse

Purpose Creates a clearinghouse on the local server system or makes an existing clearinghouse available

Synopsis `cdscp create clearinghouse clearinghouse-name`

Arguments

clearinghouse-name

The full name of the clearinghouse.

Description

The **create clearinghouse** command creates a clearinghouse on the local server system or makes an existing clearinghouse available. The server start-up command usually creates a new clearinghouse when you configure a new CDS server. Occasionally, you may need to create a second clearinghouse on a particular server; for example, if you are temporarily relocating a clearinghouse on a different server. See the *OSF DCE Administration Guide* for more information about relocating a clearinghouse.

Clearinghouses should be named only in the root. When you enter the command, CDS creates a read-only replica of the root directory and stores it in the new clearinghouse as the initial replica. Because the process that creates the new clearinghouse initiates a skulk of the root directory, all replicas of the root should be reachable when you enter the command.

Privilege Required

You need write permission to the server on which you intend to create the clearinghouse and administer permission to the cell root directory. The server principal needs read, write, and administer permission to the cell root directory.

Notes This command is usually executed only by the network configuration procedure. To ensure that all replicas of the root are reachable, perform an immediate skulk of `./:` prior to issuing this command.

Examples The following command creates a clearinghouse named `./:/Boston_CH` on the local server system:

```
cdscp> create clearinghouse ./:/Boston_CH
```

create clearinghouse(8cds)

Related Information

Commands: **clear clearinghouse(8cds)**, **delete clearinghouse(8cds)**, **list clearinghouse(8cds)**, **set cdscp preferred clearinghouse(8cds)**, **show cached clearinghouse(8cds)**, **show cdscp preferred clearinghouse(8cds)**, **show clearinghouse(8cds)**

Books: *OSF DCE Administration Guide*

create directory

Purpose Creates a directory

Synopsis `cdscp create directory directory-name [clearinghouse clearinghouse-name]`

Arguments

directory-name

The full name of the directory

clearinghouse-name

The name of the clearinghouse in which you create the directory.

Description

The **create directory** command creates a directory with the name that you specify. If you do not specify a clearinghouse, CDS creates the master replica of the directory in the same clearinghouse as the new directory's parent directory.

Privilege Required

You must have insert permission to the parent directory, and write permission to the clearinghouse that stores the master replica of the new directory. The server principal requires read and insert permission to the parent directory of the new directory.

Notes To ensure that all replicas are consistent, perform an immediate skulk of the parent directory after issuing this command.

Examples The following command creates a directory named `./:/sales`.

```
cdscp> create directory ./:/sales
```

Related Information

Commands: **delete directory(8cds)**, **list directory(8cds)**, **set directory(8cds)**, **set directory to skulk(8cds)**, **show directory(8cds)**

create link(8cdfs)

create link

Purpose Creates a soft link and optionally specifies an expiration time and an extension time

Synopsis `cdscp create link link-name CDS_LinkTarget = target-name
[CDS_LinkTimeout = (expiration-time extension-time)]`

Arguments

link-name The full name of the soft link.

target-name The full name of the entry to which the soft link points.

expiration-time

A date and time after which CDS checks for existence of the soft link's target and either extends or deletes the soft link. The value is specified as `yyyy-mm-dd-hh:mm:ss` (year-month-day-hour:minute:second).

extension-time

A period of time by which to extend the soft link's expiration time (if the server has validated that the target still exists). The value is specified as `ddd-hh:mm:ss` (days-hour:minute:second).

Description

The **create link** command creates a soft link. If you specify the **CDS_LinkTimeout** attribute, you must specify an expiration time and an extension time. If you omit the **CDS_LinkTimeout** attribute, the soft link is permanent and must be explicitly deleted.

Privilege Required

You must have insert permission to the parent directory.

Examples The following command creates a permanent soft link named `./:/sales/tokyo/price-server` that points to an object entry named `./:/sales/east/price-server`.

```
cdscp> create link ./:/sales/tokyo/price-server CDS_LinkTarget \  
> = ./:/sales/east/price-server
```

create link(8cds)

Related Information

Commands: **delete link(8cds)**, **list link(8cds)**, **set link(8cds)**, **show link(8cds)**

create object(8cds)

create object

Purpose Creates an object entry

Synopsis `cdscp create object object-name [CDS_Class = class-name
[CDS_ClassVersion = value]]`

Arguments

object-name The full name of the object entry.

class-name The class of object entry being created. You can specify an application-defined class name. A class is specified as a simple name limited to 31 characters.

value The version of the class assigned to the object entry. Specify the value as *v.n*, where *v* defines the major release number and *n* specifies the minor version number. Specifying a class version is not required but is useful for allowing the definition of a class to evolve as the application is revised.

Description

The **create object** command creates an object entry. This task is usually done through a client application.

Privilege Required

You must have insert permission to the parent directory.

Examples The following command creates an object entry named `././sales/east/floor1cp`. The object entry describes a color printer on the first floor of a company's eastern sales office.

```
cdscp> create object ././sales/east/floor1cp CDS_Class = printer CDS_ClassVersion = 1.0
```

Related Information

Commands: **delete object(8cds)**, **list object(8cds)**, **set object(8cds)**, **show object(8cds)**

create replica

Purpose Creates a replica of an existing directory in the specified clearinghouse

Synopsis `cdscp create replica directory-name clearinghouse clearinghouse-name`

Arguments

directory-name

The full name of the directory.

clearinghouse-name

The full name of the clearinghouse in which you want to create the replica.

Description

The **create replica** command creates a replica of an existing directory in the specified clearinghouse.

Privilege Required

You must have administer permission to the directory you intend to replicate and write permission to the clearinghouse that stores the new replica. The server principal needs read, write, and administer permission to the directory you intend to replicate.

Notes This command is usually executed only by the network configuration procedure. To ensure that all replicas are consistent, perform an immediate skulk of the parent directory after issuing this command.

Examples The following command creates a replica of the `./:/mfg` directory in the clearinghouse `./:/Paris_CH`.

```
cdscp> create replica ./:/mfg clearinghouse ./:/Paris1_CH
```

Related Information

Commands: **delete replica(8cds)**, **show replica(8cds)**

define cached server(8cads)

define cached server

Purpose Creates knowledge of a server in the local clerk's cache

Synopsis `cdscp define cached server name tower value`

Arguments

<i>name</i>	A simple name for the cached server.
<i>value</i>	The protocol sequence and network address of the server node. The format is <i>protocol-sequence:network-address</i> . A <i>protocol-sequence</i> is a character string identifying the network protocols used to establish a relationship between a client and server. There are two choices of protocol sequence, depending on the network address that is supplied in the binding: ncacn_ip_tcp or ncadg_ip_udp . For the <i>network-address</i> , specify an Internet address using the common Internet address notation. For more information about this format, see the RPC introduction in the <i>OSF DCE Application Development Reference</i> .

Description

The **define cached server** command creates knowledge of a server in the local clerk's cache. This command is typically used to manually provide configuration information to a clerk that cannot automatically configure itself. This is required, for instance, to give the clerk addressing information about a server across a WAN. Once the clerk knows about one server, it can find other servers through referrals.

Privilege Required

You must have write permission to the clerk.

Examples The following command creates knowledge of the server **nrl** in the local clerk's cache:

```
cdscp> define cached server nrl tower ncacn_ip_tcp:16.20.15.25
```

Related Information

Commands: **clear cached server(8cds)**, **dump clerk cache(8cds)**, **show cached server(8cds)**

Books: *OSF DCE Application Development Reference*

delete child(8cds)

delete child

Purpose Deletes a child pointer from the namespace

Synopsis `cdscp delete child child-name`

Arguments

child-name The full name of the child pointer.

Description

The **delete child** command deletes a child pointer from the namespace.

Privilege Required

You must have delete permission to the child pointer or administer permission to the parent directory.

Notes Use the **delete child** command only when the directory to which the child pointer refers is deleted and the child pointer accidentally remains.

Examples The following command deletes the child pointer that accidentally remains after the `./:/sales/east` directory is deleted:

```
cdscp> delete child ./:/sales/east
```

Related Information

Commands: **create child(8cds)**, **list child(8cds)**, **show child(8cds)**

delete clearinghouse

Purpose Deletes the specified clearinghouse from the local server system

Synopsis `cdscp delete clearinghouse clearinghouse-name`

Arguments

clearinghouse-name

The full name of the clearinghouse.

Description

The **delete clearinghouse** command deletes a clearinghouse from the local server system. This command also automatically deletes all read-only replicas from a clearinghouse. CDS does not permit you to delete a clearinghouse that contains a master replica. See the *OSF DCE Administration Guide* for more information about handling master replicas when deleting a clearinghouse.

Privilege Required

You must have write and delete permission to the clearinghouse and administer permission to all directories that store replicas in the clearinghouse. The server principal needs delete permission to the associated clearinghouse object entry and administer permission to all directories that store replicas in the clearinghouse.

Notes It is recommended that you delete all replicas except the root before issuing this command.

Examples The following command deletes a clearinghouse named `./:/sales/Orion_CH` from the local server system:

```
cdscp> delete clearinghouse ./:/sales/Orion_CH
```

delete clearinghouse(8cds)

Related Information

Commands: **clear clearinghouse(8cds)**, **create clearinghouse(8cds)**, **list clearinghouse(8cds)**, **set cdscp preferred clearinghouse(8cds)**, **show clearinghouse(8cds)**, **show cdscp preferred clearinghouse(8cds)**

Books: *OSF DCE Administration Guide*

delete directory

Purpose Deletes a directory

Synopsis `cdscp delete directory directory-name`

Arguments

directory-name

The full name of the directory.

Description

The **delete directory** command deletes a directory. The directory cannot contain any object entries, soft links, or child pointers. The master replica must be the only remaining replica in the namespace. Use the **delete replica** command if you need to remove read-only replicas.

Privilege Required

You must have delete permission to the directory and write permission to the clearinghouse that stores the master replica of the directory. The server principal needs administer permission to the parent directory or delete permission to the child pointer that points to the directory you intend to delete.

Examples The following command deletes the directory `./eng` from the namespace:

```
cdscp> delete directory ./eng
```

Related Information

Commands: **create directory(8cds)**, **delete replica(8cds)**, **list directory(8cds)**, **set directory(8cds)**, **set directory to skulk(8cds)**, **show directory(8cds)**

delete link(8cds)

delete link

Purpose Deletes a soft link

Synopsis `cdscp delete link link-name`

Arguments

link-name The full name of the soft link.

Description

The **delete link** command deletes a soft link.

Privilege Required

You must have delete permission to the soft link, or administer permission to the directory that stores the soft link.

Examples The following command deletes the soft link `./:/sales/asia`.

```
cdscp> delete link ./:/sales/asia
```

Related Information

Commands: **create link(8cds)**, **list link(8cds)**, **set link(8cds)**, **show link(8cds)**

delete object

Purpose Deletes an object entry

Synopsis `cdscp delete object object-name`

Arguments

object-name The full name of the object entry.

Description

The **delete object** command deletes an object entry. This task is usually done through the client application, except under certain circumstances (for example, if the application is obsolete or no longer has access to the namespace).

Privilege Required

You must have delete permission to the object entry, or administer permission to the directory that stores the object entry.

Examples The following command deletes the object entry `./:/sales/east/floor1pr2`.

```
cdscp> delete object ./:/sales/east/floor1pr2
```

Related Information

Commands: **create object(8cds)**, **list object(8cds)**, **set object(8cds)**, **show object(8cds)**

delete replica(8cds)

delete replica

Purpose Deletes a read-only replica of a directory from a clearinghouse

Synopsis **cdscp delete replica** *directory-name* **clearinghouse** *clearinghouse-name*

Arguments

directory-name

The full name of the directory.

clearinghouse-name

The full name of the clearinghouse.

Description

The **delete replica** command deletes a read-only replica of a directory from a clearinghouse. Use the **delete directory** command to delete the master replica of the directory.

Privilege Required

You must have administer permission to the directory whose replica you want to delete and write permission to the clearinghouse from which you are deleting the replica.

Examples The following command deletes a read-only replica of the `./:/mfg` directory from the `./:/Paris1_CH` clearinghouse:

```
cdscp> delete replica ./:/mfg clearinghouse ./:/Paris1_CH
```

Related Information

Commands: **create replica(8cds)**, **delete directory(8cds)**, **show replica(8cds)**

disable clerk

Purpose Stops the clerk on the local system

Synopsis `cdscp disable clerk`

Description

The **disable clerk** command stops the clerk on the local system, causing all active communication with any server to be aborted and all client calls in progress to fail. The clerk cache is copied to disk.

Privilege Required

You must have write permission to the clerk.

Notes If you are disabling a clerk on a system where a server is running, make sure you disable the server first.

Examples The following command stops the clerk on the local server system:

```
cdscp> disable clerk
```

Related Information

Commands: **show clerk(8cds)**

disable server(8cds)

disable server

Purpose Stops the server on the local system

Synopsis `cdscp disable server`

Description

The **disable server** command stops the server on the local system. The server is disabled after all transactions in progress are completed.

Privilege Required

You must have write permission to the server.

Examples The following command stops the server on the local system:

```
cdscp> disable server
```

Related Information

Commands: **show server(8cds)**

dump clerk cache

Purpose Displays the contents of the clerk cache

Synopsis `cdscp dump clerk cache`

Description

The **dump clerk cache** command displays the contents of the clerk cache on the screen. Use this command when solving CDS problems.

Privilege Required

You must have superuser (root) privileges on the clerk system. No CDS permissions are required.

Examples The following command displays the contents of the clerk cache on the screen:

```
cdscp> dump clerk cache(8cds)
```

Related Information

Commands: `show clerk(8cds)`

gdad

Purpose Starts the GDA daemon

Synopsis **gdad** [-v]

Options -v Prints initialization progress messages; these verify that initialization successfully completed.

Description

The **gdad** command starts the GDA daemon. The Global Directory Agent (GDA) enables intercell communication, serving as a connection to other cells through the global naming environment.

Privilege Required

You must log in as superuser (root).

Notes Use this command only when troubleshooting.

Examples To start the **gdad** process, follow these steps:

1. Make sure that a CDS server is already running somewhere within the cell.
2. Log in to the system as superuser (root).
3. Log in to DCE as the machine principal of the local host. Enter the principal name in the format **/hosts/hostname/self** as shown in the following example command for a host named **orion** whose password is **smith**.

```
# dce_login hosts/orion/self smith
```

4. Enter the following command to see if the **rpcd** process is already running:

```
# ps
```

If the **rpcd** process appears on the list of active processes, proceed to step 5. If the **rpcd** process does not appear on the list of active processes, enter the following command to start the process:

```
# rpcd
```

5. Enter the following command to start the **cadsadv** process:

```
# cadsadv
```

6. Enter the following command to start the **gdad** process:

```
# gdad
```

To stop the GDA, enter the following command:

```
# kill pid
```

where *pid* is the process identifier of the **gdad** process.

Related Information

Books: *OSF DCE Administration Guide*

list child(8cds)

list child

Purpose Displays a list of all the child pointers whose names match the specified child name

Synopsis `cdscp list child child-name [with attribute-name = attribute-value]`

Arguments

child-name The full name of a specific child pointer. The last simple name can contain wildcard characters.

attribute-name
The name of a particular attribute.

attribute-value
The value of a particular attribute.

Description

The **list child** command displays a list of all the child pointers whose names match the specified child name. The last simple name can contain wildcard characters. You can use a **with *attribute-name = attribute-value*** clause to limit output only to child pointers whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign).

Privilege Required

You must have read permission to the directory that stores the child pointer. If you use a **with *attribute-name = attribute-value*** clause in the command, you also need read or test permission to the selected child pointers.

Examples The following command displays a list of all the child pointers named in the `./:/sales` directory:

```
cdscp> list child ./:/sales/*
```

```
LIST
CHILD /.../abc.com/sales
AT 1991-10-15-15:56:00
Q1
Q2
Q3
Q4
```

Related Information

Commands: **create child(8cds)**, **delete child(8cds)**, **show child(8cds)**

list clearinghouse(8cds)

list clearinghouse

Purpose Displays a list of all the clearinghouses whose names match the specified clearinghouse name

Synopsis `cdscp list clearinghouse clearinghouse-name`
[with *attribute-name* = *attribute-value*]

Arguments

clearinghouse-name

The full name of a specific clearinghouse. The last simple name can contain wildcard characters.

attribute-name

The name of a particular attribute.

attribute-value

The value of a particular attribute.

Description

The **list clearinghouse** command displays a list of all the clearinghouses whose names match the specified name. The last simple name can contain wildcards. You can use a **with** *attribute-name* = *attribute-value* clause to limit output only to clearinghouses whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign).

Privilege Required

You must have read permission to the directory that stores the associated clearinghouse object entry. If you use a **with** *attribute-name* = *attribute-value* clause in the command, you also need read or test permission to the selected clearinghouses.

Examples The following command displays a list of all the clearinghouses named in the root directory:

```
cdscp> list clearinghouse ./:/*
```

```
LIST
CLEARINGHOUSE  /.../abc.com/*
                AT  1991-10-15-15:56:00
/.../abc.com/Munich_CH
/.../abc.com/Paris_CH
```

Related Information

Commands: **clear clearinghouse(8cds)**, **create clearinghouse(8cds)**, **delete clearinghouse(8cds)**, **set clearinghouse(8cds)**, **set cdscp preferred clearinghouse(8cds)**, **show cdscp preferred clearinghouse(8cds)**, **show clearinghouse(8cds)**

list directory(8cds)

list directory

Purpose Displays a list of all the directories whose names match the specified directory name

Synopsis `cdscp list directory directory-name [with attribute-name = attribute-value]`

Arguments

directory-name

The full name of a specific directory. The last simple name can contain wildcard characters.

attribute-name

The name of a particular attribute.

attribute-value

The value of a particular attribute.

Description

The **list directory** command displays a list of all the directories whose names match the specified directory name. The last simple name can contain wildcards. You can use a **with** *attribute-name* = *attribute-value* clause to limit output only to directories whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign).

Privilege Required

You must have read permission to the parent directory. If you use a **with** *attribute-name* = *attribute-value* clause in the command, you also need read or test permission to the selected directories.

Examples The following command displays the names of all the directories in the `./:/sales` directory:

```
cdscp> list directory ./:/sales/*
```

```
LIST
DIRECTORY  /.../abc.com/sales
AT         1991-10-15-15:43:58

atlanta
austin
boston
chicago
ontario
ny
seattle
```

Related Information

Commands: **add directory(8cds)**, **create directory(8cds)**, **delete directory(8cds)**, **remove directory(8cds)**, **set directory(8cds)**, **set directory to skulk(8cds)**, **show directory(8cds)**

list link(8cdfs)

list link

Purpose Displays a list of all the soft links whose names match the link name that you specify

Synopsis `cdscp list link link-name [with attribute-name = attribute-value]`

Arguments

link-name The full name of a specific soft link. The last simple name can contain wildcard characters.

attribute-name
The name of a particular attribute.

attribute-value
The value of a particular attribute.

Description

The **list link** command displays a list of all the soft links whose names match the link name that you specify. The last simple name can contain wildcard characters. You can use a **with attribute-name = attribute-value** clause to limit output only to soft links whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign).

Privilege Required

You must have read permission to the directory that stores the soft link. If you use a **with attribute-name = attribute-value** clause in the command, you also need read or test permission to the selected soft links.

Examples The following command displays a list of all the soft links whose names begin with the letter **l** in the directory **./admin**.

```
cdscp> list link ./admin/l*
```

```

                                LIST
                                SOFTLINK  /.../abc.com/admin
                                AT        1991-10-15-15:54:38
lnk01
lnk02
lnk03
lnk04
lnk05
lnk06
```

Related Information

Commands: **create link(8cds)**, **delete link(8cds)**, **remove link(8cds)**, **set link(8cds)**, **show link(8cds)**

list object(8cds)

list object

Purpose Lists the specifies object entries (including clearinghouse object entries)

Synopsis `cdscp list object object-name [with attribute-name = attribute-value]`

Arguments

object-name The full name of a specific object entry. The last simple name can contain wildcard characters.

attribute-name
The name of a particular attribute.

attribute-value
The value of a particular attribute.

Description

The **list object** command displays a list of all the object entries (including clearinghouse object entries) whose names match the object entry name that you specify. The last simple name can contain wildcard characters. You can use a **with** *attribute-name* = *attribute-value* clause to limit output only to object entries whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign).

Privilege Required

You must have read permission to the directory that stores the object entry. If you use a **with** *attribute-name* = *attribute-value* clause in the command, you also need read or test permission to the selected object entries.

Examples The following command displays a list of all the object entries named in the directory `./eng`.

```
cdscp> list object ./eng/*
```

```
                                LIST
OBJECT   /.../abc.com/eng
        AT   1991-10-15-15:53:06
juno
test_stats
work_disk1
work_disk2
```

Related Information

Commands: **add object(8cds)**, **create object(8cds)**, **delete object(8cds)**, **remove object(8cds)**, **set object(8cds)**, **show object(8cds)**

remove directory(8cnds)

remove directory

Purpose Removes a value from a set-valued or single-valued attribute (including application-defined attributes) of a directory

Synopsis `cdscp remove directory directory-name attribute-name [= attribute-value]`

Arguments

directory-name

The full name of the directory.

attribute-name

The name of a particular attribute. Specify only one attribute at a time. See the `cds_attributes` file for the list of attributes and corresponding data types that your application uses.

attribute-value

The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute.

Description

The **remove directory** command removes a value from a set-valued or single-valued attribute (including application-defined attributes) of a directory. If you do not specify a value, the command removes the entire attribute. This command can delete attributes created by the **add directory** and **set directory** commands. Usually this task is performed through the client application. See the *OSF DCE Administration Guide* for more information about attributes.

Privilege Required

You must have write permission to the directory.

Examples To remove the value **1** from the user-defined, set-valued attribute **dirregion** of a directory named `./:/sales`, follow these steps:

1. Read the `cds_attributes` file to check that the attribute **dirregion** is listed, as shown in the following display:

OID	LABEL	SYNTAX
1.3.22.1.3.66	dirregion	small

remove directory(8cds)

2. Enter the following command to remove the value **1** from the attribute **dirregion**.

```
cdscp> remove directory ./:/sales dirregion = 1
```

Related Information

Commands: **add directory(8cds)**, **list directory(8cds)**, **set directory(8cds)**, **set directory to skulk(8cds)**, **show directory(8cds)**

Books: *OSF DCE Administration Guide*

remove link(8cds)

remove link

Purpose Removes a soft link's timeout value attribute

Synopsis `cdscp remove link link-name CDS_LinkTimeout`

Arguments

link-name The full name of the soft link.

Description

The `remove link` command removes a soft link's timeout value attribute, `CDS_LinkTimeout`, causing the soft link to become permanent.

Privilege Required

You must have write permission to the soft link.

Examples The following command removes the timeout value attribute of a soft link named `./:/eng/link01`.

```
cdscp> remove link ./:/eng/link01 CDS_LinkTimeout
```

Related Information

Commands: `create link(8cds)`, `delete link(8cds)`, `list link(8cds)`, `set link(8cds)`, `show link(8cds)`

remove object

Purpose Removes a value from a set-valued or single-valued attribute (including application-defined attributes) of an object entry

Synopsis **cdscp remove object** *object-name attribute-name* [= *attribute-value*]

Arguments

object name The full name of the object entry.

attribute-name

The name of a particular attribute. Specify only one attribute at a time. See the **cds_attributes** file for the list of attributes and corresponding data types that your application uses.

attribute-value

The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute.

Description

The **remove object** command removes a value from a set-valued or single-valued attribute (including application-defined attributes) of an object entry. If you do not specify a value, the command removes the entire attribute. This command can delete attributes created by the **add object** and **set object** commands. Usually, this task is performed through the client application. See the *OSF DCE Administration Guide* for more information about attributes.

Privilege Required

You must have write permission to the object entry.

Examples To remove the value **ps** from the attribute **printcap** of an object entry named **./:/mlh/deskprinter**, follow these steps:

1. Read the **cds_attributes** file to check that the attribute **printcap** is listed, as shown in the following display:

OID	LABEL	SYNTAX
1.3.22.1.3.50	printcap	char

remove object(8cds)

2. Enter the following command to remove the value **ps** from the attribute **printcap**.

```
cdscp> remove object ./mlh/deskprinter printcap = ps
```

Related Information

Commands: **add object(8cds)**, **list object(8cds)**, **set object(8cds)**, **show object(8cds)**

Books: *OSF DCE Administration Guide*

set cdscp confidence

Purpose Sets the confidence level of clerk calls issued as a result of CDS control program commands

Synopsis `cdscp set cdscp confidence = value`

Arguments

value One of the following confidence levels: **low**, **medium**, or **high**. A low confidence level means the clerk obtains information from caches or the most convenient server. A medium level means the clerk obtains information directly from a server. A high level means the clerk obtains information only at master replicas. The initial value is **medium**.

Description

The **set cdscp confidence** command sets the confidence level of clerk calls issued as a result of CDS control program commands. You must use this command within the CDS control program. Exiting from the CDS control program removes the confidence level setting. You must reset the confidence level each time you enter the CDS control program.

Examples The following command sets the confidence level of clerk calls to **high**.

```
$ cdscp  
cdscp> set cdscp confidence = high
```

Related Information

Commands: **show cdscp confidence(8cds)**

set cdscp preferred clearinghouse(8cds)

set cdscp preferred clearinghouse

Purpose Specifies a preferred clearinghouse to use for satisfying read requests that result from CDS control program commands

Synopsis `cdscp set cdscp preferred clearinghouse [clearinghouse-name]`

Arguments

clearinghouse-name

The full name of the preferred clearinghouse. If you omit this argument, the command causes CDS to revert to the default, which is to use any clearinghouse.

Description

The `set cdscp preferred clearinghouse` command specifies a preferred clearinghouse to use for satisfying read requests that result from CDS control program commands. You cannot specify a preferred clearinghouse for making modifications, because these requests always use the master replica. You must use this command within the CDS control program. Exiting from the CDS control program removes the preferred clearinghouse setting. You must reset the preferred clearinghouse each time you enter the CDS control program.

Examples The following command specifies `./:Paris_CH` as the preferred clearinghouse:

```
$ cdscp
cdscp> set cdscp preferred clearinghouse ./:Paris_CH
```

Related Information

Commands: `show cdscp preferred clearinghouse(8cds)`

set directory

Purpose Changes the value of a modifiable, single-valued attribute of a directory

Synopsis `cdscp set directory directory-name attribute-name = attribute-value`

Arguments

directory-name

The full name of the directory.

attribute-name

The name of a particular attribute. Specify only one attribute at a time. See the **cds_attributes** file for the list of attributes and corresponding data types that your application uses.

attribute-value

The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute.

Description

The **set directory** command changes the value of a modifiable, single-valued attribute of a directory. If the attribute does not exist, this command creates it. Usually, this task is performed through the client application. See the *OSF DCE Administration Guide* for more information about attributes. You can specify an application-defined attribute or the following attribute, which specifies the degree of consistency among replicas:

CDS_Convergence = *value*

Specify one of the following for *value*:

- | | |
|---------------|--|
| low | CDS does not immediately propagate any updates. The next skulk distributes all updates that occurred since the previous skulk. Skulks occur at least once every 24 hours. |
| medium | CDS attempts to immediately propagate an update to all replicas. If the attempt fails, the software lets the next scheduled skulk make the replicas consistent. Skulks occur at least once every 12 hours. |
| high | CDS attempts to immediately propagate an update to all replicas. If that attempt fails (for example, if one of the replicas is unavailable), a skulk is scheduled for within one hour. Background skulks occur |

set directory(8cds)

at least once every 12 hours. Use this setting temporarily and briefly because it uses extensive system resources.

By default, every directory inherits the convergence setting of its parent at creation time. The default setting on the root directory is **medium**.

Privilege Required

You must have write permission to the directory.

Examples The following command sets a low convergence value on the `./:/mfg` directory:

```
cdscp> set directory ./:/mfg CDS_Convergence = low
```

Related Information

Commands: **create directory(8cds)**, **delete directory(8cds)**, **list directory(8cds)**, **remove directory(8cds)**, **set directory to skulk(8cds)**, **show directory(8cds)**

Books: *OSF DCE Administration Guide*

set directory to new epoch

Purpose Reconstructs a directory's replica set, allowing you to designate a new master replica or to exclude a replica

Synopsis `cdscp set directory directory-name to new epoch master clearinghouse-name [readonly clearinghouse-name...] [exclude clearinghouse-name...]`

Arguments

directory-name
The full name of the directory.

clearinghouse-name
The full name of the clearinghouse in which an individual replica is located. The first *clearinghouse-name* specifies where the master replica is stored.

Description

The **set directory to new epoch** command reconstructs a directory's replica set, allowing you to designate a new master replica or to exclude a replica. You must list each existing replica and indicate whether an existing replica needs to be included in or excluded from the new replica set. You can include or exclude more than one replica. The ellipses (...) indicates that you can specify multiple clearinghouse names, separated by spaces.

Privilege Required

You must have administer permission to the directory, and the server principal needs administer, read, and write permission to the directory. When designating a new master replica, you also need write permission to the clearinghouse that stores the current master replica, and the server principal needs write permission to the clearinghouse that stores the read-only replica that you intend to designate as the new master replica.

Examples The following command sets a new epoch for the directory `./mfg`. The master replica is in the clearinghouse `./Paris1_CH`, and read-only replicas are in the clearinghouses `./Chicago1_CH`, `./Seattle_CH`, and `./NY1_CH`. The new replica set excludes the replica in the clearinghouse `./NY1_CH`.

set directory to new epoch(8cds)

```
cdscp> set directory ./:/mfg to new epoch master ./:/Paris1_CH\  
> readonly ./:/Chicago1_CH ./:/Seattle_CH exclude ./:/NY1_CH
```

Related Information

Commands: **set directory to skulk(8cds)**, **show directory(8cds)**, **show replica(8cds)**

set directory to skulk

Purpose Starts the skulk of a directory immediately

Synopsis `cdscp set directory directory-name to skulk`

Arguments

directory-name

The full name of the directory.

Description

The **set directory to skulk** command starts the skulk of a directory immediately. The CDS control program prompt `cdscp>` does not return until the skulk is complete. The amount of time for the skulk to complete is dependent on the location, number, and availability of replicas of the directory.

Privilege Required

You must have administer, write, insert, or delete permission to the directory. The server principal needs administer, read, and write permission to the directory.

Examples The following command initiates a skulk on the `./:/admin` directory:

```
cdscp> set directory ./:/admin to skulk
```

Related Information

Commands: **add directory(8cds)**, **create directory(8cds)**, **delete directory(8cds)**, **list directory(8cds)**, **remove directory(8cds)**, **set directory to new epoch(8cds)**, **show directory(8cds)**

set link(8cde)

set link

Purpose Changes the value of a modifiable, single-valued attribute of a soft link

Synopsis `cdscp set link link-name attribute-name = attribute-value`

Arguments

link-name The full name of the soft link.

attribute-name

The name of the attribute to be modified. Specify only one attribute at a time. Valid attribute names are described later in this reference page.

attribute-value

The value of a particular attribute.

Description

The `set link` command changes the value of a modifiable, single-valued attribute of a soft link. The following are valid attributes:

CDS_LinkTarget = *fullname*

Specifies the full name of the directory, object entry, or other soft link to which the soft link points.

CDS_LinkTimeout = (*expiration-time extension-time*)

Specifies a timeout value after which the soft link is either checked or deleted. The timeout value contains both an expiration time and an extension time. If a soft link expires and its target entry is deleted, the soft link is deleted. If the soft link still points to an existing entry, its life is extended by the expiration time. Specify *expiration-time* in the format *yyyy-mm-dd-hh:mm:ss* (year-month-day-hour:minute:second). Specify *extension-time* in the format *ddd-hh:mm:ss* (day-hour:minute:second).

Privilege Required

You must have write permission to the soft link.

Examples The following command redirects a soft link named `./admin/work_disk` from its current destination name, `./admin/work_disk01`, to a new destination name, `./admin/work_disk03`.

```
cdscp> set link ./admin/work_disk CDS_LinkTarget = ./admin/work_disk03
```

Related Information

Commands: `create link(8cds)`, `delete link(8cds)`, `list link(8cds)`, `show link(8cds)`

set object(8cds)

set object

Purpose Changes the value of a modifiable, single-valued attribute of an object entry

Synopsis `cdscp set object object-name attribute-name = attribute-value`

Arguments

object-name The full name of the object entry.

attribute-name

The name of the attribute to be modified. Specify only one attribute at a time. See the `cds_attributes` file for the list of attributes and corresponding data types that your application uses.

attribute-value

The value of a particular attribute. The value of an application-defined attribute is dependent on the type of attribute.

Description

The `set object` command changes the value of a modifiable, single-valued attribute of an object entry. If the attribute does not exist, this command creates it. Usually, this task is performed through the client application. See the *OSF DCE Administration Guide* for more information about attributes.

Privilege Required

You must have write permission to the object entry.

Examples To change the value of the `sales_record` attribute to `region2` of an object entry named `./:Q1_records`, follow these steps:

1. Read the `cds_attributes` file to check that the attribute `sales_record` is listed, as shown in the following display:

OID	LABEL	SYNTAX
1.3.22.1.3.66	sales_record	char

2. Enter the following command to assign the value `region2` to the attribute `sales_record` of an object entry named `./:Q1_records`.

```
cdscp> set object ./:Q1_records sales_record = region2
```

Related Information

Commands: **add object(8cds)**, **create object(8cds)**, **delete object(8cds)**, **list object(8cds)**, **remove object(8cds)**, **show object(8cds)**

Books: *OSF DCE Administration Guide*

show cached clearinghouse(8cds)

show cached clearinghouse

Purpose Displays current information about the specified cached clearinghouse

Synopsis `cdscp show cached clearinghouse clearinghouse-name`

Arguments

clearinghouse-name

A specific clearinghouse name. The name can contain wildcard characters.

Description

The **show cached clearinghouse** command displays all the names and values of the attributes in the specified cached clearinghouse. The following are valid attributes:

Creation Time

Specifies the time at which this clearinghouse was added to the cache

Miscellaneous Operations

Specifies the number of operations other than read and write (that is, skulks, new epochs, and so on) performed by this clerk on the cached clearinghouse

Read Operations

Specifies the number of lookup operations of any sort performed by the clerk on the cached clearinghouse

Towers

Specifies the protocol sequence and Internet address of the server that maintains the cached clearinghouse

Write Operations

Specifies the number of write operations performed by this clerk on the cached clearinghouse

Privilege Required

You must have read permission to the clerk.

show cached clearinghouse(8cds)

Examples The following command displays all attributes of the cached clearinghouse **./:/Paris2_CH**.

```
cdscp> show cached clearinghouse ./:/Paris2_CH
```

```
                SHOW
  CACHED CLEARINGHOUSE  /.../abc.com/Paris2_CH
                        AT  1991-10-15-15:58:09
      Creation Time = 1991-10-01-17:03:32.32
      Read Operations = 412
      Write Operations = 618
  Miscellaneous Operations = 278
```

Related Information

Commands: **list clearinghouse(8cds)**

show cached server(8cda)

show cached server

Purpose Displays address information of a server in the local clerk's cache

Synopsis `show cached server name`

Arguments

name A simple name for the cached server. The name can contain wildcard characters.

Description

The **show cached server** command displays address information of a server in the local clerk's cache. The following list describes the valid attributes:

Name The directory cell name

Towers The protocol sequence and network address of the server node

Privilege Required

You must have read permission to the clerk.

Examples The following command displays all attributes of the cached server **emv**.

```
cdscp> show cached server emv*
```

```
          SHOW
  CACHED NAMESERVER  emv_udp
                   AT  1991-10-15-15:56:56
                   Name = /.../emv.abc.com
                   Tower = ncadg_ip_udp:14.20.14.32
                   Tower = ncacn_ip_tcp:14.20.14.32
          SHOW
  CACHED NAMESERVER  emv_tcp
                   AT  1991-10-15-15:56:57
                   Name = /.../emv.abc.com
                   Tower = ncadg_ip_udp:14.20.14.32
                   Tower = ncacn_ip_tcp:14.20.14.32
```

Related Information

Commands: **clear cached server(8cds)**, **define cached server(8cds)**

show cdscp confidence(8cds)

show cdscp confidence

Purpose Displays the current confidence level of clerk calls resulting from CDS control program commands

Synopsis `cdscp show cdscp confidence`

Description

The `show cdscp confidence` command displays the current confidence level of clerk calls. A low confidence level means the clerk obtains information from caches or the most convenient server. A medium level means the clerk obtains information directly from a server. A high level means the clerk obtains information only at master replicas.

You must use this command within the CDS control program. Exiting from the CDS control program removes the confidence level setting. You must reset the confidence level each time you enter the CDS control program.

Examples The following command displays the current confidence level of clerk calls:

```
$ cdscp
cdscp> show cdscp confidence

Confidence used is medium
```

Related Information

Commands: `set cdscp confidence(8cds)`

show cdscp preferred clearinghouse

Purpose Displays the preferred clearinghouse for satisfying read requests that result from CDS control program commands

Synopsis `cdscp show cdscp preferred clearinghouse`

Description

The **show cdscp preferred clearinghouse** command displays the preferred clearinghouse for satisfying read requests that result from CDS control program commands. You can only read attribute values for entries stored in the specified clearinghouse.

You must use this command within the CDS control program. Exiting from the CDS control program removes the preferred clearinghouse setting. You must reset the preferred clearinghouse each time you enter the CDS control program.

Examples The following command displays the current clearinghouse:

```
$ cdscp
cdscp> show cdscp preferred clearinghouse

read attribute values from clearinghouse /.../abc.com/Paris_CH
```

Related Information

Commands: **set cdscp preferred clearinghouse(8cds)**

show cell(8cds)

show cell

Purpose Displays the information you need to create a cell entry in either DNS or GDS

Synopsis `cdscp show cell [cell-name] [as type]`

Arguments

cell-name The global name of the cell.

type The global namespace in which you want to define the cell. Enter either of the values **dns** or **gds**.

Description

The **show cell** command displays the information you need to create a cell entry in either the Domain Name System (DNS) or the Global Directory Service (GDS). DCE does not support cells registered simultaneously in GDS and DNS. If you want to define a cell in DNS, you can use this command to produce a preformatted set of resource records. You can then edit the appropriate DNS data file and copy the output directly into the file. In GDS, cell information is contained in two attributes: **CDS-Cell** and **CDS-Replica**. If you want to define a cell in GDS, you can use this command to obtain the data you need to supply when creating the **CDS-Cell** and **CDS-Replica** attributes. For details, see the *OSF DCE Administration Guide*.

Privilege Required

You must have read permission to the cell root directory.

Examples The following command displays the GDS-formatted output in the local cell:

```
cdscp> show cell /.: as gds
```

```
          SHOW
          CELL   /.../abc.com
          AT     1991-10-15-15:58:25
          Namespace Uuid = 2d2d50ad-8b1a-11ba-8983-08002b0f79aa
          Clearinghouse Uuid = 2ab024a8-8b1a-11ba-8983-08002b0f79aa
          Clearinghouse Name = /.../abc.com/NY_CH
          Replica Type = Master
          Tower 1 = ncadg_ip_udp:16.18.17.33
```

show cell(8cds)

Tower 2 = ncacn_ip_tcp:16.18.17.33

Namespace Uuid = 2d2d50ad-8b1a-11ba-8983-08002b0f79aa

Clearinghouse Uuid = 49757f28-8b1a-11ba-8983-08002b0f79aa

Clearinghouse Name = ../../abc.com/Boston_CH

Replica Type = Readonly

Tower 1 = ncadg_ip_udp:16.18.17.33

Tower 2 = ncacn_ip_tcp:16.18.17.33

Related Information

Books: *OSF DCE Administration Guide*

show child(8cdfs)

show child

Purpose Displays attribute information about the specified child pointer

Synopsis **cdscp show child** *child-name* [*attribute-name*]
 [**with** *attribute-name* = *attribute-value*]

Arguments

child-name The full name of a specific child pointer. The last simple name can contain wildcard characters.

attribute-name The name of an attribute. Valid attribute names are described later in this reference page.

attribute-value The value of a particular attribute.

Description

The **show child** command displays the names and values of the attributes specified in *attribute-name*. You can use a combination of attributes in a single command. Use a space to separate multiple attributes. You can use a **with** *attribute-name* = *attribute-value* clause to limit output only to child pointers whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign).

If you do not supply any attributes, the command displays all attributes and their values. The following is a description of child pointer attributes:

CDS_CTS Specifies the creation timestamp (CTS) of the specified child pointer.

CDS_ObjectUUID
 Specifies the unique identifier of the directory to which the child pointer refers.

CDS_Replicas
 Specifies the address, UUID, and name of a set of clearinghouses where a copy of the child directory referenced by the child pointer is located. This attribute also specifies whether the directory in a particular clearinghouse is a master or read-only replica.

CDS_UTS Specifies the timestamp of the most recent update to an attribute of the child pointer.

Privilege Required

You must have read permission to the child pointer. If you specify a wildcard child name, you also need read permission to the parent directory.

Examples The following command displays all of the attributes and values of the child directory to which the child pointer `./:/admin` refers:

```
cdscp> show child ./:/admin
```

```
      SHOW
      CHILD   /.../abc.com/admin
      AT      1991-10-15-15:56:01
      CDS_CTS = 1991-10-15-19:55:52.000000003/08-00-2b-1c-8f-1f
      CDS_UTS = 1991-10-15-19:55:52.000000006/08-00-2b-1c-8f-1f
      CDS_ObjectUUID = 6b5362e8-8b1c-11ca-8981-08002b0f79aa
      CDS_Replicas = :
      Clearinghouse's UUID = 2ab024a8-8b1a-11ca-8981-08002b0f79aa
      Tower = ncadg_ip_udp:16.18.16.32
      Tower = ncacn_ip_tcp:16.18.16.32
      Replica type = master
      Clearinghouse's Name = /.../abc.com/Boston_CH
```

Related Information

Commands: **create child(8cds)**, **delete child(8cds)**, **list child(8cds)**

show clearinghouse(8cds)

show clearinghouse

Purpose Displays attribute information about the specified clearinghouse

Synopsis `cdscp show clearinghouse clearinghouse-name [attribute-name]
[with attribute-name = attribute-value]`

Arguments

clearinghouse-name

The full name of a specific clearinghouse. The last simple name can contain wildcard characters.

attribute-name

The name of a particular attribute. Valid attribute names are described later in this reference page.

attribute-value

The value of a particular attribute.

Description

The **show clearinghouse** command displays the names and values of the attributes specified in *attribute-name*. You can use a combination of attributes in any sequence in a single command. Use a space to separate multiple attributes. You can use a **with** *attribute-name* = *attribute-value* clause to limit output only to clearinghouses whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign).

If you do not supply any attributes, the command displays all attributes and their values. The following list describes the clearinghouse attributes:

CDS_AllUpTo

Indicates the date and time the clearinghouse object has been updated to reflect the **CDS_CHDictionaries** attribute.

CDS_Cellname

Specifies the name of the cell in which the clearinghouse resides.

CDS_CHDictionaries

Specifies the full name and unique identifier (UUID) of every directory that has a replica in this clearinghouse.

CDS_CHLastAddress

Specifies the current reported network address of the clearinghouse.

CDS_CHName

Specifies the full name of the clearinghouse.

CDS_CHState

Specifies the state of the clearinghouse. The state *on* indicates the clearinghouse is running and available.

CDS_CTS Specifies the creation timestamp (CTS) of the clearinghouse.

CDS_DirectoryVersion

Specifies the current version of the directory in the clearinghouse in which the directory was created.

CDS_ObjectUUID

Specifies the unique identifier of the clearinghouse.

CDS_ReplicaVersion

Specifies the current version of the replica in which the directory was created.

CDS_UTS Specifies the timestamp of the most recent update to an attribute of the clearinghouse.

The following attributes and their values are displayed only when you use this command to display all attributes and their values:

Data Corruptions

Specifies the number of times that the *data corruption* event was generated

Enable Counts

Specifies the number of times that the clearinghouse was enabled since it was last started

Read Accesses

Specifies the number of read operations directed to this clearinghouse

References Returned

Specifies the number of requests directed to this clearinghouse that resulted in the return of a partial answer instead of satisfying the client's request

Skulk Failures

Specifies the number of times that a skulk of a directory, initiated

show clearinghouse(8cds)

from this clearinghouse, failed to complete— usually because one of the replicas in the replica set was unreachable

Times Clearinghouse Entry Missing

Specifies the number of times the *clearinghouse entry missing* event was generated

Times Root Not Reachable

Specifies the number of times the *root lost* event was generated

Upgrades Failed Counts

Specifies the number of times that upgrades failed

Write Accesses

Specifies the number of write operations directed to this clearinghouse

Privilege Required

You must have read permission to the clearinghouse. If you specify a wildcard clearinghouse name, you also need read permission to the cell root directory.

Examples The following command displays the current values of the **CDS_UTS** and **CDS_ObjectUUID** attributes associated with the **./:/Chicago1_CH** clearinghouse:

```
cdscp> show clearinghouse ./:/Chicago1_CH CDS_UTS CDS-ObjectUUID
```

```
          SHOW
CLEARINGHOUSE  /.../abc.com/Chicago1_CH
          AT    1991-10-21-13:12:30
          CDS_UTS = 1991-10-21-13:04:04.000000009/08-00-2b-1c-8f-1f
          CDS_ObjectUUID = 3706d70c-8b05-11ca-9002-08002b1c8f1f
```

Related Information

Commands: **clear clearinghouse(8cds)**, **create clearinghouse(8cds)**, **delete clearinghouse(8cds)**, **list clearinghouse(8cds)**, **set cdscp preferred clearinghouse(8cds)**, **show cdscp preferred clearinghouse(8cds)**

show clerk

Purpose Displays attribute information about the CDS clerk on the local system

Synopsis `cdscp show clerk`

Description

The **show clerk** command displays all the names and values of the clerk attributes on the local system. The following are valid attributes:

Authentication Failures

Specifies the number of times a requesting principal failed authentication procedures.

Cache Bypasses

Specifies the number of requests to read attributes for which the clerk was specifically directed by the requesting application to bypass its own cache. Instead, a server is contacted to get the requested information. This attribute does not account for requests that the clerk is unable to satisfy from the cache or for requests to look up names or enumerate the contents of directories.

Cache Hits Specifies the total number of read requests directed to this clerk that were satisfied entirely by the information contained in its own cache. This attribute accounts only for requests to read attribute values and does not include requests to look up names or enumerate the contents of directories.

Creation Time

Specifies the time when this entity was created.

Miscellaneous Operations

Specifies the number of operations other than read and write (that is, skulks, enumerating contents of directories, and so on) performed by this clerk.

Read Operations

Specifies the number of lookup operations performed by this clerk. This attribute accounts only for requests to read attributes and does not include requests to look up names or enumerate the contents of directories.

show clerk(8cds)

Write Operations

Specifies how many requests to modify data were processed by this clerk.

Privilege Required

You must have read permission to the clerk.

Examples The following command displays the attributes of the clerk on the local system:

```
cdscp> show clerk
```

```
          SHOW
          CLERK
          AT   1991-10-15-15:56:50
Creation Time = 1991-10-15-15:38:19.000000051-04:00I0.000000000
Authentication failures = 0
Read Operations = 1068
Cache Hits = 137
Cache bypasses = 433
Write operations = 1250
Miscellaneous operations = 590
```

Related Information

Commands: **disable clerk(8cds)**

show directory

Purpose Displays attribute information about the specified directory

Synopsis `cdsep show directory directory-name [attribute-name]`
[with *attribute-name* = *attribute-value*]

Arguments

directory-name

The full name of a specific directory. The last simple name can contain wildcard characters.

attribute-name

The name of a particular attribute. Valid attribute names are described later in this reference page.

attribute-value

The value of a particular attribute.

Description

The **show directory** command displays the names and values of the attributes specified in *attribute-name*. You can use a combination of attributes in any sequence in a single command. Use a space to separate multiple attributes. You can use a **with *attribute-name* = *attribute-value*** clause to limit output only to directories whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign). If you do not supply any attributes, the command displays all attributes and their values. In addition to the following directory attributes, application-specific attributes can exist for a directory:

CDS_AllUpTo

Indicates the date and time of the last successful skulk on the directory. All replicas of the directory are guaranteed to receive all updates whose timestamps are less than the value of this attribute.

CDS_Convergence

Specifies the degree of consistency among replicas. This attribute's value is defined as one of the following:

low CDS does not immediately propagate an update. The next skulk distributes all updates that occurred since

show directory(8cds)

the previous skulk. Skulks occur at least once every 24 hours.

medium CDS attempts to immediately propagate an update to all replicas. If the attempt fails, the next scheduled skulk makes the replicas consistent. Skulks occur at least once every 12 hours.

high CDS attempts to immediately propagate an update to all replicas. If the attempt fails (for example, if one of the replicas is unavailable), a skulk is scheduled for within one hour. Skulks usually occur at least once every 12 hours. Use this setting temporarily and briefly, because it uses extensive system resources.

By default, every directory inherits the convergence setting of its parent at creation time. The default setting on the root directory is **medium**.

CDS_CTS Specifies the creation timestamp (CTS) of the CDS directory.

CDS_DirectoryVersion

Specifies the current version of the directory (derived from the **CDS_DirectoryVersion** attribute of the clearinghouse in which the directory was created). Multiple directory versions are supported in a cell.

CDS_Epoch A UUID that identifies a particular incarnation of the directory.

CDS_InCHName

Indicates whether a directory or any of its descendants can store clearinghouse names. If this value is **true**, the directory can store clearinghouse names. If it is **false**, the directory cannot store clearinghouse names.

CDS_LastSkulk

Records the timestamp of the last skulk performed on this directory.

CDS_LastUpdate

Records the timestamp of the most recent change to any attribute of a directory replica, or any change to an entry in the replica.

CDS_ObjectUUID

Specifies the unique identifier of the directory.

CDS_ParentPointer

Contains a pointer to this directory's parent in the namespace.

CDS_Replicas

Specifies the address, UUID, and name of every clearinghouse where a copy of this directory is located. This attribute also specifies whether the replica in a particular clearinghouse is a master or read-only replica.

CDS_ReplicaState

Specifies whether a directory replica can be accessed.

CDS_ReplicaType

Indicates whether a directory replica is a master or read-only replica.

CDS_ReplicaVersion

Specifies the version of a replica of the directory.

CDS_RingPointer

Specifies the UUID of a clearinghouse containing another replica of this directory.

CDS_UTS

Specifies the timestamp of the most recent update to an attribute of the directory.

Privilege Required

You must have read permission to the directory. If you specify a wildcard directory name, you also need read permission to the directory's parent directory.

Examples The following command displays the current values of all the attributes associated with the `./admin` directory:

```
cdscp> show directory ./admin
```

```

      SHOW
  DIRECTORY  /.../abc.com/admin
      AT     1991-10-15-15:43:59
  CDS_CTS   = 1991-10-15-13:09:47.000000003/08-00-2b-1c-8f-1f
  CDS_UTS   = 1991-10-17-08:59:50.000000006/08-00-2b-1c-8f-1f
  CDS_ObjectUUID = ba700c98-8b1a-11ca-8981-08002b0f79aa
  CDS_Replicas = :
  Clearinghouse's UUID = 2ab024a8-8b1a-11ca-8981-08002b0f79aa
      Tower = ncadg_ip_udp:16.20.16.32
      Tower = ncacn_ip_tcp:16.20.16.32
  Replica type = master
  Clearinghouse's Name = /.../abc.com/Paris_CH
  CDS_AllUpTo = 1991-10-17-08:51:18.000000032/08-00-2b-1c-8f-1f

```

show directory(8cds)

```
CDS_Convergence = medium
CDS_ParentPointer = :
  Parent's UUID = b773525c-8b1a-11ca-8981-08002b0f79aa
  Timeout = :
  Expiration = 1991-10-16-19:43:50.516
  Extension = +1-00:00:00.000
CDS_DirectoryVersion = 3.0
CDS_ReplicaState = on
CDS_ReplicaType = master
  CDS_LastSkulk = 1991-10-17-08:51:18.000000032/08-00-2b-1c-8f-1f
  CDS_LastUpdate = 1991-10-21-13:04:02.000000044/08-00-2b-1c-8f-1f
  CDS_RingPointer = 2ab024a8-8b1a-11ca-8981-08002b0f79aa
  CDS_Epoch = bd8b2c50-8b1a-11ca-8981-08002b0f79aa
CDS_ReplicaVersion = 3.0
```

Related Information

Commands: **add directory(8cds)**, **create directory(8cds)**, **delete directory(8cds)**, **list directory(8cds)**, **remove directory(8cds)**, **set directory(8cds)**

show link

Purpose Displays attribute information about the specified soft link

Synopsis **cdscp show link** *link-name* [*attribute-name*]
[with *attribute-name* = *attribute-value*]

Arguments

link-name The full name of a specific soft link. The last simple name can contain wildcard characters.

attribute-name The name of a particular attribute. Valid attribute names are described later in this reference page.

attribute-value The value of a particular attribute.

Description

The **show link** command displays the names and values of the attributes specified in *attribute-name*. You can use a combination of attributes in any sequence in a single command. Use a space to separate multiple attributes. You can use a **with** *attribute-name* = *attribute-value* clause to limit output only to soft links whose attributes have values equal to the specified values. A space must precede and follow the = (equals sign). If you do not supply any attributes, the command displays all attributes and their values. The following is a description of soft link attributes:

CDS_CTS Specifies the creation timestamp (CTS) of this soft link

CDS_LinkTarget Specifies the full name of the directory, object entry, or other soft link to which the soft link points

CDS_LinkTimeout Specifies a timeout value after which the soft link is either checked or deleted

CDS_UTS Specifies the timestamp of the most recent update to an attribute of the soft link

show link(8cds)**Privilege Required**

You must have read permission to the soft link. If you specify a wildcard soft link name, you also need read permission to the directory that stores the soft link.

Examples The following command displays the current values of all the attributes associated with the soft link `././sales/region1`.

```
cdscp> show link ././sales/region1
```

```
      SHOW
SOFTLINK  /.../abc.com/sales/region1
      AT   1991-10-15-15:54:40
      CDS_CTS = 1991-10-15-19:54:35.00000003/08-00-2b-1c-8f-1f
      CDS_UTS = 1991-10-15-19:54:35.00000006/08-00-2b-1c-8f-1f
CDS_LinkTarget = /.../abc.com/sales/service
```

```
      SHOW
SOFTLINK  /.../abc.com/sales/region1
      AT   1991-10-15-15:54:41
      CDS_CTS = 1991-10-15-19:54:36.00000077/08-00-2b-1c-8f-1f
      CDS_UTS = 1991-10-15-19:54:36.00000009/08-00-2b-1c-8f-1f
CDS_LinkTarget = /.../abc.com/sales/software
CDS_LinkTimeout = :
      Expiration = 1991-10-15-00:00:00.0
      Extension = +1-00:00:00.000
```

Related Information

Commands: **create link(8cds)**, **delete link(8cds)**, **list link(8cds)**, **remove link(8cds)**, **set link(8cds)**

show object

Purpose Displays attribute information about the specified object entry

Synopsis `cdscp show object object-name [attribute-name]
[with attribute-name = attribute-value]`

Arguments

object-name The full name of a specific object entry. The last simple name can contain wildcard characters.

attribute-name The name of a particular attribute. Valid attribute names are described later in this reference page.

attribute-value The value of a particular attribute.

Description

The **show object** command displays the names and values of the attributes specified in *attribute-name*. You can use a combination of attributes in a single command. Use a space to separate multiple attributes. You can use a **with** *attribute-name* = *attribute-value* clause to limit output only to object entries whose attributes have values equal to the specified values. If you do not supply any attributes, the command displays all attributes and their values. In addition to the following attributes, any application-defined attributes that might exist will be included in the output of this command. The following is a description of object entry attributes:

CDS_Class Specifies the class to which an object belongs.

CDS_ClassVersion

Contains the version number of the object's class. This allows applications to build in compatibility with entries created by earlier versions.

CDS_CTS Specifies the creation timestamp (CTS) of this object entry.

CDS_ObjectUUID

Specifies a unique identifier for the object being referenced.

show object(8cds)

CDS_UTS Specifies the timestamp of the most recent update to an attribute of the object entry.

Privilege Required

You must have read permission to the object entry. If you specify a wildcard object entry name, you also need read permission to the directory that stores the object entry.

Examples The following command lists all the attributes and their values of the object entry `./:/sales/east/floor1cp`.

```
cdscp> show object ./:/sales/east/floor1cp
```

```
      SHOW
OBJECT  /.../abc.com/sales/floor1cp
      AT   1991-10-15-15:53:07
CDS_CTS = 1991-10-15-19:53:03.00000003/08-00-2b-1c-8f-1f
CDS_UTS = 1991-10-15-19:53:03.00000006/08-00-2b-1c-8f-1f
```

Related Information

Commands: **add object(8cds)**, **create object(8cds)**, **delete object(8cds)**, **list object(8cds)**, **remove object(8cds)**, **set object(8cds)**

show replica

Purpose Displays attribute information about the specified replica

Synopsis **cdsep show replica** *directory-name* **clearinghouse** *clearinghouse-name*
 [*attribute-name*]

Arguments

directory-name

The full name of the directory.

clearinghouse-name

The full name of the clearinghouse.

attribute-name

The name of a particular attribute. Valid attribute names are described later in this reference page.

Description

The **show replica** command displays the directory-specific attributes as well as the per-replica attributes of the specified directory. If you do not supply any attributes, the command displays all attributes and their values. You can enter one or more of the following attributes:

CDS_AllUpTo

Indicates the date and time of the last successful skulk on the directory. All replicas of the directory are guaranteed to have received all updates whose timestamps are less than the value of this attribute.

CDS_Convergence

Specifies the degree of consistency among replicas. This attribute's value is defined as one of the following:

low CDS does not immediately propagate an update. The next skulk distributes all updates that occurred since the previous skulk. Skulks occur at least once every 24 hours.

medium CDS attempts to immediately propagate an update to all replicas. If the attempt fails, the next scheduled

show replica(8cdfs)

skulk makes the replicas consistent. Skulks occur at least once every 12 hours.

high CDS attempts to immediately propagate an update to all replicas. If the attempt fails (for example, if one of the replicas is unavailable), a skulk is scheduled for within one hour. Skulks usually occur at least once every 12 hours. Use this setting temporarily and briefly, because it uses extensive system resources.

By default, every directory inherits the convergence setting of its parent at creation time. The default setting on the root directory is **medium**.

CDS_CTS Specifies the creation timestamp (CTS) of the directory of which this replica is a copy.

CDS_DirectoryVersion Specifies the current version of the directory (derived from the **CDS_DirectoryVersion** attribute of the clearinghouse in which the directory was created).

CDS_Epoch A UUID that identifies a particular incarnation of the directory.

CDS_InCHName Indicates whether a directory or any of its descendants can store clearinghouse names. If this value is **true**, the directory can store clearinghouse names. If it is **false**, the directory cannot store clearinghouse names.

CDS_LastSkulk Records the timestamp of the last skulk performed on this particular replica of a directory.

CDS_LastUpdate Records the timestamp of the last update to any attribute of the replica, or any change to the contents of the replica, including object entries, child pointers, and soft links.

CDS_ObjectUUID Specifies the unique identifier of the directory of which this replica is a copy.

CDS_ParentPointer Contains a pointer to this directory's parent in the namespace.

show replica(8cds)**CDS_Replicas**

Specifies the address, UUID, and name of every clearinghouse where a replica of this directory is located. This attribute also specifies whether the replica in a particular clearinghouse is a master or read-only replica.

CDS_ReplicaState

Specifies the internal state of a replica. When you create or delete a replica, it goes through various states.

CDS_ReplicaType

Specifies the replica type of a directory.

CDS_ReplicaVersion

Specifies the replica version of a directory.

CDS_RingPointer

Specifies the UUID of a clearinghouse containing another replica of this directory.

CDS_UTS

Specifies the timestamp of the most recent update to an attribute of the directory.

Privilege Required

You must have read permission to the directory from which the replica is created.

Examples The following command displays the current values of all the attributes of the replica of the `./eng` directory in the `./Chicago2_CH` clearinghouse:

```
cdscp> show replica ./eng clearinghouse ./Chicago2_CH

      SHOW
REPLICA  /.../abc.com/eng
      AT   1991-10-15-15:55:29
CDS_CTS  = 1991-10-15-12:09:47.000000003/08-00-2b-1c-8f-1f
CDS_UTS  = 1991-10-17-07:59:50.000000006/08-00-2b-1c-8f-1f
CDS_ObjectUUID = 5816da70-8b1c-11ca-8981-08002b0f79aa
CDS_Replicas = :
Clearinghouse's UUID = 2ab024a8-8b1a-11ca-8981-08002b0f79aa
      Tower = ncadg_ip_udp:16.20.16.32
      Tower = ncaen_ip_tcp:16.20.16.32
      Replica type = master
Clearinghouse's Name = /.../abc.com/Chicago1_CH
      CDS_Replicas = :
Clearinghouse's UUID = 49757f28-8b1a-11ca-8981-08002b0f79aa
```

show replica(8cds)

```
Tower = ncadg_ip_udp:16.20.16.32
Tower = ncaen_ip_tcp:16.20.16.32
Replica type = readonly
Clearinghouse's Name = ../../abc.com/Chicago2_CH
CDS_AllUpTo = 1991-10-17-07:51:18.000000032/08-00-2b-1c-8f-1f
CDS_Convergence = medium
CDS_ParentPointer = :
Parent's UUID = 560f1ad0-8b1c-11ca-8981-08002b0f79aa
Timeout = :
Expiration = 1991-10-15-19:55:18.711
Extension = +1-00:00:00.000
CDS_DirectoryVersion = 3.0
CDS_ReplicaState = on
CDS_ReplicaType = readonly
CDS_LastSkulk = 1991-10-17-07:51:18.000000032/08-00-2b-1c-8f-1f
CDS_LastUpdate = 1991-10-21-12:04:02.000000044/08-00-2b-1c-8f-1f
CDS_Epoch = 58472144-8b1c-11ca-8981-08002b0f79aa
CDS_ReplicaVersion = 3.0
```

Related Information

Commands: **create replica(8cds)**, **delete replica(8cds)**

show server

Purpose Displays attribute information about the server running on the local system

Synopsis `cdscp show server`

Description

The **show server** command displays all the names and values from the attributes named in this entity. The following are valid attribute names:

Child Update Failures

Specifies the number of times the server was unable to contact all the clearinghouses that store a replica of a particular child directory's parent directory and apply the child updates that have occurred since the last skulk. This counter is incremented by the Cannot Update Child Pointer event.

Creation Time

Specifies the time when the CDS control program process was started.

Crucial Replicas

Specifies the number of times a user attempted (from this server) to remove a replica that is crucial to the connectivity of a directory hierarchy. The server background process prevents users from accidentally disconnecting lower-level directories from higher-level directories. When it detects an attempt to remove a crucial replica, it does not execute the command to do so. This counter is incremented by the Crucial Replica event.

Future Skew Time

Specifies the maximum amount of time that a timestamp on a new or modified entry can vary from local system time at the server system.

Known Clearinghouses

Specifies the clearinghouse or clearinghouses known to the server.

Read Operations

Specifies the number of read operations directed to this CDS server.

show server(8cds)**Security Failures**

Specifies the number of times a server principal for this server was found to have inadequate permissions to perform a requested operation.

Skulks Completed

Specifies the number of skulks successfully completed by this CDS server.

Skulks Initiated

Specifies the number of skulks initiated by this CDS server.

Times Lookup Paths Broken

Specifies the number of broken connections between clearinghouses on this server and clearinghouses closer to the root. Incoming requests to this server that require a downward lookup in the directory hierarchy still succeed, but requests that require a lookup in directories closer to the root will fail. This counter is incremented by the Broken Lookup Paths event.

Write Operations

Specifies the number of write operations to this CDS server.

Privilege Required

You must have read permission to the server.

Examples The following command displays the current values of all the attributes associated with the server running on the local system:

```
cdscp> show server
```

```
          SHOW
          SERVER
          AT   1991-10-15-15:56:47
    Creation Time = 1991-10-15-15:39:35.35
    Future Skew Time = 300
    Read Operations = 757
    Write Operations = 542
    Skulks Initiated = 219
    Skulks Completed = 219
    Times Lookup Paths Broken = 1
    Crucial Replicas = 0
    Child Update Failures = 1
```

show server(8cds)

```
Security Failures = 0
Known Clearinghouses = /.../abc.com/Boston_CH
                    = /.../abc.com/NY_CH
```

Related Information

Command: **disable server(8cds)**

intro

Purpose Introduction to commands for Global Directory Service (GDS) administration

Description

The following GDS administration commands are available:

gdssysadm Calls the GDS system administration

gdsditadm Calls the directory information tree administration

gdscacheadm
Command calls the cache administration

The following GDS log file evaluation commands are available:

- Evaluates the log files of GDS processes.
- The **gdscmxml** command evaluates the log files of the transport layer, which are created by the stubs.

gdscacheadm

Purpose Cache administration program

Synopsis **gdscacheadm** [*-idirectory ID*] [*-ooperation*] [*output*] [*<input*]

Options *-idirectory ID* The directory ID. Valid values are 1 to 20.

-ooperation The administration function. Valid values are

1 Object administration

2 Cache update

If this option is not specified, the **gdscacheadm** program outputs Mask 3 (see the *OSF DCE Administration Guide*).

Arguments

output Name of the output file in batch mode.

input Name of the input file in batch mode.

Description

The **gdscacheadm** shell command is used to call the cache administration. The options **-i** and **-o** are only possible in dialog mode, and *output* and *input* are only used in batch mode.

Structure of the Input File in Batch Mode

An input file is required for the directory administration in batch mode. This file has to be provided with the mask entries in the same mask sequence as for dialog mode.

Every input in a mask field must be entered in a separate line in the input file. The end-of-line character is interpreted as a key input.

Any space remaining following a value in an input field must be filled with underscores so that longer values specified beforehand for the same field are overwritten correctly.

Comments can be inserted and must be enclosed between : (colons).

gdscacheadm(8gds)

Comments that begin with ":*" (double quote, colon, asterisk, double quote) and end with "*::" are transferred from the input file to the output file.

Structure of the Output File in Batch Mode

The output file in batch mode contains the comments transferred from the input file and the program messages displayed in the masks. Error messages are thus indicated by the character string ERROR.

If the administration program is called with no further option, it outputs the Logon Menu Mask. (See the *OSF DCE Administration Guide*.)

Exit Values

If the command executes successfully, the return value is 0 (zero); otherwise, the value is nonzero.

gdscmxl

- Purpose** Controls and edits transport system interface library trace information
- Synopsis** `CMXTRACE = "-s | S [-p fac] [-r wrap]"; export CMXTRACE
gdscmxl [-cdeXt] [-v | x] file...`
- Options** The following options are available for editing the trace information using the `gdscmxl` command:
- c** Edits the transport system interface calls for
 - Attaching and detaching the application to or from the transport system interface
 - Connection setup and termination
 - Connection redirection
 - d** Edits the transport system interface calls for data exchange and flow control.
 - e** Edits the transport system interface calls for event handling.
 - X** Edits XTI calls.
 - t** In addition to logging error messages, edits the `t_error` calls explicitly. Error messages are always logged, even if this option is not specified.
 - v** Edits in detail the transport system interface calls, their parameters, their options, and user data. The extent that data is edited depends on the options specified in `CMXTRACE`. You can specify this option or the `-x` option, but not both.
 - x** Edits (to a limited extent) the calls and their parameters (excluding options), and user data. You can specify this option or the `-v` option, but not both.

When no options are specified, the default is `cdex`.

The following options are available for controlling the trace mechanism with the `CMXTRACE` environment variable:

gdscmxl(8gds)

- s Keeps an ordinary log of the transport system interface calls, their parameters, the options, and user data. You must specify either this option or the **-S** option.

- S Keeps a detailed log of the calls, their parameters, the contents of any options, and the user data in its full length. You must specify either this option or the **-s** option.

- p *fac* The decimal digit *fac* determines the buffering factor. The amount of buffering is *fac* * **BUFSIZ**, with **BUFSIZ** as defined in the **stdio.h** header. If 0 (zero) is specified for *fac*, every trace entry is written immediately to the file (unbuffered).

Any value specified for *fac* exceeding 8 is automatically reduced to 8.

If **-p *fac*** is not specified, the default is 1.

- r *wrap* The decimal number *wrap* specifies that logging is to be directed to the second temporary file **CMXLbpid** after *wrap* * **BUFSIZ** bytes (with **BUFSIZ** as defined in the **stdio.h** header). The second file **CMXLbpid** is handled by the trace mechanism in the same way as **CMXLapid**. After every *wrap* * **BUFSIZ** bytes, the trace mechanism switches between **CMXLapid** and **CMXLbpid**. When this is completed, the old contents of the second file are lost.

If **-r *wrap*** is not specified, the default value for *wrap* is 128.

Arguments

- file ...* The file or files with binary trace entries to be edited.

Description

The trace mechanism of the transport system interface library is activated and controlled by the environment variable **CMXTRACE**. The trace entries of a process are collected in compressed, binary format in a dynamically created buffer, and are periodically written to temporary files. These files are edited in a separate step with the **gdscmxl** function.

Controlling the Trace Mechanism with CMXTRACE

Every **t_attach** call issued by a process evaluates the **CMXTRACE** environment variable and, when appropriate, activates the trace mechanism.

CMXTRACE must be set before the first **t_attach** call of the process to be monitored.

After the trace mechanism is activated, the temporary file **CMXL***apid* with the process ID *pid* is opened if it is not already open. The file is granted the following access permissions:

```
rw----- (0600)
```

The file is located under the login name of the process. Storage is then dynamically reserved for buffering the trace entries. Storage and files are reserved for the life of the process.

The options specified in **CMXTRACE** control the trace mechanism. The **-s** and **-S** options determine the extent of logging, and the **-p** and **-r** options control buffering and cyclical overwriting of the file.

Editing the Trace Information with **gdscmxi**

The **gdscmxi** function reads the entries generated by the trace mechanism from the temporary file **file**, processes them in accordance with the specified options, and outputs the results on **stdout**.

Files

Files with compressed trace entries are

```
<dce-local>/var/adm/directory/gds/cstub/CMXLa<pid>
```

```
<dce-local>/var/adm/directory/gds/cstub/CMXLb<pid>
```

```
<dce-local>/var/directory/gds/adm/sstub/CMXLa<pid>
```

```
<dce-local>/var/directory/gds/adm/sstub/CMXLb<pid>
```

Exit Values

If the command executes successfully, the return value is 0 (zero); otherwise, the value is nonzero.

The following self-explanatory error messages can be output on **stderr**:

```
Syntax: gdscmxi [-cdevx] file ...
```

```
gdscmxi: error <errno> opening <filename>
```

```
gdscmxi: bad first entry <entry>
```

```
gdscmxi: cannot process format <format>
```

The logged error number *errno* can be decoded with the **errno.h** header.

gdsdirinfo

Purpose Displays information on GDS daemon processes and processes using GDS

Synopsis `gdsdirinfo`

Description

The `gdsdirinfo` command is used to obtain information on all daemon processes running for GDS and on all current processes using GDS. All of the information is read by `gdsdirinfo` from the GDS-specific shared memory area and is written to `stdout`. First, a two-line header is printed, then the information specific to the different processes (one line per process) follows. The following information is displayed:

PROCTYPE The process type. The following types can occur:

Monitor	IPC-monitoring process
DUA-Cache	DUA-cache process
C-Stub	C-stub process
S-Stub	S-stub process
DSA	DSA process
Dir-User	Process using GDS (GDS-Client)

PID The process identifier.

DIRID The directory identifier (1 - 20) with which the process is associated. If a process cannot be associated with a specific directory identifier (for example, the DUA-Cache process) a dash (-) is printed instead of a directory identifier number.

IPCID The IPC server ID with which the process is associated. This ID is used internally by GDS to establish an IPC association between an IPC client and an IPC server for sending distributed commands (for example, in the case of activation and or deactivation of the trace system). The processes are assigned IPC server IDs as follows:

1	DUA-cache process
---	-------------------

2	C-stub process
5	IPC-monitoring process
11-30	S-stub processes
31-50	DSA-processes

If the process type is **Dir-User**, the IPCID displayed refers to the GDS IPC server (for example, DUA-cache, C-stub, DSA) with which this GDS client is associated.

Note that the following relationship exists between the directory identifier and the IPC server identifier for S-stub processes and DSA processes:

S-Stub	Dir-ID = IPC-server-ID - 10
DSA	Dir-ID = IPC-server-ID - 30

STATE

Describes the state of the following:

- a GDS process during the startup phase. The following values are valid:

W1	C-stub/S-stub tries to read its own PSAP address from DUA-cache
W3	DSA tries to read its own DSA name from file
W4	DSA tries to read the internal schema
W5	DSA changes the schema object in the database
R1	The process is ready to run

- A GDS-client. The values and their meanings are as follows:

R1	IPC association between GDS client and GDS server exists
R10	DAP/DSP association between GDS client and GDS server exists

If none of the specific states is associated with the process, a dash (-) is printed instead.

gdsdirinfo(8gds)

Examples The following is an example of **gdsdirinfo** output:

#	PROCTYPE	PID	DIRID	IPCID	STATE
#	Monitor	4105	-	5	-
	DUA-Cache	4106	-	1	-
	C-Stub	4108	-	2	-
	S-Stub	4118	1	11	-
	S-Stub	4123	2	12	W1
	DSA	4130	1	31	-
	DSA	4125	2	32	-
	Dir-User	4300	-	31	R10

If the **gdsdirinfo** command is called when the GDS is inactive, the following message is written to **stderr**:

```
gdsdirinfo: can't get IPC-resources (errno = 2)
```

Exit Values

If the command executes successfully, the exit value 0 (zero) is returned; otherwise, the value is 1 or 2.

gdsditadm

Purpose Directory database administration program

Synopsis **gdsditadm** [-i*directory ID*] [-d*dsa-dn*] [-o*operation*] [-p*password*] [-u*user*]
[*output*] [<*input*]

Options -i*directory ID* The directory ID. Valid values are 1 to 20.

-d*dsa-dn* Distinguished Name (DN) of the bind Directory System Agent (DSA). If this option is not specified, a connection is set up to the default DSA. The following specifications are valid:

- DN of the DSA; for example,
/C=us/0=Smith Ltd./OU=dep.1/CN=DSA/CN=DSA1
- SPECIFIC_DSA: Mask 2 is displayed to select the DSA. (See the *OSF DCE Administration Guide*.)
- CACHE: connection to the Directory User Agent (DUA) cache.

-o*operation* The administration function.

If the connection is to a DSA, valid values are as follows:

- | | |
|---|------------------------|
| 1 | Object administration |
| 2 | Schema administration |
| 3 | Shadow administration |
| 4 | Subtree administration |

If the connection is to the DUA cache, valid values are as follows:

- | | |
|---|-----------------------|
| 1 | Object administration |
| 2 | Cache update |

If the parameter is not specified, the **gdsditadm** command outputs Mask 3. (See the *OSF DCE Administration Guide*.)

-p*password* User password (password of the administrator object). If this parameter is not specified, an anonymous bind is performed.

gdsditadm(8gds)

-uuser Distinguished Name of the user (DN of the administrator object). If this parameter is not specified, an anonymous bind is performed.

Arguments

output Name of the output file in batch mode

input Name of the input file in batch mode

Description

The **gdsditadm** shell command is used to call the directory database administration. The options **-i**, **-d**, **-o**, **-p**, and **-u** are only valid in dialog mode and *output* and *input* are only used in batch mode.

Structure of the Input File in Batch Mode

An input file is required in batch mode. This file must include the mask entries in the same mask sequence as for dialog mode.

Every input in a mask field must be on in a separate line in the input file. The end-of-line character is interpreted as a key input.

Any space remaining following a value in an input field must be filled with underscores so that longer values previously specified for the same field are overwritten correctly.

Comments can be used, and must be enclosed between colons (:).

Comments that begin with ":"* (double quote, colon, asterisk, double quote) and end with ":*" are transferred from the input file to the output file.

Structure of the Output File in Batch Mode

The output file in batch mode contains the comments transferred from the input file and the program messages displayed in the masks. Error messages are indicated by the character string ERROR.

If the administration program is called with no option, it outputs the Logon Menu Mask. (See the *OSF DCE Administration Guide*.)

Environment Variables

NLSPATH = <dce-local>/nls/msg/en_US.ASCII/%N.cat

Examples

1. The following is an example of an input file:

```
:*****TEST - ADD ADMINISTRATOR DSA OP=1,10*****:
:Directory id:1
```

```
:Password:
:Country:
:Organization:
:Organizational Unit:
:Common name:
:Options():
:****Administration ****:
:Function:1
:****AddObject /C=de *****:
:Operation:01
:Object Type Number:02
:Country:de
:Structural Object Class:Country
:Auxiliary Object Class:NO
:ACL:NO
:Attribute Name:
:Attribute Value:
:Attribute Value:
:Attribute Name:
:Attribute Value:
:Attribute Value:
:Attribute Name:
:Attribute Value:
:Attribute Value:
:More attr:_
:****AddObject /C=de/O=Smith *****:
:Operation:01
:Object Type Number:03
:Country:de
:Organization:Smith
:Structural Object Class:Organization
:Auxiliary Object Class:NO
:ACL:NO
:Attribute Name:
:Attribute Value:
:Attribute Value:
:Attribute Name:
:Attribute Value:
:Attribute Value:
:Attribute Name:
:Attribute Value:
:Attribute Value:
:More attr:_
:****AddObject /C=de/O=Smith/OU=Sales *****:

```

gdsditadm(8gds)

```
:Operation:01
:Object Type Number:04
:Country:de
:Organization:Smith
:Organizational Unit:Sales
:Structural Object Class:Organizational-Unit
:Auxiliary Object Class:NO
:ACL:NO
:Attribute Name:
:Attribute Value:
:Attribute Value:
:Attribute Name:
:Attribute Value:
:Attribute Value:
:Attribute Name:
:Attribute Value:
:Attribute Value:
:More attr:_
:****AddObject /C=de/O=Smith/OU=Sales/CN=Schmid *****:
:Operation:01
:Object Type Number:05
:Country:de
:Organization:Smith
:Organizational Unit:Sales
:User:Schmid
:Structural Object Class:Organizational-Person
:Auxiliary Object Class:NO
:ACL:NO
:Attribute Name:Surname
:Attribute Value:schmid'
:Attribute Value:
:Attribute Name:User-Password
:Attribute Value:schmid'
:Attribute Value:
:Attribute Name:
:Attribute Value:
:Attribute Value:
:More attr:_
:****ModifyACL /C=de/O=Smith/OU=Sales/CN=Schmid *****:
:Operation:9
:Object Type Number:05
:Country:de
:Organization:Smith
:Organizational Unit:Sales
```

:User:Schmid
:Modify Public:YES
:Read Standard:YES
:Modify Standard:YES
:Read Sensitive:YES
:Modify Sensitive:YES
:Distinguished Name:/C=de/O=Smith/OU=Sales/CN=Schmid_____
:Distinguished Name:_____
:IP:OBJECT
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:/C=de/O=Smith/OU=Sales/CN=Schmid_____
:Distinguished Name:_____
:IP:OBJECT
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:/C=de/O=Smith/OU=Sales/CN=Schmid_____
:Distinguished Name:_____
:IP:OBJECT
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:/C=de/O=Smith/OU=Sales/CN=Schmid_____
:Distinguished Name:_____

gdsditadm(8gds)

```

:IP:OBJECT
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:/C=de/O=Smith/OU=Sales/CN=Schmid_____
:Distinguished Name:_____
:IP:OBJECT
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:Distinguished Name:_____
:Distinguished Name:_____
:IP:
:****END****:
:Operation:00
:****END TEST****:
:Operation:00
    
```

2. The following is an example of an output file:

```

BIND                elapsed time:      0.0000 sec
***Administration ***:
***AddObject /C=de *****:
***AddObject /C=de/O=Smith *****:
***AddObject /C=de/O=Smith/OU=Sales *****:
***AddObject /C=de/O=Smith/OU=Sales/CN=Schmid *****:
***ModifyACL /Code/O=Smith/OU=Sales/CN=Schmid *****:
***END****:
    
```

Exit Values

If the command executes successfully, the return value is 0 (zero); otherwise, the value is nonzero.

gdsstep

Purpose GDS trace evaluation program

Synopsis `gdsstep [-d] [-l=nnn[k]] [-s=nllmlh] [-r=nnn] filename...`

Options

- d** Data records are dumped; no analysis of session user data is performed.
- l=*nnn*** Dumps are restricted to a maximum of *nnn* bytes (rounded off to a multiple of 16); the length of the data is indicated in the message output limit reached.
- l=*nnn*** Dumps are restricted to a maximum of *nnn* Kbytes.
- s=** Security level for the analysis of session user data. Valid values are described in the following list. The default value is **m**.
 - n** No security level switched on.
 - l** Passwords are not listed.
 - m** User identifications, account numbers, and passwords are not listed.
 - h** Similar to **m**, but filenames are not listed in this case either.
- r=*nnn*** Option for suppressing evaluation of **ros_trace**. *nnn* is a logical OR of the following values:
 - 1** No ros internal trace information is evaluated.
 - 2** No ros interface trace is evaluated.
 - 4** No ros activity trace is evaluated.
 - 8** No user header is printed.

Arguments

filename Name of the log file that is to be evaluated. More than one log file can be specified.

gdsstep(8gds)

Description

The **gdsstep** program evaluates a trace file containing a GDS trace. The result of the evaluation is in printable form.

Logging files that are generated by GDS applications can be evaluated and displayed at a later time with the **gdsstep** program. The logging records can be used by application programmers to debug their applications, and by users to determine network problems or protocol problems with remote systems. The information is intended to be self-explanatory.

Files

See the *OSF DCE Administration Guide* for the names of the log files and the command that is used to evaluate them.

gdssysadm

Purpose Directory system administration program

Synopsis `gdssysadm -function [-ddirectory ID]`
`[-mconf.mode]`
`[-cconfiguration type]`
`[-Cnumber of clients]`
`[-snumber of servers]`
`[-udelta update] [-ooperation]`
`[-Mdata medium type] [-k[password]]`
`[-Fformatting] [-vvolume number]`
`[-p] [-X]`

Options `-function` The administration function to be executed. This option must always be specified as the first option. Valid values are

c	Configure directory system
A	Activate directory installation
d	Deactivate directory installation
s	Save local directory data files
r	Restore directory data files saved
t	Deactivate trace system of the directory
T	Activate trace system of the directory
i	Display status information of the directory system

`-ddirectory ID` The directory ID. Valid values are 1 to 20.

`-mconf.mode` The configuration mode. Valid values are

1	Enter configuration data
2	Delete configuration data
3	Display configuration data
4	Change configuration data

gdssysadm(8gds)

-cconfiguration type

The configuration type. Valid values are

- 1 Client system
- 2 Client/server system

-Cnumber of clients

The number of clients that can have access to the directory at the same time. Valid values are 1 to 256.

-snumber of servers

The number of server processes to be activated. Valid values are 1 to 256.

-udelta update

The delta update mode. Valid values are

- 0 With no delta update operations
- 1 With delta update operations

-ooperation The backup operation code. Valid values are

- 1 Initialize saving or restoring of data files
- 2 Write or read data files to or from data medium
- 3 End saving or restoring of data files
- 4 Determine number of data media required

-Mdata medium type

The type of media to be used. Valid values are

- 0 Diskette
- 1 Tape

-kpassword The password for protecting the directory system data files saved (maximum of 10 characters).

-Fformatting The code for the data medium formatting. Valid values are

- 0 No formatting of data medium
- 1 Formatting of data medium

- v***volume number*
The volume number of the security data medium.
- p**
Suppresses error messages.
- X**
Turns on a trace of the administration procedure **gdssysadm**. The default is no trace.

Description

The **gdssysadm** command calls the GDS system administration program.

Combining Options

The following table shows which options can be specified in the command call depending on the administration function to be executed. If one of these options is missing in the call, the **gdssysadm** command outputs the corresponding administration mask (see the *OSF DCE Administration Guide*), so that the user can enter the missing parameter interactively. The table also contains the mask number that the **gdssysadm** command outputs if one of the options is not specified. In these mask outputs, all input fields where **gdssysadm** expects no input are omitted, because the parameter is already specified as an option. (See the *OSF DCE Administration Guide* for the relevant masks.)

gdssysadm(8gds)

Combining Options in the Command Call														
Function	Options													Mask
	d	m	c	C	s	u	o	M	k	F	v	p	X	No.
c (configure)	d	m	c	C	s	u						p	X	3
A (activate)												p	X	
d (deactivate)												p	X	
s (save)	d						o	M	k	F	v	p	X	5
r (restore)	d						o	M	k		v	p	X	6
t (do not log)												p	X	
T (log)												p	X	
i (display)												p	X	

Environment Variables

NLSPATH = <dce-local>/nls/msg/en_US.ASCII/%N.cat

Notes

The function code must always be specified as the first option. All the other options can be used in any sequence.

No blank spaces can be inserted between the ID and the value of an option.

Examples

The following command can be used to save the data files of a directory using the internal interface (as in dialog):

gdssysadm -fs [-ddirectory ID] [-Mdata medium type] [-k[password]] [-Fformatting]

The following calls are required in order to save the data files of a directory using a separate user interface:

gdssysadm -fs -ddirectory ID -o1 -Mdata medium type -k[password]

This call ensures that the DSA or DUA cache returns **TOO_BUSY** to each request.

gdssysadm -fs -ddirectory ID -o4 This call returns the number of data media to the **stdout** file.

It is necessary to perform the following commands once for each data medium required:

gdssysadm -fs -ddirectory ID -o2 -vvolume number -Mdata medium type -ffformatting

This command indicates that the directory data is written to the media.

gdssysadm -fs -ddirectory ID -o3

This command ensures that the DSA or the DUA cache is accepting requests again.

The following command can be used to restore the data files of a directory using the internal interface (as in dialog):

```
gdssysadm -fr [-ddirectory ID] [-Mdata medium type] -k[password]
```

The following calls are required in order to restore the data files of a directory using a separate user interface:

```
gdssysadm -fr -ddirectory ID -o1 -vvolume number of first data medium
```

```
-Mdata medium type -k[password]
```

This call ensures that the DSA or DUA cache returns **TOO_BUSY** to each request.

```
gdssysadm -fr -ddirectory ID -o4
```

This call returns the number of data media to the **stdout** file.

It is necessary to perform the following commands once for each data medium required:

```
gdssysadm -fr -ddirectory ID -o2 -vvolume number -Mdata medium type
```

This command specifies that the data medium is read.

```
gdssysadm -fr -ddirectory ID -o3
```

This command ensures that the DSA or the DUA cache is accepting requests again.

Exit Values

The following table lists the exit values and explains their meaning.

Exit Values		
Values	Type	Meaning
1	Fatal error	Wrong syntax (<i>parameter</i>)
1	Fatal error	Illegal function code (<i>parameter</i>)
1	Fatal error	Parameter not allowed (<i>parameter</i>)
1	Error	The selected function cannot be executed
2	Error	The selected function can be executed
3	Error	The directory system is active
4	Error	The directory system is not configured
5	Error	The directory system is not active
6	Error	Configuration information does not exist
7	Error	Configuration data already exists
8	Error	Cannot send distributed command to directory system processes
9	Error	Wrong selection
10	Error	The directory system is still in use
11	Error	Cannot read media volume label
12	Error	Wrong media volume label
13	Error	This directory identifier is not configured
14	Error	Directory ID is not configured
15	Error	Invalid media volume configuration information
16	Error	Cannot format media volume
17	Error	Cannot write data to media volume
18	Error	Cannot read data from media volume
19	Error	Cannot read file list
20	Error	The selected language is not installed
21	Error	One file exceeds the size of the media volume
22	Error	Cannot find any files that can be saved
23	Error	The selected directory system installation does not exist
25	Error	Important directory process cannot be started
26	Warning	The restored data base does not fit in the DSA
27	Warning	The restored data base does not fit i the DUA cache
28	Error	Invalid input

Chapter 3

Distributed Time Service Commands

Intro

Purpose Introduction to the DCE DTS control program commands

Description

The DCE Distributed Time Service control program (**dtscp**) allows you to synchronize, adjust, and maintain the system clocks in a distributed network. The DTS control program commands are

- advertise** Configures the DTS server as a global server
- change** Modifies the epoch and sets the local time to a new time
- create** Establishes a DTS entity (a clerk or server)
- delete** Causes DTS to exit on the local node
- disable** Suspends a DTS entity
- enable** Starts a DTS entity
- set** Modifies characteristics of a DTS entity
- show** Displays characteristics of a DTS entity
- synchronize** Synchronizes the system clock with the time obtained from DTS servers in the network
- unadvertise** Removes the global server entry
- update** Gradually adjusts the system clock to a new time

Related Information

Commands: **dtscp(8dts)**

Books: *OSF DCE Administration Guide*, *OSF DCE Administration Reference*

advertise

Purpose Configures the system as a global server by adding the server's entry to the cell profile

Synopsis `advertise`

Description

The **advertise** command causes DTS to forward the name and attributes of the server to CDS by binding the server's protocol tower to the CDS object and adding an entry for the server in the cell profile. Once the server's entry is in the cell profile, it is configured as a global server, and servers outside of the LAN can access it.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Examples `dt.scp> advertise`

Related Information

Commands: **unadvertise (8dts)**

change(8dts)

change

Purpose Alters the epoch number and time on the local node

Synopsis `change epoch integer [time absolute-time]`

Arguments

epoch *integer* Specifies the new epoch number. This argument is required.

time absolute-time

Specifies a clock setting for the new epoch. If you do not supply this argument and a value, the server uses the current clock time with an unspecified inaccuracy and initiates a synchronization. This argument is optional.

Description

The **change** command sets the time and changes the epoch of the DTS server on which it is entered. Use this command to isolate a server from the rest of the servers in the network before changing the time.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Notes This command is valid only for servers. The new epoch number you specify must be different from the current epoch number.

Examples The following example shows how to change the epoch number:

```
dtscp> change epoch 1
```

The following example shows how to change the epoch number and time:

```
dtscp> change epoch 1 time 1990-11-30-10:58:00.000-05:00I0.000
```

create

Purpose Creates the DCE DTS entity on the specified node

Synopsis `create type type`

Arguments

type *type* Specifies the type of DCE DTS entity to be created on the specified node. Specify one of the following for *type*:

clerk The DCE DTS entity is created as a clerk. (The default setting is **clerk**.)

server The DCE DTS entity is created as a server.

Description

The **create** command creates a time server or time clerk entity on the system where the command is entered.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Examples `dtscp> create type server`

delete(8dts)

delete

Purpose Deletes the DCE DTS entity

Synopsis `delete`

Description

The **delete** command deletes the DCE DTS entity from the system where the command is entered. When **delete** is executed, the DTS daemon process completes execution. To restart the DTS daemon, use the **dce_config** shell command.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Notes The DCE DTS entity cannot be deleted until you enter the **disable** command, which causes the status attribute **state** to be set to **off**.

Examples `dt:scp> delete`

Related Information

Commands: **disable (8dts)**

disable

Purpose Stops the DCE DTS entity on the specified node

Synopsis `disable`

Description

The **disable** command turns off the DCE DTS entity on the system where the command is entered. When the command is executed, the status attribute **state** is set to **off**.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Notes The DCE DTS entity cannot be disabled until it is enabled with the **enable** command. You must enter the **disable** command before you can delete the entity.

Examples `dt.scp> disable`

Related Information

Commands: **enable (8dts)**, **delete (8dts)**, **create (8dts)**

dtscp(8dts)

dtscp

Purpose DTS control program

Synopsis /usr/bin/dtscp

Description

This section describes the commands for the DCE Distribute Time Service (DTS) control program (**dtscp**). The DTS control program is a command-line interface that enables you to synchronize, adjust, and maintain the system clocks in a distributed network. For a detailed explanation of system clock synchronization and management, see the *OSF DCE Administration Guide*.

The DTS control program commands are

- **advertise**
- **change**
- **create**
- **delete**
- **disable**
- **enable**
- **set**
- **show**
- **synchronize**
- **unadvertise**
- **update**

You can use control program commands from within the control program or from the system prompt. To enter DTS commands from within the control program, first start the control program by entering the **dtscp** command. For example:

```
$ dtscp
dtscp>
```

At this prompt you can enter any control program command; for example:

```
dtscp> show current time
```

To leave the control program and return to the system prompt, enter the **exit** command.

To enter DTS commands from the system prompt (interactively or in a command procedure) enter the **dtscp** command with an internal command of the control program as the first argument. The control program executes the command without displaying the control program prompt. For example, you can enter the **synchronize** command as follows:

```
$ dtscp synchronize
```

Some **dtscp** commands have optional arguments or attributes; there may also be optional variables for the arguments and attributes.

A sample command follows:

```
dtscp> update time 1990-08-03-05:45:28.000+01:00I00.500
      /   /   /
      Command [Argument] Variable
      -----
      [Attribute]
```

Abbreviations

You can enter as few as three characters for each DTS command or argument; DTS commands and arguments are unique for three characters or more. For example, rather than entering the command **enable set clock true**, you can enter the following abbreviated command:

```
dtscp> ena set clo tru
```

Attributes

The **dtscp set** and **show** commands have several attributes—pieces or sets of data associated with them. The attribute groups are categorized as follows:

Characteristics

Set or show the entity's operation.

Counters

Show the number of occurrences of an event since the entity was enabled.

Status

Show the current state of the entity. (The DTS entity has four status attributes.)

Global Servers

Show the global servers known by this DTS entity.

dtscp(8dts)

Local Servers

Show the local servers known by this DTS entity.

Individual attributes within each of the previously listed groups are described in the reference pages for the **set** and **show** commands. The **show** command also allows you to specify attribute groups.

Related Information

Books: *OSF DCE Administration Guide*

dtstd

Purpose Restarts the DTS daemon

Synopsis `dtstd`

Description

The **dtstd** command restarts the DTS daemon (clerk or server process). When the host system is rebooted, this command is automatically executed as part of the overall DCE configuration procedure. You can enter the command manually if a DTS daemon fails to start automatically upon reboot, or if you want to restart a daemon that you disabled and delete to perform a backup or do diagnostic work. After the process starts, you must execute the **dtscp** commands **create** and **enable** to run DTS.

Privilege Required

Only the local host machine principal can start the DTS daemon. See the Security reference pages for information about principals.

Notes Use **dtstd** interactively only when troubleshooting; use the **dce_config** shell command to start the DTS daemon. On some systems the superuser is associated with the machine principal so it is not necessary to use the **dce_config** shell command to start the DTS daemon.

Examples To restart the daemon, follow these steps:

1. Log in to the system as superuser (**root**).
2. Log in to DCE as the machine principal of the local host. Enter the principal name in the format **hosts/hostname/self** as shown in the following example command for a host named **mystic** whose password is **smith**:

```
$ dce_login hosts/mystic/self smith
```
3. Enter the following command to see if the **rpcd** process is already running:

```
$ ps
```

dttd(8dts)

If the **rpcd** process appears on the list of active processes, proceed to step 4.

If the **rpcd** process does not appear on the list of active processes, enter the following command to start the process:

```
$ rpcd
```

4. Enter the following command to restart the server:

```
$ dttd
```

Related Information

Commands: **dttd** (8dts), **create** (8dts), **enable** (8dts)

Books: *OSF DCE Administration Guide*

enable

Purpose Starts the DTS entity on the specified node.

Synopsis `enable [set clock boolean]`

Arguments

set clock *boolean*

Specifies whether the clock is abruptly set or gradually adjusted to the computed time. This argument is optional. Valid values for *boolean* are

false The clock is gradually adjusted. This is the default condition.

true The clock is abruptly set.

Description

The **enable** command creates the DTS entity on the system where the command is entered. When the command is executed, the status attribute **state** is set to **on**.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Notes The DTS entity cannot be enabled until it is created with the **create** command; the DTS entity must be in the **off** state.

Examples The following example shows how to enable the entity and adjust the clock gradually to the computed time following the first synchronization:

```
dtscp> enable
```

The following example shows how to enable the entity and abruptly set the clock to the computed time following the first synchronization:

```
dtscp> enable set clock true
```

enable(8dts)

Related Information

Commands: **create (8dts)**, **disable (8dts)**

exit

Purpose Causes the DTSCP to complete execution

Synopsis `exit`

Description

The **exit** command causes the DTS control program (DTSCP) to complete execution and returns operation to the parent process.

Examples The following command shows how to leave the DTSCP and return to the parent process:

```
dt.scp> exit
```

Related Information

Commands: **quit(8dts)**

help(8dts)

help

Purpose Displays help information about DTSCP commands

Synopsis `help [help-topic]`

Arguments

help-topic Specifies the help topic for which help information is to be displayed. The following are valid help topics:

- **advertise**
- **change**
- **create**
- **delete**
- **disable**
- **enable**
- **set**
- **show**
- **synchronize**
- **unadvertise**
- **update**

Description

The **help** command displays information about DTSCP commands.

Examples The following command shows how to get help about the help subtopic **unadvertise**.

```
dtscp> help unadvertise
```

quit

Purpose Causes the DTSCP to complete execution

Synopsis `quit`

Description

The **quit** command causes the DTSCP to complete execution and returns operation to the parent process.

Examples The following command shows how to leave the DTSCP and return to the parent process:

```
dt.scp> quit
```

Related Information

Commands: **exit(8dts)**

set(8dts)

set

Purpose Modifies characteristics for the DTS entity

Synopsis `set characteristic`

Arguments

characteristic The name and value of one or more characteristics to be modified. Valid values for *characteristic* are described in the following list. These values are described in more detail in the description section.

- **check interval** [*relative-time*]
- **courier role** [*role*]
- **error tolerance** [*relative-time*]
- **global set timeout** [*relative-time*]
- **local set timeout** [*relative-time*]
- **maximum inaccuracy** [*relative-time*]
- **query attempts** [*integer*]
- **server entry name** [*name*]
- **server principal name** [*name*]
- **servers required** [*integer*]
- **synchronization hold down** [*relative-time*]
- **global set timeout** [*relative-time*]

Description

The **set** command modifies the characteristics you specify for the DTS entity. The modifiable characteristics and their values are described in the following list.

check interval [*relative-time*]

Specifies the amount of time between checks for faulty servers. Applicable only for servers that have external time providers.

Default: **0-01:30:00.000**

Value: **0-00:00:30.000 99-99:99:99.999**

courier role [*role*]

Specifies a server's interaction with the set of global servers.

Default: **backup courier**

The following values are valid:

backup courier

The local server becomes a courier if none are available on the LAN.

courier

The local server synchronizes with the global set of servers.

noncourier

The local server does not synchronize with the global set of servers.

error tolerance [*relative-time*]

Specifies the maximum separation allowed between the local clock and the computed time before synchronizations become abrupt rather than gradual (monotonic).

Default: **0-00:10:00.000**

Value: **0-00:00:00.500 1334399-21:21:00.685**

global set timeout [*relative-time*]

Specifies the amount of time the node waits for a response to a global synchronization request before sending another request or declaring a global server to be unavailable. The number of attempts made to reach the server is controlled by the **query attempts** characteristic.

Default: **0-00:00:15.000**

Value: **0-00:00:01.000-0-00:10:00.000**

set(8dts)**local set timeout** [*relative-time*]

Specifies the amount of time the node waits for a response to a local synchronization request before sending another request or declaring a server to be unavailable. The number of attempts made to reach the server is controlled by the **query attempts** characteristic.

Default: **0-00:00:02.000**

Value: **0-00:00:01.000-0-00:05:00.000**

maximum inaccuracy [*relative-time*]

Specifies the inaccuracy limit for the node. When the node exceeds the maximum inaccuracy setting, it attempts to synchronize.

Default: **0-00:00:00.100**

Value: **0-00:00:00.000-10675199-00:00:00.000**

query attempts [*integer*]

Specifies the number of attempts that a node makes to contact a server before the node considers the server unavailable.

Default: **3**

Value: **1-10**

server entry name [*name*]

Specifies a server's CDS entry name; *hostname* represents the name of the system or node that is the server's client. The default setting is the recommended value.

Default: **./:/hosts/hostname/dts-entity**

server principal name [*name*]

Specifies a server's principal name for authentication purposes; *hostname* represents the name of the system or node that is the server's client. The default setting is the recommended value.

Default: **./:/hosts/hostname/dts-entity**

servers required [*integer*]

Specifies the minimum number of servers required for a synchronization. Settings of 1 or 2 may cause unreliable computed times.

Default: **1 (clerks) 3 (servers)**

Value: **1-10**

synchronization hold down [*relative-time*]

Specifies the interval a node must wait to synchronize. Also specifies synchronization frequency when a node reaches the value specified by the **maximum inaccuracy** characteristic.

Clerks:

Default: **0-00:10:00.000**

Value: **0-00:00:30.000-01-00:00:00.000**

Servers:

Default: **0-00:02:00.000**

Value: **0-00:00:30.000-01-00:00:00.000**

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Examples The following example shows how to set the check interval:

```
dtscp> set check interval 00-00:00:30.000
```

The following example shows how to set the number of servers required before the entity can synchronize:

```
dtscp> set servers required 4
```

The following example shows how to set the courier role for a server:

```
dtscp> set courier role backup courier
```

Related Information

Commands: **show (8dts)**

show(8dts)

show

Purpose Displays current information about the DTS entity

Synopsis `show attribute-group | attribute-name`

Arguments

attribute-group

The name of an attribute group or individual attribute to be displayed. The following values are valid:

- **all**
- **all characteristics**
- **all counters**
- **all status**
- **global servers**
- **local servers**

attribute-name

The name of a specific attribute from the **characteristics**, **counters**, or **status** group. The attribute specifiers **global servers** and **local servers** do not contain any other attributes.

Description

The **show** command displays the names and values of the specified attributes or attribute groups. For attribute groups, if you do not supply a group name with the **all** argument, all characteristics and their values are displayed. The following sections list names of individual attributes, categorized by group.

Characteristics

Specifies whether a backup courier is currently functioning as a courier. If the role is **noncourier**, the node is not attempting to synchronize with global servers.

Default: **noncourier**

Value: **courier** or **noncourier**

advertisement interval

Specifies the amount of time between a server's advertisement broadcasts.

Default: **0-01:30:00.000**

Value: **0-00:00:30.000 - 99-99:99:99.999**

automatic tdf change

Specifies whether automatic changes to the time differential factor are enabled or disabled; the value is determined by the operating system.

Default: **true**

Value: **true/false**

check interval

Specifies the amount of time between checks for faulty servers. Applicable only to servers that have external time providers.

Default: **0-01:30:00.00**

Value: **0-00:00:30.000 - 99-99:99:99.999**

clock adjustment rate

Specifies the rate at which the DTS server or clerk entity adjusts the node's clock during a synchronization.

clock resolution

Specifies the amount of time between system clock ticks. The value is determined by the operating system.

courier role

Specifies a server's interaction with the set of global servers.

backup courier

The local server becomes a courier if none are available on the LAN.

courier The local server synchronizes with the global set of servers.

noncourier The local server does not synchronize with the global set of servers.

Default: **noncourier**

show(8dts)

DTS version

Specifies the DTS software version installed on the node.

epoch number

Specifies the server's epoch number. The **change** command modifies this characteristic.

Default: **0**

Value: **0-100**

error tolerance

Specifies the maximum separation allowed between the local clock and the computed time before synchronizations become abrupt rather than gradual (monotonic).

Default: **0-00:10:00.000**

Value: **0-00:00:00.500 - 99-99:99:99.999**

local set timeout

Specifies the amount of time the node waits for a response to a synchronization request before sending another request or declaring a server to be unavailable. The number of attempts made to reach the server is controlled by the **query attempts** characteristic.

Default: **0-00:00:05.000**

Value: **0-00:00:00.000 - 0-00:01:00.000**

local servers

Specifies the transport protocol and the local servers known by this DTS entity.

maximum clock drift rate

Specifies the worst-case drift rate of the node's clock, in nanoseconds per second, as determined by the manufacturer's specifications.

maximum inaccuracy

Specifies the inaccuracy limit for the node. When the node exceeds the maximum inaccuracy setting, it attempts to synchronize.

Default: **0-00:00:00.100**

Value: **0-00:00:00.0 99 99:99:99.99**

next tdf change

Specifies the future time at which the time differential factor is automatically changed. The value is determined by the operating system.

query attempts

Specifies the number of attempts that a node makes to contact a server before the node considers the server unavailable.

Default: 3

Value: 1-10

server entry name

Specifies a server's ACL entry name; *hostname* represents the name of the system or node that is the server's client. The default setting is the recommended value.

Default: *./:/hosts/hostname/dts-entity*

server group name

Specifies the security group name for the time servers within the cell.

Default: *./:/subsys/dce/dts-servers*

server principal name

Specifies a server's principal name for authentication purposes; *hostname* represents the name of the system or node that is the server's client. The default setting is the recommended value.

Default: *./:/hosts/hostname/dts-entity*

servers required

Specifies the minimum number of servers required for a synchronization. Settings of 1 or 2 may cause unreliable computed times.

Default: 3

Value: 1-10

synchronization hold down

Specifies the interval a node must wait to synchronize. Also specifies synchronization frequency when a node reaches the value specified by the **maximum inaccuracy** characteristic.

show(8dts)

Clerks:

Default: **0-00:10:00.0**

Value: **0-00:00:30.0 - 01 00:00:00.00**

Servers:

Default: **0-00:02.00.0**

Value: **0-00:00:30.0 - 01 00:00:00.00**

time differential factor

Specifies the Time Differential Factor (TDF), which is the amount of time the server varies from Greenwich mean time (GMT) or Coordinated Universal Time (UTC).

Default: **0-00:00:00.000**

Value: **-13-00:00:00 - 13-00:00:00**

time provider present

Specifies whether or not the entity used an external time-provider at the last successful synchronization. This attribute applies to servers only.

time representation version

Specifies the timestamp format used by the node.

type

Specifies whether the node is a DTS server or clerk. The **create** command modifies this characteristic.

global set timeout

Specifies the amount of time the node waits for a response to a WAN synchronization request before sending another request or declaring a global server to be unavailable. The number of attempts made to reach the server is controlled by the **query attemps** characteristic.

Default: **0-00:00:15.000**

Value: **0-00:00:00.000 0-00:10:00.000**

Counters

clock settings

Specifies the number of times the node clock has been set nonmonotonically (abruptly).

creation time

Specifies the time at which the DTS entity was created and the counters were initialized.

different epochs detected

Specifies the number of times the node received time response messages from servers or clerks that had epoch numbers different from its own.

disable directives completed

Specifies the number of times DTS has been disabled.

enable directives completed

Specifies the number of times DTS has been enabled.

epoch changes completed

Specifies the number of times the server's epoch has changed.

faulty servers detected

Specifies the number of times a server has detected faulty servers (other than itself).

insufficient resources detected

Specifies the number of times the node has been unable to allocate virtual memory.

invalid messages detected

Specifies the number of times the local node failed to process a received message that had an unknown message type or incorrect length.

local faults detected

Specifies the number of times the node's time interval failed to intersect with the computed interval of the servers.

no global servers detected

Specifies the number of times the courier server could not contact any global servers.

protocol mismatches detected

Specifies the number of times the local node failed to process a received message containing an incompatible protocol version.

show(8dts)

servers not responding

Specifies the number of times the courier server could not contact a specific global server.

synchronizations completed

Specifies the number of times the node successfully synchronized.

system errors detected

Specifies the number of times a DTS operation detected a system error.

time provider failures detected

Specifies the number of times the external time-provider signaled a failure or the node was unable to access the time-provider.

time representation version mismatches detected

Specifies the number of times the local node failed to process a received message containing an incompatible timestamp format.

too few servers detected

Specifies the number of times a node failed to synchronize because it could not contact the required minimum number of servers.

too many servers detected

Specifies the number of times the responses to a synchronization request overflowed the assigned system buffer.

updates initiated

Specifies the number of times a server has attempted to update its clock.

Status

current time

Specifies the current time on the node.

global servers

Specifies the set of global servers known by the node.

last synchronization

Specifies the computed time at the last synchronization.

local servers

Specifies the set of local servers known by the node.

state

Specifies the state of the DTS entity.

off The DTS entity is disabled.

on The DTS entity is enabled.

synchronizing

The DTS entity is synchronizing.

updating The DTS entity is updating the time.

uid

Specifies the entity's unique identifier, which is generated when the entity is created.

Privilege Required

You must have read permission on the ACL associated with the DTS entity in order to execute the command.

Examples

1. The following command shows how to display the current time:

```
dtscp> show current time
```

```
Current Time = 1990-11-30-12:11:41.718-05:00I0.359 EST
```

2. The following command shows how to display all of the entity's characteristic attribute settings:

```
dtscp> show all
```

3. The following command shows how to display all of the local servers known to the entity:

```
dtscp> show local servers
```

Related Information

Commands: **set(8dts)**

synchronize(8dts)

synchronize

Purpose Causes the DTS entity to synchronize the clock in the system where the command is entered

Synopsis `synchronize [set clock boolean]`

Arguments

set clock *boolean*

Specifies whether the clock is abruptly set or gradually adjusted to the computed time. This argument is optional. The following values are valid:

false The clock is gradually adjusted. This is the default condition.

true The clock is abruptly set.

Description

The **synchronize** command causes the DTS clerk or server to solicit time intervals from servers, compute the intersection of the time intervals, and adjust the system clock to the midpoint of the computed time interval. This command overrides the functions of the **synchronization hold down** characteristic.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Notes The **synchronize** command does not execute if the entity is already synchronizing or is disabled; the entity must be in the **on** state.

Examples The following example shows how to initiate a synchronization for the entity, followed by a gradual clock adjustment:

```
dt.scp> synchronize
```

The following example shows how to initiate a synchronization for the entity, followed by an abrupt reset of the clock:

```
dt.scp> synchronize set clock true
```

unadvertise

Purpose Removes the global server entry from the cell profile

Synopsis **unadvertise**

Description

The **unadvertise** command causes DTS to remove the server's name from the cell profile and binding from the related CDS entry, deleting the server's global status.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Examples `dt:scp> unadvertise`

Related Information

Commands: **advertise(8dts)**

update(8dts)

update

Purpose Gradually adjusts the clock on the specified server node to the time specified by the argument

Synopsis `update time absolute-time`

Arguments

time *absolute-time*

Specifies the absolute time to which the clock is adjusted. This argument is required.

Description

The **update** command gradually adjusts the system clock to a new time, beginning at the time specified in the argument. The difference between the current clock value and the absolute time specified in the argument is used to adjust the clock.

Privilege Required

You must have write permission on the ACL associated with the DTS entity in order to execute the command.

Notes The **update** command is valid only for servers. The combined time and inaccuracy value you specify must be contained within the interval formed by the current time and inaccuracy. That is, the new setting must be more accurate than the current time.

Examples The following example shows how to update the time for a server; the clock is gradually adjusted to the specified time:

```
dtscp> update time 1990-12-30-11:24:00.000-05:00I0.000
```

Chapter 4

Security Service Commands

intro(5sec)

intro

Purpose Introduction to the DCE Security administrative files

Description

This section describes DCE Security files for system administration. These files are

passwd_override

Contains override entries that let you override password, GECOS, home directory, and shell entries in the registry database for a local machine.

v5srvtab

Contains server machine passwords on the local machine.

See the file's reference page for further information on each file.

passwd_override

Purpose The registry database override file

Synopsis *dcelocal/etc/passwd_override*

Description

The *dcelocal/etc/passwd_override* administrative file lets you override UNIX ID, group ID, password, GECOS, home directory, and shell entries from the registry database.

The **passwd_override** file is stored on each host machine. Any changes you make to it are in effect for the local machine only, and have no effect on the centralized registry.

You may find **passwd_override** especially useful for excluding people from using certain machines, establishing local root passwords, or tailoring local user environments.

The **passwd_override** File Format

The format of the **passwd_override** entries is similar to entries in the UNIX password file. The format is

```
principal:passwd:UNIX_id:grp_id:GECOS:home_dir:login_shell
```

Field Descriptions

In an override entry, the *principal*, *unix_id*, and *grp_id* fields are keyfields. You must enter one of them to identify the principal or group to which the overrides apply. The keyfield is used to perform a lookup in the override file. The lookup is performed in order as the entries are specified in an override entry: first by principal name, then UNIX ID, and finally group ID. If you specify more than one keyfield in an override entry, the first keyfield specified is used as the lookup key; subsequent keyfields are used as overrides.

Each of the entries in the **passwd_override** file is described below.

principal A principal name that identifies the account to which the overrides apply.

passwd The UNIX encrypted password. You can specify an override in this field. If you do, the password you enter here will be in effect for this local machine only. For no override, leave the field empty.

passwd_override(5sec)

When you override a principal's password, only the principal's local credentials are obtained at login, not the principal's network credentials. Without network credentials, the principal cannot access the network registry and obtain the information normally provided at network login. Therefore, you must supply all this information in the **passwd_override** file entry. For overrides to passwords, you must enter all fields in the override entry, including all keyfields.

You can also specify **OMIT** in the *passwd* field to disallow login on the local machine. The use of **OMIT** in conjunction with an option to the **passwd_export** command also prevents the inclusion of this user in the password file created by **passwd_export**. (See the section on using **OMIT**, later in this reference page, for details.)

UNIX_id The UNIX user ID. You must enter a UNIX ID, group ID, or principal name to identify the account. Enter a UNIX ID when you want to apply the overrides to all accounts for a principal (those identified by the principal's primary name and any aliases). UNIX IDs are especially useful for overrides to **root**. If, for example, **root** has an alias of **wizard**, an override keyed by principal name applies only when **root** logs in as **root**, not when **root** logs in as **wizard**. If the override is keyed by **root**'s UNIX ID, it applies when **root** logs in as **root** or **wizard** or any other alias.

Note that if you supply a principal name and a UNIX ID, the UNIX ID is used only to override the network UNIX ID, not a keyfield for lookup.

grp_id The UNIX group ID that identifies the group to which the override applies. Enter a only a UNIX group ID to apply the override to all members of the group. Enter a UNIX group ID and a principal name or principal UNIX ID to override the principal's group ID.

GECOS The account's GECOS field. You can specify an override in this field. To keep it unchanged, leave it empty.

home_dir The account's home directory. You can specify an override in this field. To keep it unchanged, leave it empty.

login_shell The account's log-in shell. You can specify an override in this field. To keep it unchanged, leave it empty.

The *principal*, *unix_id*, and *grp_id* fields are keyfields. You must enter one of them to identify the principal or group to which the overrides apply. The keyfield is used to perform a lookup in the override file. The lookup is performed in order first by principal name, then UNIX ID, and finally group ID. If you specify more than one

of these fields in an override entry, the principal name is used as the lookup key and the UNIX ID and group ID are used as overrides.

Using OMIT

If you enter either the word **OMIT** or another invalid password string (such as * (asterisk) or **NO GOOD**) in the *passwd* field, the user (or set of users) will be unable to log in to the local machine. If you specify **OMIT** and run **passwd_export** with the **-x** option, the named user (or set of users) will not appear in the */etc/passwd* file produced by **passwd_export**.

You should also be aware that, if you have omitted users from the */etc/passwd* file, information about those users will not be available to any programs that use the password file. For example, the **ls -l** and the **finger** commands both access the password file to obtain further information about a user identified by a user ID. If the user is omitted, no password entry will exist and no information will be available. For this reason, you should use **OMIT** to omit users from the */etc/passwd* file only if your user community is very large and either of the following conditions occur:

- The **passwd** file is taking up too much space.
- User-ID-to-name mapping is too slow (during **ls -l**, for example).

Notes

Users can update their entries in the override file for the local host by using **chpass**. Refer to the **chpass** reference page for details.

Examples

To prevent users with a UNIX ID of 25 from logging in to a node, the entry in the **passwd_override** file would be as follows:

```
:exclude:25:::
```

To prevent users in the **evil_users** group from logging in to a node and to omit them from inclusion in the password file: put **OMIT** in the *passwd* field:

```
:OMIT::25:::
```

Then run the following **passwd_export** command with the **-x** option to cause these users to be omitted from */etc/passwd* file

```
dcelocal/etc/passwd_export -x
```

To change the password, home directory, and initial shell for **mozart**'s account, the entry would be as follows:

```
mozart:sq1Rc1Urrb1L6:678:893:Wolfgang A. Mozart:/aria/wolfgang:/bin/csh
```

passwd_override(5sec)

Related Information

Commands: **login(1)**, **crypt(1)**, **chpass(1)**, **finger(1)**, **adduser(8)**, **rgy_edit(8sec)**,
passwd_export(8sec)

Functions: **getpwent(3)**

Files: **group(5)**

/krb5/v5srvtab

Purpose The server and machine keytab file

Description

The **/krb5/v5srvtab** file is a file on the local node created by the **rgy_edit** command, the **sec_create_db** command, or any application that makes **sec_key_mgmt()** calls. The file contains passwords for servers and machine accounts. To view or manipulate the contents of this file, use the **sec_key_mgmt** API, described in the *OSF DCE Application Developer's Guide* and the *OSF DCE Application Developer's Reference*.

Related Information

Books: *OSF DCE Application Developer's Guide*, *OSF DCE Application Developer's Reference*.

intro

Purpose Introduction to the DCE Security administrative commands

Description

This section describes DCE Security commands for system administration. These commands are

passwd_export

Updates local password and group files from DCE registry data

passwd_import

Creates DCE registry entries based on password and group file entries

passwd_override

Establishes DCE registry overrides for a principal on a local node

rgy_edit

Edits the registry database

sec_admin

Administers the Security Server

sec_clientd

The client daemon that must run on each machine that is a client of the registry

sec_create_db

The registry creation utility

secd

The Security Server daemon

See the command's reference page for further information on each command.

passwd_export

Purpose Creates local password and group files

Synopsis `dcshared/bin/passwd_export [-d dir_name] [-m max_entries] [-h[elp]] [-n] [-s] [-v] [-x]`

The `dcshared/bin/passwd_export` command creates local password and group files from registry data. These files are used when the network registry is unavailable and by programs that use the original UNIX `passwd` and `group` interfaces instead of the DCE interfaces. Use **passwd_export** to keep these local files consistent with the registry database.

Options

- d *dir_name*** Specifies the name of the directory to store the password, group, and organization files created by **passwd_export**. If you do not enter a directory name, the files are stored in the `/etc` directory.
- h[elp]** Displays help information.
- n** Specifies that **passwd_override** file entries should be ignored. Without this option, **passwd_export** applies the override entries to the local password and group files it creates.
- m *max_entries*** Specifies the maximum number of entries that can be stored in the local files.
- s** Specifies that the entries in the local file should be sorted by UNIX number. If you do not specify this option, the entries are in order as they are retrieved from the registry.
- v** Specifies that the command should run in verbose mode.
- x** Do not create entries for users with **OMIT** as their encrypted password in the **passwd_override** file. See the **passwd_override** reference page for further details about omitting users.

Description

When **passwd_export** runs, it makes backup copies of the current password and group files, if they exist. The files are named, respectively, **passwd.bak** and **group.bak**.

passwd_export(8sec)

By default, the backups are stored and the new files created in the */etc* directory. You can override the default by supplying a directory name with the **-d** option.

Running passwd_export

The **passwd_export** command is commonly run with an entry in */usr/lib/crontab*. For example, to update the files every hour, the entry is

```
0 * * * * dceshared/bin/passwd_export
```

In large network environments, it is a good idea to stagger the times **passwd_export** is run.

Related Information

Files: **passwd_override(5sec)**

passwd_import

Purpose Creates registry database entries based on information in UNIX group and password files

Synopsis `dcshared/bin/passwd_import [-c] -d pathname [-i] [-o org] [-p password] [-u username] [-v]`

Options

- c** Run in check mode: process the command, showing all conflicts, but make no requests for resolution.
- d *pathname*** The *pathname* argument is the path to the directory containing the foreign password and group files to be imported.
- i** Names are not in conflict, but represent the same identity.
- o *org*** The *org* argument is the name of an organization to be assigned to all imported entries.
- p *password*** The password for the account with whose privileges **passwd_import** will run.
- u *username*** The principal name of the account with whose privileges **passwd_import** will run. This account must have the privileges to access the registry and add principals, groups, accounts, and members to groups. This principal name and password are used to obtain network authentication. If you do not supply them, **passwd_import** prompts for them, even if you have already performed a network login.
- v** Run in verbose mode: generate a verbose transcript of **passwd_import** activity.

Description

The **passwd_import** command is a mechanism for creating registry database entries that are consistent with foreign password and group file entries.

Use **passwd_import** to ensure consistency between DCE and foreign protection mechanisms when you do the following:

- Attach DCE node(s) to an existing UNIX network

passwd_import(8sec)

- Attach UNIX node(s) to a DCE network
- Connect DCE and UNIX networks

If the password and group file entries do not exist in the DCE registry, **passwd_import** creates them. If there are duplicate entries, **passwd_import** follows your directions on how to handle them.

The Process

The DCE registry database must exist and be running before you can use **passwd_import**. If you are simply adding a few DCE nodes to a foreign network, you can create a new, but empty, registry to meet this requirement.

As **passwd_import** processes, it performs the following steps:

1. Examines the group file and creates group entries in the registry.
2. Examines the **passwd** file and creates principal and account entries in the registry. The organization assigned may be specified as input to **passwd_import**.
3. Reexamines the group file and creates membership lists.

Conflicts

During this process, **passwd_import** can find conflicts in name strings (for example, in the password file, **joe 102**; in the registry database, **joe 555**) and in UNIX IDs (for example, in the password file, **joe 102**; in the DCE, **carmelita 102**). When **passwd_import** finds a conflict, it prompts for changes to make to the **/etc/passwd** and **/etc/group** entries. No changes are made to the registry entries. In other words, all conflicts are resolved in favor of the registry entry.

The **-i** option specifies that duplicate names are not in conflict but, in fact, represent the same identity. Therefore, when duplicate names arise, no action is necessary. If you do not use the **-i** option, **passwd_import** prompts for how to handle the name conflicts.

Resolving Conflicts

The **passwd_import** command prompts for instructions to resolve the conflicts it finds. You have the following choices:

- You can create an alias to resolve a UNIX ID conflict. This action creates an alias for the registry object in conflict. This alias is assigned the same name as the conflicting entry in the **/etc/group** or **/etc/passwd** file. For example, if the entry **joe 555** exists in the registry and the entry **tim 555** exists in the **/etc/passwd** file, this option creates the alias **tim** for **joe 555**.

- You can generate a new UNIX ID automatically or enter a new one explicitly to resolve a UNIX ID conflict. For example, if there is a conflict between the entry **joe 555** in the registry and **tim 555** in the **/etc/passwd** file, you can generate a new UNIX ID for **tim**.
- You can enter a new name to resolve a name conflict. For example if there is a conflict between the entry **joe 555** in the registry and **joe 383** in the **/etc/passwd** file, you can generate a new name for **joe 383**. This new name will then be added to the registry.

In addition, you are given the option of ignoring the conflict and skipping this entry.

Registry Database Entries

New registry entries created by **passwd_import** are assigned the following values:

For Principal and Group Entries:

fullname A blank string; no fullname is added for the entry.

alias/primary

If the **/etc/passwd** file contains two entries with the same UNIX number, **passwd_import** creates a primary name entry for the first occurrence of the UNIX number and alises entries for each subsequent occurrence.

projlist_ok Yes (for groups only).

membership list

For new groups only, all principals listed in the group file, and all principals with accounts in the password file with that group.

For Account Entries:

account_valid

False.

gecos Same as password file.

homedir Same as password file.

Same as password file.

passwd Randomly generated. Note that you must modify or reset randomly generated passwords before user authentication is possible.

passwd_import(8sec)

passwd_dtm Date and time **passwd_import** was run.

passwd_valid False.

account expiration date

None.

Server flag True.

Client flag True.

Postdated ticket flag

False.

Forwardable ticket flag

True.

TGT authentication flag

True.

Renewable ticket flag

True.

Proxiable ticket flag

False.

Duplicate ticket flag

False.

Good since date

Time of account creation.

Maximum ticket renewable

Default to registry authentication policy.

Maximum ticket lifetime

Default to registry authentication policy.

Related Information

Commands: **rgy_edit(8sec)**, **sec_admin(8sec)**, **secd(8sec)**

rgy_edit

Purpose Edits the registry database

Synopsis *dcshared/bin/rgy_edit* [-a | -g | -l | -o | -p | -s | -v]

Options The following options are supplied when **rgy_edit** is invoked. You can specify only one of the options **-a**, **-p**, **-g**, and **-o**. If you specify the **-l** option, you can specify no other options.

- a** (default) Edits or views accounts.
- g** Edits or views groups.
- l** Edits or views entries in local registry.
- o** Edits or views organizations.
- p** Edits or views principals.
- s** Uses the specified registry site.
- v** Views selected entries.

Description

The **rgy_edit** tool views and edits information in the registry database. You can invoke **rgy_edit** from any node.

You can edit and view names, accounts, and policies in the network registry (the default) or perform a subset of those functions on the local registry (using the **-l** option). Changes made by **rgy_edit** apply only to the registry. They do not apply to the local override file or the local password and group files, both of which can be edited manually. You can view and change only those registry objects to which you are granted the appropriate permissions.

Invoking rgy_edit

When you invoke **rgy_edit**, it displays the following prompt:

```
rgy_edit=>
```

At this prompt, you can enter any of the **rgy_edit** subcommands, and **rgy_edit** will prompt you for the required information. Alternatively, you can enter the

rgy_edit(8sec)

subcommand followed by all the required information. The **rgy_edit** command may prompt you for any required information you do not enter.

Subcommands

Principal, Group, and Organization Subcommands

v[iew] [*name | unix_number*] [-f] [-m] [-po]

Views name entries.

Whether the name applies to a principal, group, or organization depends on the domain in which you run **rgy_edit**. Use the **do[main]** subcommand (described later in this reference page) to change domains.

If you specify *unix_number*, **rgy_edit** displays all matching entries, including any aliases.

The **-f** option displays entries in full (all fields except the membership list and organization policy).

If you are viewing groups or organizations, **-m** displays the membership list. For principals, **-m** lists all groups of which the principal is a member, including groups that cannot appear in a project list.

If you are viewing organizations, **-po** displays policy information. If you do not enter the **-po** option, **rgy_edit** shows only the organization's name and the UNIX number.

a[dd] [*principal_name unix_number [fullname]*] [-al] [-q quota]]

a[dd] [*group_name unix_number [fullname]*] [-nl]]

a[dd] [*organization_name unix_number [fullname]*]

Creates a new name entry.

If you do not specify a *principal*, *group*, or *organization* name, the **add** subcommand prompts you for each field in the entry. If you are adding organizations, the command prompts you for policy information as well. If you specify only a *principal*, *group*, or *organization* name and no other arguments, the object's fullname defaults to "" (that is, blank), the object's UNIX number is assigned automatically, and the object's creation quota defaults to unlimited.

For principals, the **-al** option creates an alias entry. If *unix_number* is already assigned to a principal and you do not specify **-al**, an error occurs and you must either choose a different *unix_number* or specify **-al**. If you use **-al** to create an alias and *unix_number* is not already associated with a primary name, **rgy_edit** issues a warning

but creates the alias. The **-q** option specifies the total number of registry objects that can be created by the principal.

For groups, the **-nl** option indicates that the group is not to be included on project lists; omitting this option allows the group to appear on project lists.

Use quotation marks to embed spaces (or quotation marks) in a *fullname*. A single space between quotation marks indicates a null *fullname*.

c[hange] [*principal_name* [-**n** *name*] [-**f** *fullname*] [-**al** | -**pr**] [-**q** *quota*]]

c[hange] [*group_name* [-**n** *name*] [-**f** *fullname*] [-**nl** | -**l**]

c[hange] [*org_name* [-**n** *name*] [-**f** *fullname*]]

Changes a principal, group, or organization.

If you do not specify a *principal_name*, *group_name*, or *org_name*, the **change** subcommand prompts you for a name. If you do not specify any fields, the subcommand prompts you for each field in succession. To leave a field unchanged, press <RETURN> at the prompt. If you are changing organization entries in the interactive mode, the subcommand prompts you for policy information as well.

Use **-n name** and **-f fullname**, to change a name entry's principal, group, or organization name, and fullname, respectively.

For all entries, the **-al** option changes a primary name into an alias, and the **-pr** option changes an alias into a primary name. This change can be made only from the command line, not in the interactive mode.

The **-q** option specifies the total number of registry objects that can be changed by the principal.

For group entries, the **-nl** option disallows the group from appearing in project lists, while the **-l** option allows the group to appear in project lists.

For organization entries, you can change policy information only in the interactive mode.

A single space between quotation marks indicates a **nil** *fullname*.

Changes to a principal name are reflected in membership lists that contain the principal name. For example, if the principal **ludwig** is a member of the group **composers** and the principal name is changed to **louis**, the membership list for **composers** is automatically changed to include **louis** but not **ludwig**.

rgy_edit(8sec)

The only fields of reserved entries that you can change are the *fullname* and (for *groups*) the property that allows a *group* to appear in project lists.

m[ember] [*group_name* | *org_name* [-**a** *member_list*] [-**r** *member_list*]]

Edits the membership list for a group or organization.

If you do not specify a group or organization, the **member** subcommand prompts you for names to add or remove.

To add principal names to a membership list, use the **-a** option followed by the principal names (separated by commas) to be added. To delete principal names from a membership list, use the **-r** option followed by the principal names (separated by spaces) to be removed. If you do not include either option on the command line, **rgy_edit** prompts you for names to add or remove.

Adding a principal to a membership list permits creation of a login account for that principal with that group or organization.

Removing *principal* from the membership list for a group or organization has the side effect of deleting all login accounts for *principal* that are associated with the group or organization (and, of course, eliminating any permissions granted to the principal as a result of the group membership the next time the principal's ticket-granting ticket is renewed).

del[ete] {-**p** *pname* | -**g** *gname* | -**o** *oname*}

Deletes a name entry.

If you delete a principal name, **rgy_edit** deletes the principal's account. If you delete a group or organization, **rgy_edit** deletes any accounts associated with the group or organization. You cannot delete reserved names.

adopt uuid principal_name [-**u** *unix_number*] [-**f** *fullname*]

[-**q** *object_creation_quota*]

adopt uuid group_name [-**u** *unix_number*] [*fullname*]] [-**nl**]

adopt uuid org_name [-**u** *unix_number*] [-**f** *fullname*]]

Creates a primary name entry for the specified UUID.

The UUID must be an orphan (a UUID for which no name exists in any domain). The *uuid* is hexadecimal numbers in RPC string format.

An error occurs if you specify a name or UNIX number that is already defined within the same domain of the database.

A single space between quotation marks indicates a nil *fullname*.

Account Subcommands

v[iew] [*name* | *unix_number*] [-f]

Displays login accounts.

Without the **-f** option, **view** displays only the user fields in each account entry: principal, group, organization, encrypted password, miscellaneous information, home directory, and login shell.

With **-f**, **view** displays the full entry, including the administrative fields as well as the user fields. Administrative information includes who created the account, when it was created, who last changed it, when it was last changed, when it expires, whether it is valid, whether the password is valid, and when the password was last changed.

a[dd] [*pname*] [-g *gname*] [-o *oname*]
 [[-rp | -pw *password*] -mp *password*]
 [-m *misc*] [-h *homedir*] [-s *shell*]
 [-pnv | -pv] [-x *account_exp* | none] [-anv | -av]
 [[-ena[ble] *option* | -dis[able] *option*]...]
 [-gsd *date_and_time*] [-mcr *lifespan*] [-mcl *lifespan*]

Creates a login account.

The optional *pname* argument specifies the principal for whom the account should be created. The **-g** and **-o** options specify the principal's group and organization. If the principal specified in *account* is not already a member of the specified group and organization, **rgy_edit** automatically attempts to add the principal to the membership lists. If you do not have the appropriate permissions for the group and organization, the attempt will fail and the account will not be created.

The **-rp** option generates a random password for the account. The primary use of this option is to create passwords for accounts that will not be logged into (since the random password can never be supplied.) The **-pw** option is used to supply a password for the account on the command line.

If you use the **-rp** option or the **-pw** option, you must also use the **-mp** option to supply your password so your identity can be validated.

rgy_edit(8sec)

If you do not specify the **-rp** option or the **-pw** option, **rgy_edit** prompts for the account's password twice to ensure you did not make a typing mistake. Then it prompts for your password to verify your identity. You can supply an encrypted *password* for the account's new password. If you supply one in plain text, the system performs the encryption. The format of *password* must adhere to the policy of the associated organization or the policy of the registry as a whole, whichever is more restrictive.

The information supplied with the **-m** option is used to create the GECOS field for the account in the */etc/passwd* file. If you run the **passwd_export** command, this entry contains that concatenation of the principal's full name and the information specified with the **-m** option. Use quotes to include spaces, hyphens, or quotes in *misc*.

The **-h** option specifies the pathname of the principal's home directory. The default *homedir* is */*. The **-s** option specifies the pathname of the principal's login shell. The default *shell* is the null string.

Use a single space between quotation marks to indicate a nil *password*, *misc_info*, *homedir*, or *shell*.

The **-pnv** (password not valid) option specifies that the password is invalid and at the next login (for a newly created account, the first login), the user must change the password. The **-pv** option indicates the password is currently valid and need not be changed (the default).

The **-x** option sets an expiration date for the account; the default is "none," meaning that the password will never expire.

The **-anv** (account not valid) option specifies that the account is not currently valid for login. The **-av** option indicates the account is currently valid (the default).

The **-enable** and **-disable** options set or clear the following options:

- | | |
|-----------------|---|
| c[lient] | Allows the principal to act as a client and log in, acquire tickets, and be authenticated. If you disable client , the principal cannot act as a client. The default is enabled. |
| s[erver] | Allows the principal to act as a server and engage in authenticated communication. If you disable server , the principal cannot act as a server that engages in authenticated communication. The default is enabled. |

po[stdated] Allows tickets with a start time some time in the future to be issued to the account's principal. The default is disabled.

f[forwardable]

Allows a new ticket-granting ticket with a network address that differs from the present ticket-granting ticket address to be issued to the account's principal. The default is enabled.

pr[oxiable] Allows a new ticket with a different network address than the present ticket to be issued to the account's principal. The default is enabled.

T[GT_authentication]

Allows service tickets issued to the account's principal to use the standard DCE ticket-granting ticket authentication mechanism. The default is enabled.

r[enewable] Allows tickets issued to the principal's ticket-granting ticket to be renewed. If this option is enabled, the DCE Security Service allows the principal's ticket-granting ticket to be renewed for as long as its lifetime is valid. The default is enabled.

dup[_session_key]

Allows tickets issued to the account's principal to have duplicate keys. The default is disabled.

The **-gsd** (good since date) option is the date and time the account was last known to be valid. When accounts are created, this date is set to the account creation time. If you change the good since date, any tickets issued before the changed date are invalid. Enter the date in yy/mm/dd.hh:mm format.

The **-mcr** (maximum certificate renewable) option is the number of hours before a session with the principal's identity expires and the principal must log in again to reauthenticate. The default is 4 weeks.

The **-mcl** (maximum certificate lifetime) option is the number of hours before the Authentication Service must renew a principal's service certificates. This is handled automatically and requires no action on the part of the principal. The default is 1 day.

c[hange] -p pname [-g gname] [-o oname]
[-np pname] [-ng gname] [-no oname]
[[[-rp | -pw password] mp password]

rgy_edit(8sec)

[-m misc] [-h homedir] [-s shell]
[-pnv | -pv] [-x account_exp | none] [-anv | -av]
[[-ena[ble] option | -dis[able] option]...]
[-gsd date_and_time] [-mcr lifespan] [-mcl lifespan]

Changes an account.

The **-p**, **-g**, and **-o** options identify the account to change. The **-np**, **-ng**, and **-no** options change the account's, principal, group, and organization, respectively.

All options have the same meaning as described in the **add** command for accounts. Note that the **-rp** option can be used to change the random passwords of the reserved accounts created by **sec_create_db** when the registry database is created.

del[ete] -p pname [-g gname] [-o oname]

Deletes the specified account.

Enter the **-p** option to delete the specified principal's account. Enter the **-g** or **-o** option to delete accounts associated with the specified group or organization. If you enter the **-g** or **-o** option, **rgy_edit** prompts individually for whether to delete each account associated with the group or organization.

ce[ll] cellname [-ul unix_num] [-uf unix_num] [-gl gname] [-ol oname]
[-gf gname] [-of oname] [-mp passwd]
[-fa name] [-fp passwd]
[-m misc_info] [-x account_expiration_date]

Creates a cross-cell authentication account in the local and foreign cells.

This account allows local principals to access objects in the foreign cell as authenticated users and vice versa. The administrator in the foreign cell must have also set up a standard account, whose ID and password the administrator of the foreign cell must supply to you.

The *cellname* variable specifies the local account's principal name.

The **-ul** option specifies the UNIX number for the local cell's principal. The **-uf** option specifies the UNIX number for the foreign cell's principal. If you do not specify these UNIX numbers, they are generated automatically.

The **-gl** and **-ol** options specify the local account's group and organization. The **-gf** and **-of** options specify the foreign account's group and organization.

The **-mp** option specifies the password of the person who invoked **rgy_edit**.

The **-fa** option specifies the name identifying the account in the foreign cell, and the **-fp** option specifies the account's password.

The information supplied with the **-m** option is used to create the GECOS field for the account in the **/etc/passwd** file. If you run the **passwd_export** command, this entry contains that concatenation of the principal's full name and the information specified with the **-m** option. Use quotes to include spaces, hyphens, or quotes in *misc*.

The **-x** option specifies the account expiration date for both the local and foreign accounts. The default for this option is "none."

Note that the object creation quota for the local account defaults to 0 (zero), meaning that principals in the foreign cell cannot create objects in the local cell. You can change this with the **rgy_edit change** subcommand.

Key Management Subcommands

The key management subcommands must be run in command-line mode.

kta[dd] -p principal_name [-pw password] [-a[uto]] [-r[egistry]] [-f keyfile]

Creates a password for a server or machine in the keytab file on the local node.

The **-p** option specifies the name of the server or machine principal for which you are creating a password.

The **-pw** option lets you supply the password on the command line. If you do not enter this option or the **-auto** option, **ktadd** prompts for the password.

The **-a** option generates the password randomly. If you use this option, you must also use the **-r** option. If you do not specify this option, you are prompted for a password. The **-r** option sets the principal's password in the registry to the same password set in the keytab file.

rgy_edit(8sec)

The **-f** option specifies the name of the server keytab file on the local node to which you are adding the password. If you do not specify a keytab file name, **/krb5/v5srvtab** is used. Note that you must be root to add entries in the default keytab file.

ktl[ist] [-p *principal_name*] [-f *keyfile*]

Displays principal names and password version numbers in the local keytab file.

The **-p** option specifies the name of the server or machine principal for which you are displaying passwords.

The **-f** option specifies the name of the server keytab file on the local node for which you want to display entries. If you do not specify a keytab file name, **/krb5/v5srvtab** is used.

ktl[delete] -p *principal_name* -v *version_number* [-f *keyfile*]

Deletes a sever or machine principal's password entry from a keytab file.

The **-p** option specifies the name of the server or machine principal for whom you are deleting a password entry.

The **-v** option specifies the version number of the password you want to delete. Version numbers are assigned to a principal's password whenever the principal's password is changed. This allows any servers or machines still using tickets granted under the old password to run without interruption until the ticket expires naturally.

The **-f** option specifies the name of the server keytab file on the local node from which you want to delete passwords. If you do not specify a keytab file name, **/krb5/v5srvtab** is used. Note that you must be root to delete entries in the default keytab file. You must have the appropriate access rights to delete entries in other keytab files.

Miscellaneous Commands

do[main] [p | g | o | a]

Changes or displays the type of registry information being viewed or edited.

You can specify **p** for principals, **g** for groups, **o** for organizations, or **a** for accounts. If you supply no argument, **rgy_edit** displays the current domain.

s[ite] [[*name*] | -u[*pdate*] | -q[*uery*]]

Changes or displays the registry site being viewed or edited.

The *name* variable is the fully qualified name of the cell that contains the registry to which you want access. If you supply no argument, **rgy_edit** displays the current site.

The **-u** option indicates you want to edit the registry. **-q** indicates that you want to only view the registry.

prop[erties]

Changes or displays registry properties.

This command prompts you for changes. Press **<Return>** to leave information unchanged.

po[licy]

Changes or displays registry standard policies. (Note that policies for an organization are set with the **change** command in the organization domain.

This command prompts you for changes. Press **<Return>** to leave information unchanged.

au[th_policy]

Changes and/or displays registry authentication policies.

This command prompts you for changes. Press **<Return>** to leave information unchanged.

def[aults]

Changes or displays the the home directory, login shell, password valid option, account expiration date, and account valid option default values that **rgy_edit** uses.

When you run the **defaults** subcommand, it changes the defaults for that **rgy_edit** session and saves them in **~/rgy_editrc**. The newly saved defaults are used the next time you run **rgy_edit**.

h[elp] [command]

Displays usage information for **rgy_edit**.

If you do not specify a particular command, **rgy_edit** lists the available commands.

q[uit]

Exits **rgy_edit**.

e[xit]

Exits **rgy_edit**.

rgy_edit(8sec)

l[ogin]

Lets you log into a local machine and assume a new local identity from within the **rgy_edit** session.

The **rgy_edit** login command prompts for a principal name and password.

sc[ope] [name]

Limits the scope of the information displayed by the **view** subcommand to the directory (specified by *name*) in the registry database.

Valid Commands for the Local Registry

To edit or view the local registry, invoke **rgy_edit** with the **-l** option while you are logged into the machine whose local registry you want to maintain. This section lists the commands that are valid for editing or viewing the local registry. Only the subcommands and options listed here are valid for maintaining the local registry. Those not listed cannot be run against the local registry.

v[iew]

Displays local registry entries.

del[ete] *principal_name*

Deletes the account and credential information for *principal_name* from the local registry.

pu[rge]

Purges expired local registry entries.

pr[operties]

Changes and/or displays local registry properties and policies.

h[elp] [command]

Displays usage information for **rgy_edit**.

q[uit]

Exits **rgy_edit**.

sec_admin

Purpose Security administrative tool

Synopsis *dcshared/bin/sec_admin*

Description

The **sec_admin** command is the Security Server administration tool. Using **sec_admin**, you can

- Generate a new master key used to encrypt principal passwords
- Stop **secd**, the Security Server
- Put the master registry into and out of the maintenance state

Note that **sec_admin** cannot add, delete, or modify registry information, such as names and accounts; use **rgy_edit** to modify registry entries.

Subcommands

Once invoked, **sec_admin** enters an interactive mode in which it accepts the following subcommands:

mas[ter_key] The **master_key** subcommand generates a new master key and re-encrypts all passwords in the registry database using this key. The new master key is derived from a randomly generated number.

stop The **stop** subcommand stops the Security Server (**secd**).

sta[te] -maintenance | -service

Puts the master server into maintenance state or takes it out of maintenance state.

The **-maintenance** option causes the master server to save its database to disk and refuse any updates.

The **-service** option causes the master server to reload its database from disk, return to its normal "in service" state, and (if its database and replica list are writable) start accepting updates.

sec_admin(8sec)

- h[elp]** Lists the **sec_admin** subcommands and shows their allowed abbreviations.
- q[uit]** The **quit** subcommands ends the **sec_admin** session.

sec_clientd

Purpose Security client daemon

Synopsis *dcelocal/bin/sec_clientd* [-d[ebug]] [-p[urge]] [-v[erbose]]

Options

- d[ebug] Runs in debug mode. In debug mode, **sec_clientd** runs as a foreground process.
- p[urge] Purges the existing machine content.
- v[erbose] Runs in verbose mode.

All options start the Security Client on the local node.

Description

The **sec_clientd** daemon must run on all machines that use the services of the Security Server and access the network registry. In addition to acting as the client side of the Security Server, **sec_clientd** logs in the local machine at invocation and automatically updates the keytab file that contains the local machine's password. (The default keytab file is */krb5/v5srvtab*.) It changes the machine's registry account password at the intervals required by the registry. The password is randomly generated. You must have root privileges to invoke **sec_clientd**.

A Remote Procedure Call daemon (**rpcd**) must be running on the local node when **sec_clientd** is started. Typically, both **rpcd** and **sec_clientd** are started at boot time. The **sec_clientd** daemon places itself in the background when it is ready to service requests.

Examples To start the Security Client Server, become root and enter

```
$ dcelocal/bin/sec_clientd
```

Related Information

Commands: **secd(8sec)**, **rpcd(8rpc)**

sec_create_db

Purpose Registry database creation utility

Synopsis *dcshared/bin/sec_create_db -my[name] my_server_name*
[-cr[erator] creator_name] [-cu[nix_id] creator_unix_id]
[-g[roup_low_id] g_unix_id] [-k[eyseed] keyseed]
[-ma[x_unix_id] max_unix_id] [-o[rg_low_unix_id] o_unix_id]
[-pa[ssword] default_password] [-p[erson_low_unix_id] p_unix_id]
[-u[uid cell_uid] [-v[erbose]]]

Options

- my[name]** Specifies the name that will be used by the Directory Service to locate the machine on which the cell's Security Server is running.
- cr[erator]** Specifies the principal name of the initial privileged user of the registry database (known as the "registry creator").
- cu[nix_id]** Specifies the UNIX ID of the initial privileged user of the registry database. If you do not enter the UNIX ID, it is assigned dynamically.
- g[roup_low_unix_id]**
Specifies the starting point for UNIX IDs automatically generated by the Security Service when groups are added with the **rgy_edit** command.
- k[eyseed]** Specifies a character string used as the initial keyseed for the master registry.
- ma[x]** Specifies the highest UNIX ID that can be assigned to a principal, group, or organization.
- o[rg_low_unix_id]**
Specifies the starting point for UNIX IDs automatically generated by the Security Service when organizations are added with the **rgy_edit** command.
- pa[ssword]** The default password assigned to the accounts created by **sec_create_db**, including the account for the registry creator. If you do not specify a default password, **-dce-** is used. (Note that the **hosts/local_host_name/principal_name none none**, **krbtgt/cell_name none none**, and **nobody none none** accounts are not assigned the default password, but instead a randomly generated password.)

- p[erson_low_unix_id]** Specifies the starting point for UNIX IDs automatically generated by the Security Service when principals are added with the **rgy_edit** command.
- u[uid]** Specifies the cell's UUID. If you do not enter this UUID, it is assigned dynamically.
- v[erbose]** Specifies that **sec_create_db** runs in verbose mode and displays all activity.

Description

The **sec_create_db** tool creates a new registry database on the local node in *dcelocal/var/security/rgy_data*. The database is initialized with names and accounts, some of them reserved. The **sec_create_db** utility ordinarily needs to run only once in a cell. You must be root to invoke **sec_create_db**.

When the registry database is created, default ACL entries for registry objects are also created. These entries give the most privileged permission set to the principal named in the **-cr[erator]** option. If the principal is not one of the reserved names and accounts, **sec_create_db** adds it as a new principal and adds an account for that new principal. If the **-cr** option is not used, root is the creator.

The **sec_create_db** command also creates a registry configuration file, named *dcelocal/etc/security/pe_site*, that contains the network address of the machine on which the database is created. This file supplies the binding address of the **secd** server if the Naming Service is not available.

Files

/dcelocal/etc/security/pe_site

The file containing the network address of the machine on which the security database is created.

/dcelocal/var/security/rgy_data

The directory in which the registry database files are stored.

Related Information

Commands: **secd(8sec)**

secd(8sec)

secd

Purpose The DCE Security Server

Synopsis *dcelocal/bin/secd* [-b[ootstrap]] [-lockpw] [-locksm[ith]] [*pname*] [-rem[ote]] [-v[erbose]]

Options -locksm[ith] Restarts a registry server in locksmith mode. Use this mode if you cannot access the registry as the principal with full registry access, because that principal's account has been inadvertently deleted or its password lost.

pname The *pname* argument is the name of the locksmith principal. If no registry account exists for this principal, **secd** creates one.

-lockpw Prompts for a new locksmith password when running in locksmith mode. This option allows you to specify a new password for the locksmith account when the old one is unknown.

-rem[ote] Allows the locksmith principal to log in remotely. If this option is not used, the principal must log in from the local machine on which **secd** will be started.

-bo[otstrap] Always waits only one minute between tries to export binding information to the Cell Directory Service during DCE configuration. If you do not specify this option, during initialization **secd** sleeps for 1 minute if CDS is not available when it tries to export binding information. If the export fails a second time, it sleeps for 2 minutes before it tries again. If it still fails, it sleeps for 4, 8, and 16 minutes between retries. Then, sleep time stays at 16 minutes until the binding export succeeds.

-v[erbose] Runs in verbose mode.

All options start the Security Server on the local node.

Description

The **secd** daemon is the Security Server. It manages all access to the registry database. You must have root privileges to invoke the **secd**.

The Security Server can be replicated, so that several copies of the registry database exist on a network, each managed by a **secd** process. Only one Security Server, the master, can perform database update operations (such as adding an

account). Replicated Security Servers are slave servers that perform only lookup operations (such as validating a login attempt).

A Remote Procedure Call daemon (**rpcd**) must be running on the local node when **secd** is started. Typically, both **rpcd** and **secd** are started at boot time. The **secd** server places itself in the background when it is ready to service requests.

Locksmith Mode

The **secd -locksmith** option starts **secd** in locksmith mode. In locksmith mode, the principal name you specify to **secd** with *pname* becomes the locksmith principal. As the locksmith principal, you can repair malicious or accidental changes that prevent you from logging in with full registry access privileges.

If no account exists for *pname*, **secd** establishes one and prompts you for the account's password. (Use this password when you log in to the account as the locksmith principal.) If an account for *pname* exists, **secd** changes the account and policy information as described in the following tables. These changes ensure that even if account or registry policy was tampered with, you will now be able to log in to the locksmith account.

In locksmith mode, all principals with valid accounts can log in and operate on the registry with normal access checking. The locksmith principal, however, is granted special access to the registry: no access checking is performed for the authenticated locksmith principal. This means that, as the locksmith principal, you can operate on the registry with full access.

Table 4–1. Locksmith Account Changes Made by the Security Server

If the Security Server finds	It changes
Password-Valid flag is set to no	Password-Valid flag to yes
Account Expiration date is set to less than the current time plus one hour	Account Expiration date to the current time plus one hour
Client flag is set to no	Client flag to yes
Account-Valid flag is set to no	Account-Valid flag to yes
Good Since date is set to greater than the current time	Good Since date to the current time
Password Expiration date is set to less than current time plus one hour	Password Expiration date to the current time plus one hour

secd(8sec)

Table 4–2. Registry Policy Changes Made by the Security Server

If the Security Server finds	It changes
Account Lifespan is set to less than the difference between the locksmith account creation date and the current time plus one hour	Account Lifespan to the current time plus one hour minus the locksmith account creation date
Password Expiration date is set to greater than the time the password was last changed but less than the current time plus one hour	Password Expiration date to the current time plus one hour

Use the **-lockpw** option if the locksmith account exists but you do not know its password. This option causes **secd** to prompt for a new locksmith password and replace the existing password with the one entered.

Use the **-remote** option to allow the locksmith principal to log in from a remote machine.

The **secd** program normally runs in the background. When you start **secd** in locksmith mode, it runs in the foreground so that you can answer prompts.

Examples All of the commands shown in the following examples must be run by **root**:

1. Start the master Security Server after you create the master database with **sec_create_db**.

```
$ dcelocal/bin/secd
```

2. Start a slave replica on a node for the first time.

```
$ dcelocal/bin/secd -create
```

3. Restart an existing replica (master or slave).

```
$ dcelocal/bin/secd
```

4. Start the Security Server in locksmith mode and allow the **master_admin** principal to log in on a remote machine.

```
$ dcelocal/bin/secd -locksmith master_admin -remote
```

Chapter 5

Distributed File Service Commands

intro

Purpose 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 referenced in the *OSF DCE Administration Guide*. Refer to that document for more information on 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 on any of the administrative-level (**8dfs**) commands referenced in the files in this list; see the *DCE User's Guide and Reference* for information on any of the user-level (**1dfs**) commands referenced in the files in this list.

Configuration files:

BosConfig(4dfs)

CacheInfo(4dfs)

dfsatab(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)

DFSLog(4dfs)

FileLog(4dfs)

FILog(4dfs)

FMSLog(4dfs)

FtLog(4dfs)

RepLog(4dfs)

SalvageLog(4dfs)

TE(4dfs)

TL(4dfs)

UpLog(4dfs)

BakLog

Purpose 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 domain, 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 append messages to.

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)**, **bakserver(8dfs)**

BosConfig

Purpose 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 (**bosserv** 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

BosConfig(4dfs)

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 also runs independently of any other processes; however, 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 process's binary file is used. For a **cron** process, two command parameters are used: the complete pathname of 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. 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. Process entries must be manipulated with the appropriate **bos** commands. Otherwise, the BOS Server will have incorrect internal information about the processes it is to monitor and control.

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 fs1.abc.com backup cron "dcelocal/bin/fts clonesys -server fs1.abc.com.br -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 **bosserv** process, on **fs1.abc.com** are to be restarted every Sunday morning at 4:00 a.m.

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

Related Information

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

BosLog(4dfs)

BosLog

Purpose 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 (**bosserv** process). The BOS Server runs on every server machine in a domain, 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 append messages to.

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)**, **bosserv(8dfs)**

CacheInfo

Purpose 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 created during DFS client installation (see the *OSF DCE Administration Guide* 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 specifies 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 will not work.

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

- The second field names a local directory to serve as the cache directory. This is the directory in which the Cache Manager stores the **Vn**, **CacheItems**, and **FilesetItems** files that it creates, as well as the **DFSLog** file to which output is written if debugging is enabled with the **cm debug** command. The default entry is *dcelocal/var/adm/dfs/cache*. You can change this to a directory on another partition if more space is available. An entry must appear in this field even if memory caching is used 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 percent of the cache directory's partition for a disk cache and do not use more than 20 to 25 percent of available memory for a memory cache.

CacheInfo(4dfs)

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 percent of the cache directory's partition for a disk cache and do not use more than 20 to 25 percent 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 debug(8dfs)**, **cm getcachesize(8dfs)**, **cm setcachesize(8dfs)**, **dfsd(8dfs)**

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

CacheItems

Purpose 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)**

DFSLog

Purpose Contains debugging output from the `cm debug` command

Description

The **DFSLog** file contains debugging output produced by the `cm debug` command. The amount of information written to the file depends on the value used with the `-level` option of the `cm debug` command. The following values can be used with the `-level` option:

- 1 General debugging information is included
- 2 RPC-related debugging information is included
- 4 Replication-related debugging information is included

The values can be added to produce any combination of the possible information (for example, a value of `7` produces all three types of output).

If the **DFSLog** file does not already exist when the `dfsd` process is started, the process creates it 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. If the **DFSLog** file already exists when the process is started, the process simply truncates (clears) it.

When it executes, the `cm debug` command appends its output to the current contents of the **DFSLog** file. As a result, the file can become very large; it will eventually fill the available disk space on the partition if an extensive trace or multiple traces are run. Use the `df` or `du` command to make sure the partition containing the `dcelocal/var/adm/dfs/cache` directory has enough space to accommodate the **DFSLog** file. Although you cannot remove the file, you can clear its contents to reduce the amount of space it occupies. (If you are using a disk cache, you can also free additional space on the partition by using the `cm setcachesize` command to reduce the amount of space reserved for the cache by the Cache Manager.)

The file is an ASCII file, so it can be viewed with the `more` command (or a similar command appropriate to the local operating system). To read the file, log in as `root` on the machine. Because the file contains debugging information (such as `vnode` operation calls or a record of any RPC errors encountered), interpreting the information written to the file requires an understanding of the DFS source code.

Cautions Never delete the **DFSLog** file; doing so causes the kernel to panic when the **cm debug** command is issued. If the file is accidentally deleted, reboot the machine to restore normal performance.

Related Information

Commands: **cm debug(8dfs)**, **dfsd(8dfs)**

Files: **CacheInfo(4dfs)**

FMSLog

Purpose 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).

The tape size reported in the file should be reduced by 10 to 15 percent before being used in the **TapeConfig** file. The EOF mark size in the file should be increased by 10 to 15 percent 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. In the example, the tape size for the tape drive is 151,224,320 bytes, and the EOF mark size for the drive is 2,375,680 bytes.

```
fms test started
wrote 9230 blocks
Tape capacity is 151224320 bytes
File marks are 2375680 bytes
```

Related Information

Commands: **fms(8dfs)**

Files: **TapeConfig(4dfs)**

FileLog(4dfs)

FileLog

Purpose Contains messages generated by the File Exporter

Description

The **FileLog** file contains execution messages and error messages generated by the File Exporter. The File Exporter is initialized on every File Server machine with the **fxd** command. It runs in the kernel of a File Server machine, managing the delivery and storage of files on the machine, and tracking status information about exported files.

If the **FileLog** file does not already exist when the **fxd** command is issued, the command creates the file in the directory named *dcelocal/var/dfs/adm*. The File Exporter then appends to the file any messages it generates. If the file exists when the **fxd** command is run, the command moves the current version of the file to the **FileLog.old** file in the same directory (overwriting the current **FileLog.old** if it exists) before creating a new version to which the File Exporter can append messages.

The File Exporter can write different types of output to the file, depending on the actions it performs and any problems it encounters. Because it is an ASCII file, the file 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 contents of the log file can help you evaluate problems encountered by the File Exporter.

Related Information

Commands: **fxd(8dfs)**

FilesetItems

Purpose Records location mappings for filesets accessed by the Cache Manager

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 the Cache Manager. The mapping enables the Cache Manager to respond correctly to 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)**

FILog

Purpose Contains messages generated by the Fileset Location Server

Description

The **FILog** 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 domain, providing the interface by which authorized users can modify the Fileset Location Database (FLDB).

If the **FILog** 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 **FILog.old** file in the same directory (overwriting the current **FILog.old** file if it exists) before creating a new version to append messages to.

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 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 domain. 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 append messages to.

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 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 directory named *dcelocal/var/dfs* 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, constantly monitor for the presence of the file.

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 user **anonymous**. This is a serious security risk and should be used only during installation or emergencies (such as server encryption key emergencies).

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 before they execute the command. By default, DFS authorization checking is always enabled on every server machine.

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 to the **bosservice** command used to start the BOS Server). The BOS Server removes the file when a similarly authorized 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)**, **bossver(8dfs)**

RepLog

Purpose 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 domain, 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 append messages to.

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)**

SalvageLog

Purpose Contains output generated by the DFS Salvager

Description

The **SalvageLog** file contains output generated by the DFS Salvager. The Salvager is executed with the **salvage** command. It is used to analyze and repair inconsistencies in file systems on DCE LFS aggregates. The DCE LFS log mechanism can usually return consistency to a file system after a system failure, but some conditions require that the Salvager be run to repair file system inconsistencies.

If the **SalvageLog** file does not already exist when the Salvager initially generates a message, the Salvager creates the file in the directory named *dcelocal/var/dfs/adm*. It then appends any subsequent messages to the file once the file exists.

All output displayed on the screen by the Salvager is also written to the **SalvageLog** file. The Salvager can write different types of output to the file, depending on the actions it performs and the problems it encounters. The amount of output written to the file depends on the options included with the **salvage** command and the extent of the damage to the aggregate being salvaged.

Because it is an ASCII file, the **SalvageLog** file 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 contents of the log file can help in evaluating file system problems encountered by the Salvager.

Related Information

Commands: **salvage(8dfs)**

TE_device_name

Purpose 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 client machine having a tape drive and an associated Tape Coordinator). The **butc** program prompts for new tapes 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 the process is associated with. The file is located in the directory named *dcelocal/var/dfs/backup* on the local disk of the Tape Coordinator machine. The 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 run for a tape drive and Tape Coordinator pair, it automatically creates the error file if the file does not already exist; in this case, the issuer of the **butc** command must have write and execute permissions on the *dcelocal/var/dfs/backup* directory. If the file does exist, the process reinitializes the file (clears its contents), in which case the issuer of the command requires only write permission on the error file. (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 filesets were not added to the Backup Database; messages also indicate why each fileset was not added to the database (in all three cases, dumps having the specified dump IDs already existed).

```
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_device_name

Purpose 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 client machine having a tape drive and an associated Tape Coordinator). The `butc` program prompts for new tapes 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. The output written to the file by the process is 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 run for a tape drive and Tape Coordinator pair, it automatically creates the log file if the file does not already exist; in this case, the issuer of the `butc` command must have write and execute permissions on the `dcelocal/var/dfs/backup` directory. If the file does exist, the process reinitializes the file (clears its contents), in which case the issuer of the command needs only write permission on the log file. (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: port offset 1, debug
level: 0
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 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 — (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 percent 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 percent 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 6. The Backup System can track a maximum of seven tape drives per Tape Coordinator machine; a machine can house one drive, seven drives, or any number of drives in between.

TCIDs can be specified in any order; it is not necessary to assign TCIDs 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 on a machine.

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 the 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 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 append messages to.

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 Contains a chunk of data cached in a disk cache

Description

A Vn 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 a number; each V file has a different number (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 computed as the number of cache blocks divided by 12. You can override the default number of chunks with the **dfsd** command 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 percent 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 using the **-chunksize** option. Specify an integer between 0 (zero) and 20 to be used as an exponent of 2; the default unit of measure is bytes. For example, a value of 16 equals the default chunk size (2^{16} equals 65,536); a value of 10 sets the chunk size to one kilobyte (2^{10} equals 1024). Values less than 10 are not recommended. A value equal to 0 or greater than 20 sets the chunk size to the default, as does a value of 16.

Vn(4dfs)

Note that you cannot override the total size of the cache by setting **-files** or **-chunksize** to a larger size.

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 Contains the administrative list for the Backup Server

Description

The **admin.bak** file is an administrative list of all users and groups who can use the Backup Server to modify the Backup Database. The file typically contains the UUIDs of users and groups only; it is not necessary to add server machines to the **admin.bak** file.

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 machine running the **bakserver** process.

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.

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

Independent versions of the **admin.bak** file should not be maintained on each Backup Database in a domain. Because the Backup Database is a Ubik database, any of the secondary sites may be obligated to assume the role of synchronization site for the Backup Database at any time. A system administrator listed in the **admin.bak** file on the machine housing the former synchronization site may not be

admin.bak(4dfs)

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 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 **bosserv** process, runs on every DFS server machine in a domain. An **admin.bos** file must reside on each machine running the **bosserv** 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 server processes to run in no-authentication mode. Furthermore, 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 radmin** 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 using the **upserver** process running on the System Control machine.

admin.bos(4dfs)

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.

Related Information

Commands: **bos addadmin(8dfs)**, **bos lsadmin(8dfs)**, **bos radmin(8dfs)**, **bossver(8dfs)**

admin.fl

Purpose Contains the administrative list for the Fileset Location Server

Description

The **admin.fl** file is an administrative list of all user and groups who can use the Fileset Location Server (FL Server) to modify the Fileset Location Database (FLDB). The **admin.fl** file typically contains the UUIDs of users and groups only; it is not necessary to add a server machine to the **admin.fl** file.

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 machine running the **flserver** process.

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 rmadm** command. The **bos lsadm** 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.

It is recommended that a single version of the **admin.fl** file be created and maintained on a domain's System Control machine. The **upclient** processes running on the domain's Fileset Database machines can then reference the file using 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 domain. Because the FLDB is a Ubik database, any of the secondary sites may be obligated 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

admin.fl(4dfs)

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 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 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

admin.ft(4dfs)

running on the domain's File Server machines can then reference the file using 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.

Related Information

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

admin.up

Purpose 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 rmdadmin** 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

admin.up(4dfs)

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 radmin(8dfs)**, **upclient(8dfs)**, **upserver(8dfs)**

dfsatab

Purpose Lists DCE LFS aggregates and non-LFS partitions that have been exported

Description

The **dfsatab** file lists information about each DCE LFS aggregate and non-LFS partition that has been exported from the local disk to the DCE namespace. The file is created automatically in the *dcelocal/var/dfs* directory (the same directory **dfstab** is located in) when the **dfsexport** command is first used to export an aggregate or partition to the DCE namespace. The file is then updated whenever the command is used to export a new aggregate or partition.

When the **dfsexport** command is used to export one or more aggregates or partitions, it first checks the **dfstab** file to determine whether the aggregates and partitions to be exported are listed in the file. If they are, it then checks the **dfsatab** file to ensure that they are not listed in that file (which would mean that they have been exported already). If they are listed in the **dfsatab** file, the **dfsexport** command will not export them again.

The **dfsatab** file is an ASCII file that contains a one-line entry for each aggregate or partition exported previously by the **dfsexport** command. Each entry appears on its own line and contains the same information that appears for it in the **dfstab** file. The following information appears for each entry:

Device name The block device name of the exported aggregate or partition.

Aggregate name

The name associated with the aggregate or partition.

File system type

The identifier for the type of file system housing the aggregate or partition.

Aggregate ID

An aggregate ID unique to the **dfstab** file.

Fileset ID

The unique fileset ID associated with the fileset on an exported non-LFS partition. This field does not apply for DCE LFS aggregates.

The **more** command (or the command appropriate to the system being used) can be used to view the file, which requires read permission on the file. The **dfsexport** command can also be issued with no options to list the aggregates and partitions that have entries in the file, which requires being logged into the local machine as

dfsatab(4dfs)

root. You must have write permission on the **dfsatab** file to remove an entry from it; you must have write and execute permissions on the *dcelocal/var/dfs* directory to remove the file entirely.

The **dfsexport** command will not export an aggregate or partition that has an entry in the **dfsatab** file. After rebooting and before initially issuing the **dfsexport** command, remove the entire **dfsatab** file. This is usually done by including the **rm** command (or the command appropriate to the system being used) in the proper initialization file (*/etc/rc* or its equivalent). The **dfsexport** command is typically included there as well.

Examples The **dfsatab** file that follows indicates that one non-LFS partition (*/dev/lv02*) and two DCE LFS aggregates (*/dev/lv03* and */dev/lv04*) were already exported:

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

Related Information

Commands: **dfsexport(8dfs)**

Files: **dfstab(4dfs)**

dfstab

Purpose Specifies DCE LFS aggregates and non-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 being 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.

dfstab(4dfs)**Aggregate ID**

A positive integer different from any other aggregate ID in the **dfstab** file. When an entry is created 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**. When creating an 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. It exports every specified aggregate or partition listed in the file.

It also creates a **dfsatab** file listing the one-line entry from the **dfstab** file for each aggregate or partition that it exports. Because the **dfsatab** file lists those aggregates and partitions already exported, **dfsexport** will not export an aggregate or partition listed in the **dfsatab** file; you must remove an entry from the **dfsatab** file if it is to be exported again.

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)**

Files: **dfsatab(4dfs)**

intro

Purpose Introduction to the DFS commands

Description

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

- | | |
|------------|---|
| bak | Issued by system administrators to operate the DFS Backup System. |
| bos | Issued by users to list system administrator information; they are used by system administrators to use the Basic OverSeer Server (BOS Server). |
| cm | Issued by users to determine machine and cell information; they are used by system administrators to alter and configure the Cache Manager. |
| fts | Issued by users to check quota information; they are used by system administrators to manipulate filesets. |

DFS commands are divided into two groups: user-level commands and administrative commands. User-level commands are designated by the number 1 in the string (**1dfs**) following the command name; for example, **fts lsquota(1dfs)**. Administrative commands are designated by the number 8 in the string (**8dfs**) following the command name; for example, **bos delete(8dfs)**. User-level commands are documented in the *OSF DCE User's Guide and Reference*.

All administrative users can issue user-level commands.

Privileges Required

The majority of administrative commands 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 where the command is to be issued. The exact privilege needed to execute a command is detailed with each command.

DFS Command Types

DFS commands follow these general naming rules. Commands that begin with one of the following:

add or **rm** (add or remove)

Affects lists or groups of DFS objects. For example, **bos addadmin** adds an administrative user to an administrative list.

intro(8dfs)

- cr** or **del** (create or delete)
Affects DFS objects. For example, **fts crserverentry** creates a DFS object, a server entry.
- ls**
Used to display objects and groups of objects.
- set**
Used to assign values to parameters; for example, **fts setrepinfo** assigns replication parameters. Commands beginning with **get** are used to display parameters; for example, **cm getcacheinfo** 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 "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
```

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. For example:

```
$ man bak
```

Reference page for an individual command

To view the reference page for each 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.* For example:

```
$ man bak_dump
```

intro(8dfs)

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**. For example:

```
$ 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: For example:

```
$ bak help dump
```

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 quotes or other delimiters; type all strings in lowercase letters. For example, the following command produces a list of all **bos** commands with the word **create** in their name or description:

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

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 appropriate introductory page for that suite.

bak(8dfs)

bos(8dfs)

cm(8dfs)

fts(8dfs)

bak

Purpose 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** Specifies the DCE pathname of the File Server machine (for example, */.../abc.com/hosts/fs1*) to use with the command.
- tapehost** Specifies the DCE pathname of the client machine (for example, */.../abc.com/hosts/bak1*) where a Tape Coordinator is being added or removed.
- tcid** 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 7. The default for the TCID is 0 (zero); therefore, 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 **intro(8dfs)** reference page.

Description

The **bak** commands are issued to the DFS Backup System by system administrators. The commands copy user and system files to backup tapes and restore information from the tapes. All **bak** commands are restricted to administrative users only.

The Backup System has two main components: a Backup Database, installed on one or more server machines, and Tape Coordinator machines, which can be any client machine. The Backup Database stores two types of records: dump set records, which list the fileset families and tapes in the dump set; and administrative records, which list fileset families and their entries, dump levels, and tape hosts.

A Tape Coordinator machine must be a client machine with a tape drive; it must also run an instance of the **butc**, which is the Backup Tape Coordinator process. A Tape Coordinator 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.

bak(8dfs)

Each TCID is unique to the Backup System with 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.
- Wildcards (UNIX shell metacharacters) do not have to be escaped; in regular command mode, all wildcards must be placed in double quotes or escaped with a **** (backslash).
- 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.
- New connections do not have to be established to the **bakserver** and **flserver** processes each time a command is issued, so execution time is quicker.

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.

The complete syntax for the command is

jobs [-help]

The **-help** option prints 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 `restore.fileset`
- 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 or restore 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, reissue the `bak restore` command.

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 are in the form `fileset_family_name.dump_level`. Because the `bak jobs` command always

bak(8dfs)

assigns the **restore.fileset** identifier to restores, it is impossible to distinguish them by name.

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

Monitoring And Command Connections

When using the Backup System, you must open a separate monitoring connection for each Tape Coordinator process running on the machine. The Tape Coordinator runs in the foreground; any prompts from the Backup System appear in this window.

Aliases

An alias is an alternative 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**

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 pages 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** *command*

Displays a short description of any commands that match the specified string.

Consult the **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 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.

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 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)**

bak adddump

Purpose 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, the elements in the pathname preceding the last one must already exist in the dump hierarchy. However, 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 except a . (period). When including shell metacharacters, escape each metacharacter with a \ (backslash) or “ ” (double quotes).

-expires *date* Defines the expiration date(s) to be associated with each new dump level. Expiration dates can be specified as absolute or relative values.

Absolute expiration dates have the following format

at *mmlddy [hh:mm]*

The word **at** is followed by a date (*month/day/year*) and, optionally, a time (*hours:minutes*). 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 they correspond to (2038-2069) exceed the capacity of the standard UNIX representation of dates (the number of seconds since 12:00 a.m. on January 1 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 [*integery*] [*integerm*] [*integerd*]

The word **in** is followed by a number of years (maximum 9999), months (maximum 12), and days (maximum 31), or a combination of these arguments. At least one of the three 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 extend beyond the year 2038 is truncated to the number of years remaining until 2038.

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

Description

The **bak addump** 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 by 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 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 expiration date, 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 without question. Expiration dates can be either absolute or relative; details are provided earlier in this reference page.

bak adddump(8dfs)

If multiple dump levels are specified with **-level**, a single expiration date can be used for all of the levels or a separate date can be indicated for each level. If more dump levels than expiration dates are specified, the last expiration date provided is applied to all of the extra dump levels.

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 January 1 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** 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*
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*
The File Server machines that house the filesets in the fileset family entry. Legal values are the DCE pathname of a specific machine (for example, */.../abc.com/hosts/fs1*) or the regular wildcard expression (*.**), which matches any machine name. If using the wildcard in noninteractive mode, surround it with double quotes (*“.*”*).
 - aggregate** *name*
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. If using the wildcard in noninteractive mode, surround it with double quotes.
 - fileset** *name*
The fileset to be included in the fileset family entry. Legal values are a specific fileset name, the regular wildcard expression (*.**), or a character string or regular expression that includes the UNIX shell metacharacters described later in this reference page. If using wildcards in noninteractive mode, surround the entire expression with double quotes or escape each wildcard with the ** (backslash).
 - help**
Prints the online help for this command. All other valid options specified with this option are ignored.

bak addftentry(8dfs)**Description**

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

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 machine name or aggregate name found in the Fileset Location Database (FLDB). The **bak** program initiates a search of the entire FLDB to resolve wildcards.

For the **-fileset** option, a wider range of notation is acceptable and can be combined with specific character strings. The characters in regular expressions 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 UNIX shell metacharacters. Note that metacharacters cannot be used for machine or aggregate names.

- * (asterisk) Matches any number of characters (0 or more) and can be combined with any other expression.
- [] (brackets) Around a list of characters match a single instance of any of the characters, but no other characters. For example, **[abc]** matches **a** or **b** or **c** but not **d** or **A** or **ab**. This expression can be combined with the asterisk.
- ^(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**. This expression can be combined with the asterisk.
- ? (question mark) Matches any single character. For example, **?** matches **a** or **A** or **1** or *****.
- \ (backslash) Can precede any of the characters in this list so that they match only their literal value. For example, the expression ***** matches a single asterisk, and the expression **** matches a single backslash.

In the following example, the combination of letters and a wildcard 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 expressions that include wildcards

bak addftentry(8dfs)

in double quotes or to escape the wildcards with the \ (backslash); for example, “**user.*.bak**” or **user.*.bak** are equivalent to the previous example. Otherwise, the command shell attempts to resolve the wildcards rather than passing them to the **bak** command interpreter for resolution. This results in either the failure of the command or the creation of incorrect fileset entries.

Possible values for the arguments of the **bak addftentry** command follow. The fileset family entries are listed here from most to least inclusive. 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 specified 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 specified 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 specified 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 **-servers**, 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 specified 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 specified with **-server**, the aggregate name with **-aggregate**, and the fileset name with **-fileset**

bak addftentry(8dfs)

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

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

Options **-family** *fileset_family_name*

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 except a period. (When a dump set is transferred to tape, the fileset family name and the dump level name are joined by a period to form the name of the dump set.)

Any UNIX shell metacharacters 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, metacharacters 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 periods in the fileset family name. The names of tapes containing dumps of this fileset family consist of the fileset family name and the dump level name joined by a period.

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.

bak addftfamily(8dfs)

Privilege Required

You 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 Adds a Tape Coordinator entry to the Backup Database

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

Options **-tapehost** *machine*

Names the client machine for which the Tape Coordinator is to be added. Specify the DCE pathname of the machine (for example, */.../abc.com/hosts/bak1*).

-tcid *tc_number*

The Tape Coordinator ID (TCID) to be assigned to the Tape Coordinator. Legal values are integers from 0 to 7. A value must match the TCID assigned to the Tape Coordinator in the *dcelocal/var/dfs/backup/TapeConfig* file on the **-tapehost** machine. 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 on a Tape Coordinator machine 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 where the Tape Coordinator is assigned (specified by **-tapehost**).
- The Tape Coordinator's TCID (specified by **-tcid**).
- The UUID of the Tape Coordinator (generated automatically when the command is issued). The UUID is used by **bak** commands to identify the Tape Coordinator that is to perform an operation.

bak addhost(8dfs)

Repeat the command once for each Tape Coordinator on a Tape Coordinator machine. The Backup Database allows a maximum of eight Tape Coordinators on the same Tape Coordinator machine.

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 (-**tapehost**). 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

You 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 Shows each help entry containing a specified string

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

Options **-topic *string*** The keyword string to search for. If it is more than a single word, surround it 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.

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.

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

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(1dfs)**

bak deletedump

Purpose 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 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(8dfs)

bak dump

- Purpose** 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*
The fileset family (already defined in the Backup Database using **bak addftfamily** and **bak addftentry**) to be dumped.
 - level** *dump_level*
The dump level (already defined in the Backup Database using **bak adddump**) to be used in dumping the fileset family. Provide a full pathname for the dump level. 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*
The Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive containing the tape. If omitted, it defaults to 0 (zero).
 - noaction** Prints out 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 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**.

A **full dump** records the structure of all the 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 the directories in each fileset in the fileset family, but it only includes data for those files in the fileset family that changed since the fileset family was dumped at the **-level** parent dump level; such files have modification time stamps later than the date and time when

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 when 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 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 omits the fileset from the dump rather than stopping the dump entirely. If the Tape Coordinator performing the dump is initialized at debug level 1, a report on the failure to include the fileset appears in the 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 name on the label is *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**) and *index* distinguishes this tape from others that contain this same dump. If a single tape contains the entire dump, 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*.

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 is not used for the dump; if it is smaller, the dump fails, but only after it fills the tape and determines that the tape is too small for the drive.

bak dump(8dfs)

In addition, the Backup system checks the expiration date on the tape before it writes data to it. If the date is 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 is expired or if no expiration date is associated with the tape, the system overwrites the contents of the tape without question.

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 print out 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.

Privilege Required

You 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 **bakdump** is issued in interactive mode.

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

```
Starting dump of fileset family 'fileset family' (dump set
'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 a TCID of 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.** fileset family is dumped at the **/full** dump level:

```
$ bak dump sys.osf1_pmax full -n
Starting dump of fileset family 'sys.osf1_pmax' (dump set '/full')
Total number of filesets: 24
Would have dumped the following filesets:
osf1_pmax 124857238
osf1_pmax.bin 124857241
osf1_pmax.etc 124857246
.
.
```

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 restoredisk(8dfs)**, **bak restoreft(8dfs)**, **bak rmdump(8dfs)**, **bak rmftfamily(8dfs)**

bak dumpinfo

Purpose 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 **-ndumps** 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 backups. If a number is specified with **-ndumps**, information about that number of dumps is displayed (information about the most recent **-ndumps** is displayed); if a specific dump ID number is specified with **-id**, information about that dump is displayed; if both options are omitted, information about the last 10 dump operations 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, specify both the **-id** option and the **-verbose** option.

Output The following information is displayed for each dump listed:

- The dump ID number of the dump
- The date the dump was created
- The parent dump level of the dump

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

Privilege Required

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

Related Information

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

bak ftinfo(8dfs)

bak ftinfo

Purpose Queries the Backup Database about a fileset

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 where it resides. If the dump was made of the backup version, as is usual, then **-fileset** must include the **.backup** extension.

Privilege Required

You 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:

Dump ID The dump set's ID number. This is a unique identifier that the Backup System uses internally; it is not related to the dump ID assigned to dump and restore operations when they are started. 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.

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.

- parent ID** 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).
- creation date** The date and time at which the Backup System started executing the dump operation that created this 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 tape containing the dump set.

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

```
$ bak ftinfo user.smith.backup
```

```
Dump ID  lvl  parent ID  creation date  clone date    tape name
654946323 0          0 09/29/91 5:36  9/28/90 4:31 users.week.1
654960415 1 654946323 09/30/91 5:11  9/30/90 4:16 users.monday.1
654972910 1 654946323 10/01/91 5:07 10/01/90 4:01 users.tuesday.1
```

Related Information

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

bak help

Purpose 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. Use 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.

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(8dfs)

bak labeltape

Purpose 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 that goes 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 that actually goes on the tape; *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 the kilobyte. 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 uses this size by default if the **-size** option is omitted.

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.

It is possible to use this command to overwrite an existing label. This is useful if the backup data on a tape is no longer needed, but an unexpired expiration date in the tape's label prevents the tape from being used for dumps.

Privilege Required

No privileges are necessary to label a blank tape. You must be listed in the **admin.bak** files on all Backup Database machines to label a tape that contains information.

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(8dfs)

bak lsdumps

Purpose Displays the dump hierarchy from the Backup Database

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.

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 position at the left margin indicates they are full dump levels).

```
$ bak lsdumps
/month
  /week1
    /tuesday
      /thursday
  /week2
    /tuesday
      /thursday
/monthday
  /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 of 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 **/month/week1** (it contains 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.

bak lsdumps(8dfs)

Related Information

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

bak lsftfamilies

Purpose 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.

Output The output includes a separate entry for each fileset family. The entry defines the fileset family entries that make up the fileset family, assigning each fileset family entry an index number. The issuer of the **bak rmftentry** command uses the index number 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(8dfs)

bak lshosts

Purpose 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 the command is issued for the machine to be displayed with this command.

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.

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

- The name of the machine for 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 UUID of the Tape Coordinator. The UUID is automatically generated when the Tape Coordinator is added. It is used by **bak** commands to identify the Tape Coordinator that is to perform an operation.
- The TCID of the Tape Coordinator. Valid TCIDs for the Tape Coordinators on a machine are integers from 0 to 7.

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

```
$ bak lshosts
```

```
Tape hosts:
```

```
Host /.../abc.com/hosts/bak1, e6b78d0c-8fe9-11ca-acb3-02608c2ef6aa, TCID 0
```

```
Host /.../abc.com/hosts/bak1, fa8c8f9c-8fe9-11ca-8dba-02608c2ef6aa, TCID 1
```

```
Host /.../abc.com/hosts/bak2, af1921fe-8fe9-11ca-823e-02608c2ef6aa, TCID 1
```

```
Host /.../abc.com/hosts/bak3, f197c02a-8fe9-11ca-bfdf-02608c2ef6aa, TCID 0
```

```
Host /.../abc.com/hosts/bak3, f3e7f570-8fe9-11ca-9d0b-02608c2ef6aa, TCID 1
```

```
Host /.../abc.com/hosts/bak3, f09a788e-8fe9-11ca-98c3-02608c2ef6aa, TCID 3
```

Related Information

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

bak readlabel(8dfs)

bak readlabel

Purpose 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 read from the label of the tape in the indicated tape drive. These values are placed on the tape either with **bak dump** or **bak labeltape**.

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 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 no suffix follows the size, it is a number of kilobytes.

If a tape is labeled as empty, the output reads `NULL` size. If a tape is 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 TCID 6.

```
$ bak read 6
Tape read was labelled: sys.Monthly.3 size: 2097152
```

The following command shows that the unlabeled tape in the drive with TCID 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(8dfs)

bak restoredb

Purpose 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 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 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. 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 where it is located.

After the database is removed, restart all **bakserver** processes on the machines where they were running. Use **bak addhost** to add tape hosts for the Tape Coordinator(s) from which you will restore the Backup Database. Then use **bak restoredb** 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 installed with the **bak addhost** command).

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

Privilege Required

You 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** Restores the entire contents of an aggregate
- 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 server name as a DCE pathname (for example, */.../abc.com.hosts/fs1*).
 - aggregate name**
Specifies the device name or aggregate name of the aggregate on the indicated **-server** 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 the issuer is placing the necessary tapes. If omitted, it defaults to 0 (zero).
 - newserver machine**
Names the File Server machine to which to restore the data. This is necessary only if it is different from **-server**. Specify the server name as a DCE pathname (for example, */.../abc.com/hosts/fs2*).
 - newaggregate name**
Specifies the device name or aggregate name of the aggregate on **-newserver** 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. This is necessary only if the aggregate is different from **-aggregate**.
 - noaction** Specifies that the program print out the list of tapes necessary to perform the indicated restore but not actually perform the restore.
 - 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 by **-server** and **-aggregate** 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.

This command is useful if a disk or machine crash destroys the data on an entire aggregate.

The issuer can restore the aggregate contents to alternate sites. To restore the contents to

- A different aggregate on the same File Server machine, specify the aggregate with **-newaggregate**
- An aggregate of the same name on a different File Server machine, specify the File Server machine with **-newserver**
- A completely different site, specify both the File Server machine with **-newserver** and the aggregate with **-newaggregate**

The **-noaction** option instructs the program to produce a list of the tapes the Backup System would need to perform the indicated restore without actually performing the restore.

To restore a defined group of filesets, see the **bak restoreft** command.

Privilege Required

You 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 to which aggregates are to be restored.

Output

If the issuer does not include the **-noaction** option, this command returns the unique dump ID number associated with this 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 this command is issued in interactive mode.

bak restoredisk(8dfs)

If the issuer includes 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 `/dev/lv01` aggregate of `fs5` to that aggregate. The issuer places the tapes in the drive with TCID 3.

```
$ bak restoredi ../abc.com/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 `/dev/lv02` aggregate of `fs1` to a new site, `/dev/lv01` on `fs3`. The issuer places the tapes in the drive with TCID 0 (zero).

```
$ bak restoredi ../abc.com/fs1 /dev/lv02 -news ../abc.com/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)**

Files: **dfstab(4dfs)**

bak restoreft

Purpose 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 on which to restore the fileset. Specify the server name as a DCE pathname (for example, */.../abc.com/hosts/fs1*). If the fileset currently exists at a site other than the one specified with this option and the **-aggregate** option, the restored fileset is stored on the specified machine and the current fileset is removed.

-aggregate name

Specifies the device name or aggregate name of the aggregate to which to restore the 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, the restored fileset is stored on the specified machine and the current fileset is removed.

-fileset name Names each fileset to be restored. Specify the names as they are recorded in the Backup Database. Provide the name of the read-write version of each fileset, even if (because of its fileset entry definition in the fileset family) the backup version of a fileset was actually dumped.

-extension name_extension

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

bak restoreft(8dfs)

-date *date* Specifies the date prior to which a dump must be done to be included in the restore. The **-date** option indicates a date-specific restore; only dump sets dated before **-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/ddlyy 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 time. For example, **1/23/90** or **11/22/91**.

mm/ddlyy 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 time. The time must be in 24-hour format (for example, **20:30** is 8:30 p.m.).

-tcid *tc_number* Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator for the tape drive where the issuer is placing the necessary tapes.

-noaction Specifies that the program produce the list of tapes necessary to perform the indicated restore but not actually perform the restore.

-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 the indicated filesets 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 there are incremental dumps, the issuer is prompted to insert the necessary tapes into the tape drive.

The issuer 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 up to the final one done, it includes only the full dump and any incremental dumps done before the indicated date.

In general, the issuer needs to provide the name of the fileset's read/write version, even if, as recommended, the definition in the fileset family causes its backup version to be dumped. Even if the backup version was dumped, the Backup Database records the fileset under its read/write name; the issuer of this command needs to identify the fileset in that way.

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 overwrite a fileset's current contents with data restored from tape, omit the **-extension** option. The results are as follows:

- If **-server** and **-aggregate** specify the fileset's current site, the restored data overwrites the current contents. There is no change in the Fileset Location Database (FLDB) entry for the fileset.
- If **-server** and **-aggregate** indicate a new site, the restored data is stored in a new fileset at that site, and the current fileset is deleted from the system. The FLDB records the change in 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 **-server** and **-aggregate** then contains the restored data. It has the same name as the current fileset, with the addition of the distinguishing extension. The Fileset Server assigns the new fileset a fileset ID number automatically; a new FLDB entry records all the appropriate information about the new fileset.

It is also possible to restore a fileset that no longer has an existing counterpart in the file system. A new fileset at the site specified with **-server** and **-aggregate** contains the restored data.

It is possible to have the program produce a list of the tapes that the Backup System would need to perform the indicated restore without having the system actually perform the restore. To do so, include the **-noaction** option.

Privilege Required

You must be listed in the **admin.bak** files on all Backup Database machines. You 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 to which filesets are to be restored.

Cautions Overwriting an existing fileset destroys any file 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 **-extension** to restore data to a new fileset.

Output If you do not include the **-noaction** option, this command returns the unique dump ID number associated with this restore operation. The dump ID is displayed in the

bak restoreft(8dfs)

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 this 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 this restore operation. No dump ID number is reported because none is assigned.

Examples The following command restores the fileset **user.pat** to **/dev/lv01** on **../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)**

Files: **dfstab(4dfs)**

bak rmdump

Purpose 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.
- `-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 **-level** 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.

Privilege Required

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

Examples The following command deletes the dump level called **Week3** from the dump hierarchy:

```
$ bak rmd Week3
```

Related Information

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

bak rmftentry(8dfs)

bak rmftentry

Purpose 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

You 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 previously 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 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 Removes a Tape Coordinator entry from the Backup Database

Synopsis **bak rmhost -tapehost** *machine* [**-tcid** *tc_number*] [**-help**]

Options **-tapehost** *machine*

Names the client machine from which the Tape Coordinator is to be removed. Specify the DCE pathname of the machine (for example, */.../abc.com/hosts/bak1*).

-tcid *tc_number*

Specifies the Tape Coordinator ID (TCID) of the Tape Coordinator to be removed. Legal values are integers from 0 to 7. A value must match the TCID assigned to the Tape Coordinator in the *dcelocal/var/dfs/backup/TapeConfig* file. 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 (**-tapehost**). Remove the entry for the Tape Coordinator from the **TapeConfig** file.

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

You 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 TCID 1 on **bak2** from the Backup Database:

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

Related Information

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

Files: **TapeConfig(4dfs)**

bak savedb(8dfs)

bak savedb

Purpose 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 will 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 **Ubik_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

You 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 TCID 3:

```
$ bak save 3
```

Related Information

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

bak scantape

Purpose 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

bak scantape(8dfs)

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

If the **-dbadd** option is omitted, no privileges are required. If the **-dbadd** option is specified, you must be listed in the **admin.bak** files on all Backup Database machines.

Cautions Using the **-dbadd** option on this command creates 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*.

creationTime

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

expirationDate

When the tape expires. A value of **Wed Jan 1 00:00:00 1970** (the UNIX time equivalent of 0 (zero)) indicates that no expiration date is defined. (The time displayed in this case is different in different time zones; it is adjusted to reflect the difference between the time zone where data was dumped to the tape and Greenwich mean time.)

cell The cell the dump set was created in.

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.

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 of the fileset.

dumpSetName
The dump set the fileset belongs to.

dump ID The dump ID number of the dump set named by **dumpSetName**. This is a unique identifier used internally by the Backup System.

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).

parent ID 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. 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

bak scantape(8dfs)

Examples The following command shows the output for the first two filesets on a tape:

```
$ bak scantape
Tape label
-----
name = guests.monthly.1
creationTime = Fri Nov 22 05:59:31 1990
expirationDate = Wed Dec 31 19:00:00 1969
cell = abc.com
size = 20103324 Kbytes
dump path = /monthly
-- End of tape label --
-- fileset --
fileset name: user.guest10.backup
fileset ID 0,,112262
dumpSetName: guests.monthly
dump ID 169580184
level 0
parent ID 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 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.

`-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 January 1, 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 following format:

in [*integery*] [*integerm*] [*integerd*]

The word **in** is followed by a number of years (maximum 9999), months (maximum 12), and days (maximum 31), 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 extend beyond the year 2038 is truncated to the number of years remaining until 2038.

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

bak setexp(8dfs)**Description**

The **bak setexp** command sets the expiration date on each dump level specified with **-level**. The 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 the tape contents 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 expiration date, 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 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

You 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 December 31, 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/week -exp 7d
```

Related Information

Command: **bak adddump(8dfs)**, **bak dump(8dfs)**, **bak labeltape(8dfs)**

bak status

Purpose 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.

Output

If the indicated Tape Coordinator is not currently performing an operation, the output reports `Tape coordinator is idle`. Otherwise, it reports

- 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** or **bak restoredisk** 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.

bak status(8dfs)

- 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 **restore.fileset** if the operation is a restore; **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 TCID 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.

Related Information

Commands: **bak dump(8dfs)**, **bak labeltape(8dfs)**, **bak restoredb(8dfs)**, **bak restoredisk(8dfs)**, **bak restoreft(8dfs)**, **bak savedb(8dfs)**, **bak scantape(8dfs)**, **bak jobs**, **bak kill** (see the **bak.8dfs** reference page for information about the **bak jobs** and **bak kill** commands)

bak verifydb(8dfs)

bak verifydb

Purpose 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).

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

bak verifydb(8dfs)

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)**

bakserver(8dfs)

bakserver

Purpose Initializes the Backup Server

Synopsis **bakserver** [-**adminlist** *filename*] [-**bakgroup** *server_group*] [-**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*.

-**bakgroup** *server_group*

Names a different CDS server group to be contacted by the Backup Server process. Do not use this option; it is intended for debugging purposes only.

-**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** reference 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 the Backup Database (or a copy of it). It is usually started and controlled by the BOS Server; if it is not, execute the **bakserver** as a background process. The binary file for the **bakserver** process resides in *dcelocal/bin/bakserver*.

The first time it is initialized, the **bakserver** creates the Backup Database in *dcelocal/var/dfs/backup*; all database files have a root name of **bkdb**. The **bakserver** also creates the *dcelocal/var/dfs/admin.bak* administrative list file if the file does not already exist. The principals and groups in the **admin.bak** administrative list are authorized to issue commands that affect the Backup Database (examining the database, dumping the database to tape, and so on).

Because the Backup Database is a replicated database, the **admin.bak** administrative list must contain the same principals and groups for all **bakserver** processes in an administrative domain.

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.

In addition, when the **bakserver** 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 respond to.

When it is started, the **bakserver** 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** 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

You must be logged in as **root** on the local machine.

Output If problems are encountered during initialization, the **bakserver** prints error messages to the standard error output. The **bakserver** keeps an event log file in *dcelocal/var/dfs/adm/BakLog*.

Related Information

Files: **admin.bak(4dfs)**, **BakLog(4dfs)**

bos

Purpose 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** Names the machine running the BOS Server that is to execute the command. Always specify the name of the machine as a DCE pathname (for example, */.../abc.com/hosts/fs1*).
- noauth** Directs the **bos** program to use the unprivileged identity **anonymous** as the identity of the issuer of the command. If authorization checking has been disabled (with the **bos setauth** command), the identity **anonymous** has the necessary privileges to perform any operation. If you use this option, do not use the **-localauth** option.
- localauth** Directs the **bos** program to use the identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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. For complete details about receiving help, see the **intro(8dfs)** reference page.

Description

The **bos** commands are used by system administrators to contact the Basic OverSeer Server (BOS Server). The BOS Server runs on every File Server machine and monitors the other 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 information and lists of administrative users.

The BosConfig File

The information in the *dcelocal/var/dfs/BosConfig* file on the local disk of each File 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.

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 radmin** commands. Never edit the **admin.bos** file directly; always use the appropriate **bos** commands.

The Key File

A **/krb5/v5srvtab** key file is stored on the local disk of each File Server machine. A key 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 key file.

The keys in a key 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 key file. Never edit a key 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 pages 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 using the command in DFS.*

bos(8dfs)

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

\$ **bos help** *command*
 Displays the syntax for a single command.

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

Consult the **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 key 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(1dfs)**, **bos create(8dfs)**, **bos delete(8dfs)**, **bos exec(8dfs)**, **bos gckey(8dfs)**, **bos genkey(8dfs)**, **bos getdates(8dfs)**, **bos getlog(8dfs)**, **bos getrestart(8dfs)**, **bos help(1dfs)**, **bos install(8dfs)**, **bos lsadmin(8dfs)**, **bos lscell(1dfs)**, **bos lskeys(8dfs)**, **bos prune(8dfs)**, **bos restart(8dfs)**, **bos radmin(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)**

Books: User-level commands, designated by **(1dfs)**, are documented in the DCE DFS portion of the *OSF DCE User's Guide and Reference*.

bos addadmin

Purpose 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 containing the administrative list to which principals and groups are to be added. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.
- adminlist** *filename*
Names the administrative list to which principals and groups 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 to add to the file. User and server names are specified with the same option because they both have principal names.
- group** *name* Specifies the name of each group to add to the file.
- createlist** Specifies that the **-adminlist** file is to be created if it is not found. This option has no effect if the specified file already exists. Because the **admin.bos** list must already exist to issue this command, specifying **admin.bos** as the value of **-adminlist** has no effect.
- noauth** Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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.

bos addadmin(8dfs)**Description**

The **bos addadmin** command adds the specified users, groups, and servers to the administrative list specified by **-adminlist** on the server machine indicated by **-server**. 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. Users added to the administrative list 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

You 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 Converts a string into a server encryption key and adds it to a key file

Synopsis **bos addkey -server** *machine* **-kvno** *version_number* **-password** *string*
[**-principal** *name*] [**-localonly**] [{**-noauth** | **-localauth**}] [**-help**]

Options

- server** *machine*
Names the server machine whose key file is to have a new key added to it. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.
- kvno** *version_number*
Defines the key version number of the new key. The version number must be a decimal integer in the range 0 (zero) to 255, and it must be unique for the principal specified by **-principal** in the key file on the machine specified by **-server**.
- 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 machine. It can include any characters; it can also include spaces if the entire string is enclosed in 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 key file on the machine indicated by **-server**, but that the Registry Database is not to be updated. The default is both to add the key to the local key file and to update the Registry Database.
- noauth**
Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS

bos addkey(8dfs)

Server is run. The issuer must be logged into the 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 specified by **-principal** in the **/krb5/v5srvtab** key file on the machine specified by **-server**. The key is derived from **-password** and is given the version number specified by **-kvo**. The server encryption key associated with the specified principal is automatically updated in the Registry Database unless **-localonly** is specified, in which case the Registry Database is not updated. The key file must already exist before this command can be used to add a key to it. (Key files are created using the **rgy_edit** command.)

The **bos genkey** command is a more secure way of adding a key to a key file because it generates a random key. It also always updates the Registry Database.

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.

Examples The following command adds a new server encryption key with key version number **14** to the key 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 key file.

```
$ bos addk ../abc.com/hosts/fs1 14 “fourteenth new key” -localonly
```

Related Information

Commands: **bos gkeys(8dfs)**, **bos genkey(8dfs)**, **bos lskeys(8dfs)**, **bos rmkey(8dfs)**

bos create

Purpose Creates a new process in the BosConfig file and start 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. Specify the server name as a DCE pathname.

-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

repsrver For the Replication Server process

bakserver For the Backup Server process

-type *process_type*

Specifies the process type. Legal values are **simple** and **cron**. Specify **simple** for independent processes and **cron** for processes that are to run only at specified times.

-cmd 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**

bos create(8dfs)

processes take options, in which case the entire argument must be surrounded by 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.

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 1:00 to 12:00 with am or pm, for example, **"2:30 pm"**. As shown in the example, enclose the entire entry in double quotes if it contains spaces.

To execute the command only once, specify **now** instead of a day or a day and time, or issue the command directly (using **bos exec**); the process entry remains in the **BosConfig** file. To place the process entry in the **BosConfig** file without ever executing it, specify **never** instead of a day or a day and time.

- noauth** Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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**:

```
$ 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(8dfs)

bos delete

Purpose 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. Specify the server name as a DCE pathname.

-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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 exec(8dfs)

bos exec

Purpose Executes a command on a server machine

Synopsis **bos exec -server** *machine* **-cmd** *command* [{**-noauth** | **-localauth**}] [**-help**]

Options **-server** *machine*

Names the server machine on which to execute the command. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.

-cmd *command*

Specifies the complete pathname of the command to be executed. Surround this option with quotes if the command contains spaces.

-noauth

Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 exec** command executes the command specified by **-cmd** on the server machine specified by **-server**. The command executes with **root** permission, so it should be used very carefully. Its main purpose is to allow rebooting of server machines remotely.

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples The following command reboots **fs2** by executing the proper command (**/etc/reboot** in the example). Before rebooting, the issuer entered the **bos shutdown** command to shutdown all DFS server processes.

```
$ bos exec /.../abc.com/hosts/fs2 /etc/reboot
```

Related Information

Commands: **bos shutdown(8dfs)**

bos gckey(8dfs)

bos gckey

Purpose Removes obsolete server encryption keys from a key file

Synopsis **bos gckey -server** *machine* [-**principal** *name*] [{-**noauth** | -**localauth**}] [-**help**]

Options -**server** *machine*

Names the server machine whose key file is to have obsolete keys removed from it. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.

-**principal** *name*

Provides the principal name for which obsolete keys are to be removed from the key file. The default is the DFS principal name of the machine specified by -**server**.

-**noauth**

Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 gckey** command removes obsolete server encryption keys from the **/krb5/v5srvtab** key file on the server machine specified by -**server**. Only obsolete keys associated with the principal name specified by -**principal** are removed from the key file.

Keys are removed based on age and lack of use. The removal process, referred to as "garbage collection", affects only the key 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**.

Examples The following command removes obsolete keys associated with the principal **hosts/fs1/dfs-server** from the key 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 gckey /.../abc.com/hosts/fs3 hosts/fs1/dfs-server
```

Related Information

Commands: **bos addkey(8dfs)**, **bos genkey(8dfs)**, **bos lskeys(8dfs)**, **bos rmkey(8dfs)**

bos genkey

Purpose Generates a random key and adds it to a key file

Synopsis **bos genkey -server** *machine* **-kvno** *version_number* [**-principal** *name*]
[**-noauth** | **-localauth**] [**-help**]

Options **-server** *machine*

Names the server machine whose key file is to have a new key added to it. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.

-kvno *version_number*

Defines the key version number of the new key. The version number must be a decimal integer in the range 0 (zero) to 255, and it must be unique for the principal specified by **-principal** in the key file on the machine specified by **-server**.

-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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 specified by **-principal** in the **/krb5/v5srvtab** key file on the machine specified by **-server**. A random key is generated and is given the version

number specified by **-kvno**. The server encryption key associated with the specified principal is automatically updated in the Registry Database. The key file must already exist before this command can be used to add a key to it. (Key files are created with the **rgy_edit** command.)

The **bos addkey** command can also be used to add a key to a key 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.

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.

Examples The following command generates a new server encryption key with key version number **14** and adds it to the key 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)**

bos getdates(8dfs)

bos getdates

Purpose Lists timestamps 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 where the binary files to be checked reside. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.

-file *binary_file*

Names the current version of each binary file whose timestamps are to be displayed. The timestamps on the three versions (current, **.BAK**, and **.OLD**) of each file are displayed. Only the last element of the pathname is necessary. All specified files must reside in the same directory.

-dir *alternate_dest*

Specifies the complete pathname of the directory the specified files reside in. This is necessary only if the files reside in a directory other than *dcelocal/bin* (the directory the DFS binary files reside in).

-noauth

Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 timestamps for the current version, **.BAK** version (if any), and **.OLD** version (if any) of each binary file whose current version is specified with the **-file** option. The **-dir** option can be used to specify the name of the directory where the files reside if it is different from *dcelocal/bin*. Use the **-server** option to specify the name of the server machine where the files reside.

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 the **.BAK** version (if it exists, the **.OLD** version replaces the **.BAK** version). Use the **bos prune** command to remove old versions of **.BAK** and **.OLD** files.

Output For each argument of **-file**, the output reports the timestamp on the current, **.BAK**, and **.OLD** versions. The output reports any versions that do not exist.

Examples The following command displays the timestamps 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

Commands: **bos install(8dfs)**, **bos prune(8dfs)**, **bos uninstall(8dfs)**

bos getlog(8dfs)

bos getlog

Purpose 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. Specify the server name as a DCE pathname.

`-file log_file` Names the log file to display. If a simple filename is provided, with no slashes, it is assumed to reside in *dcelocal /var/dfs/adm*; the standard choices from that directory are **BakLog**, **BosLog**, **FILog**, **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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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` from 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
FILog	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.

Privilege Required

No privilege is required if the filename specified by **-file** does not contain a / (slash). If the name contains a / (slash), you 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)**, **FILog(4dfs)**, **FtLog(4dfs)**, **RepLog(4dfs)**, **UpLog(4dfs)**

bos getrestart(8dfs)

bos getrestart

Purpose 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 server name as a DCE pathname.

-noauth Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged in to the 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 **-server**:

- The time each week when the BOS Server process automatically restarts itself and all processes that have the status flag of **Run** in the **BosConfig** file
- The time each day when the BOS Server automatically restarts any process executed from a file in **dcelocal/bin** whose binary file has a time stamp later than the last restart time for the process

Output The output consists of the following two lines:

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

Possible values for *time* include

never Indicates that the BOS Server never performs that type of restart

now Indicates that the BOS Server performs that type of restart each time it restarts and reads the **BosConfig** file, but not between those times

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.abc.com restarts at sun 4:00 am
```

```
Server fs2.abc.com restarts for new binaries at 2:15 am
```

Related Information

Commands: **bos setrestart(8dfs)**

Files: **BosConfig(4dfs)**

bos install(8dfs)

bos install

- Purpose** Installs a new version of a binary file
- Synopsis** **bos install -server** *machine* **-file** *binary_file...* [**-dir** *alternate_dest*]
 [{**-noauth** | **-localauth**}] [**-help**]
- Options**
- server** *machine*
Names the server machine where the new binary files are to be installed. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.
 - file** *binary_file*
Specifies the complete pathname of each binary file on the source (local) machine to be installed on the machine specified by the **-server** option. The pathname can specify a directory other than *dcelocal/bin*, but the last element (the filename) must match the file in *dcelocal/bin* that the specified file is replacing; for example, **fileserver** for the Fileset Location Server. See the description of the **bos create** command for a list of the standard binary filenames.
 - dir** *alternate_dest*
Specifies the complete pathname of the directory on the machine specified by the **-server** option where the binary files are to be installed. All files specified by the **-file** option are installed in this directory. If this option is omitted, the files are installed in the *dcelocal/bin* directory.
 - noauth** Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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. This machine should be the Binary Distribution machine for its CPU/operating system type. If it is not, the new binary files will be overwritten the next time the **upclient** process on the machine copies new versions of the binary files using the **upserver** process on the Binary Distribution machine. Unless the **-dir** option names a different directory, the files are installed in the *dcelocal/bin* directory.

By default, it takes the Update Server five minutes to propagate newly installed binary files to the other machines of the same CPU/OS type as **-server**. (Note that the Update Server, not the BOS Server, handles this propagation.)

The command automatically preserves the current version of the binary file by adding a **.BAK** extension to its name. If there is a current **.BAK** version at least seven days old, it replaces the current **.OLD** version. If there is a current **.BAK** version less than seven 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.

To make the machine specified by **-server** start using the new binary files immediately, issue the **bos restart** command. Otherwise, the new 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 if* the affected processes are already under the control of the BOS Server.)

Use the **bos uninstall** command to replace the current version of a binary file with the **.BAK** version of the file (if it exists, the **.OLD** version replaces the **.BAK** version). Use the **bos getdates** command to check the time stamps on binary files. Use the **bos prune** command to remove old **.BAK** and **.OLD** files from the *dcelocal/bin* directory (the command can also be used to remove core files from the *dcelocal/var/dfs/adm* directory).

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)**

Files: **BosConfig(4dfs)**

bos lsadmin(8dfs)

bos lsadmin

- Purpose** 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 contains the administrative list whose principals and groups are to be displayed. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname (for example, */../abc.com/hosts/fs1*).
 - 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 users, groups, and servers found in the administrative list specified by **-adminlist** on the server machine specified by **-server**. 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.

Output The output lists the name of each user, group, and server entitled to issue administrative commands for the server process associated with the administrative list. The names of users and servers are displayed as principal names, which is how they are added and removed from administrative lists.

Related Information

Commands: **bos addadmin(8dfs)**, **bos rmdiradmin(8dfs)**

Files: **admin.bak(4dfs)**, **admin.bos(4dfs)**, **admin.fl(4dfs)**, **admin.ft(4dfs)**, **admin.up(4dfs)**

bos lskeys

- Purpose** Displays server encryption key information from a key file
- Synopsis** **bos lskeys -server** *machine* [**-principal** *name*] [{**-noauth** | **-localauth**}] [**-help**]
- Options**
- server** *machine*
Names the server machine whose key file is to have keys listed. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.
 - 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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** key file on the server machine specified by **-server**. It displays information for keys associated with the principal name specified by **-principal**; the DFS principal name of the server machine specified with **-server** is used by default.

Authorization checking must be disabled on the machine specified by **-server** to display the string of octal numbers that compose the key (use the **bos setauth** command to disable authorization checking). This 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 keys are being examined (both turning off authorization checking and displaying keys on a screen are serious security risks).

If 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

You must be listed in the **admin.bos** file on the machine specified by **-server**. If authorization checking is enabled, checksums are displayed instead of the actual keys. No privilege is required to see the actual keys, but authorization checking must be disabled first with **bos setauth**, which requires being listed in the **admin.bos** file on the machine.

Output

The **bos lskeys** command displays one line for each server encryption key associated with **-principal** in the key file on the machine specified by **-server**. Each key is identified by its key version number. If authorization checking is enabled on the machine, a checksum is displayed with each version number; if authorization checking is disabled, the octal numbers that comprise 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.

Examples

The following command shows the checksums for the keys associated with the principal name of **fs3.abc.com** in the key file on that machine. The checksums appear instead of the actual keys because 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.
```

bos lskeys(8dfs)

The following command lists the keys associated with **fs3** after authorization checking is disabled with **bos setauth**:

```
$ 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)**

bos prune

- Purpose** 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. Specify the server name as a DCE pathname.
- 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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

bos prune(8dfs)

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 getdates** command to determine the time stamps on binary files. Use the **bos uninstall** command to replace the current version of a binary file with its **.BAK** version (if it exists, the **.OLD** version replaces the **.BAK** version).

Privilege Required

You 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 Restarts processes on a server machine

Synopsis **bos restart -server** *machine* [{-bosserver | -process *server_process...*}]
[{-noauth | -localauth}] [-help]

Options **-server** *machine*

Names the server machine on which to restart the indicated processes. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.

-bosserver *server_process*

Indicates that the current BOS Server instance is to stop running. A new BOS Server instance immediately starts and restarts all processes with the status flag **Run** in the **BosConfig** file. Provide this option or provide the **-process** option.

-process *server_process*

Specifies each process to be stopped; if a process has the status flag **Run** in the **BosConfig** file, it is then restarted. 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). Provide this option or provide the **-bosserver** option.

If this option and the **-bosserver** option are both omitted, all processes running on the server machine except the BOS Server are stopped; those with the status flag **Run** in the **BosConfig** file are restarted. If this option is omitted and the **-bosserver** option is specified, all processes running on the server machine, including the BOS Server, are stopped; a new BOS Server instance starts immediately, and it then restarts all processes with the status flag **Run** in the **BosConfig** file.

-noauth Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS

bos restart(8dfs)

Server is run. The issuer must be logged into the 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 and immediately restart the specified processes. Processes are specified with the following options:

- The **-bosservice** option indicates that all processes on **-server**, including the BOS Server, are to stop. A new BOS Server starts immediately and then starts a new instance of any process marked with the status flag **Run** in the **BosConfig** file.
- The **-process** argument causes all specified processes on **-server** to stop. It then restarts any specified process marked with the status flag **Run** in the **BosConfig** file. Do not include **bosservice** in the list of processes; use the **-bosservice** option instead.
- The absence of both **-process** and **-bosservice** indicates that the BOS Server is to stop all processes, except itself, running on the indicated machine. It 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 in the **BosConfig** file for a process.

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples The following command stops all processes running on **../abc.com/hosts/fs3**, including the BOS Server. A new BOS Server instance starts immediately and restarts all processes marked 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 currently running. The BOS Server then restarts all processes marked 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 radmin

- Purpose** Removes a user, group, or server from an administrative list
- Synopsis** **bos radmin -server** *machine* **-adminlist** *filename* [**-principal** *name...*] [**-group** *name...*] [**-removelist**] [**{-noauth |-localauth}**] [**-help**]
- Options**
- server** *machine*
Names the server machine containing the administrative list whose principals or groups are to be removed. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.
 - adminlist** *filename*
Names the administrative list from which principals or groups 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 to be removed from the administrative list. User and server names are specified with the same option because they both have principal names.
 - group** *name* Specifies the name of each group to be removed from the administrative list.
 - removelist** Specifies that the **-adminlist** file is to be deleted if it is empty after the specified users and groups are removed. This option has no effect if the specified file is not empty after the specified users and groups are removed.
 - noauth** Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 rmadm** command removes the specified users, groups, and servers from the administrative list specified by **-adminlist** on the server machine specified by **-server**. 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. Users removed from the administrative list 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

You 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 rmadm -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 Removes server encryption keys from a key file

Synopsis **bos rmkey -server** *machine* **-kvno** *version_number...* [**-principal** *name*]
[**(-noauth | -localauth)**] [**-help**]

Options **-server** *machine*

Names the server machine whose key file is to have keys removed from it. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.

-kvno *version_number*

Specifies the key version number of each key to be removed from the key file. Each key having a specified version number and associated with the specified **-principal** is removed. Each version number must be a decimal integer in the range 0 (zero) to 255.

-principal *name*

Provides the principal name associated with the keys being removed from the key file. The default is the DFS principal name of the machine specified by **-server**.

-noauth

Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 each server encryption key specified by **-kvno** and associated with the principal specified by **-principal** from the **/krb5/v5srvtab** key file on the server machine specified by **-server**. It has no effect on the Registry

Database. Any client machines using a key when it is removed from a key file can no longer access data on the server machine from which the key is removed.

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**.

Examples The following command removes two keys from the key 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 gckey(8dfs)**, **bos genkey(8dfs)**, **bos lskeys(8dfs)**

bos setauth(8dfs)

bos setauth

Purpose 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***
Specifies the server machine where the status of DFS authorization checking is to change. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.
- authchecking**
Determines whether or not server processes on the machine check for authorization. A value of **on** enables DFS authorization checking; a value of **off** disables it.
- noauth**
Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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. The state of DFS authorization checking on a server machine determines whether or not all DFS server processes running on the machine check that the issuer of a command is correctly authorized to execute the command (is included in the necessary administrative lists). If DFS authorization checking is disabled on a server machine, the DFS server processes on that machine perform any action for anyone, even the unprivileged user

anonymous, which is a serious security risk. DFS authorization checking on a machine should always be enabled except during installation or emergencies.

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, constantly monitor for the presence of this file; they do not check for authorization when it is present. To indicate that DFS authorization checking is enabled (the normal state), 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 **bossver** 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 machine. It is not necessary to restart currently running 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 **-noauth** option available with many **bos** and **fts** commands is used when authentication information is unavailable or unnecessary. Use the **-noauth** option when DFS authorization checking is disabled.

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server** to modify the state of DFS authorization checking on that machine in any way.

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: **bossver(8dfs)**

Files: **NoAuth(4dfs)**

bos setrestart(8dfs)

bos setrestart

Purpose 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. Specify the server name as a DCE pathname.

-general *time* Sets the time when the BOS Server restarts itself then restarts all server processes 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 or 1:00 through 12:00 with **am** or **pm** — for example, “**12:30 pm**”. As shown in the example, enclose the entry in quotes if it contains a space.

Also, the issuer can use either of two additional definitions instead of a day and time:

never Indicates that the BOS Server never performs the indicated type of restart

now Indicates that the BOS Server performs the indicated restart immediately and not again

If this option is never used to set the general restart time, the default general restart time is Sunday at 4:00 a.m.

-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 is once per day, so it is standard to specify only a time of day. Use the conventions described for times under **-general** to express the time of day, and enclose it in double quotes if it contains a space. The remarks under **-general** 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 **-server** is to perform one of two types of restarts, recording 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 **-bosservice** 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 is 5:00 a.m. each day.

You must issue the command twice to change both times.

bos setrestart(8dfs)

Privilege Required

You must be listed in the **admin.bos** file on the machine specified by **-server**.

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 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 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** 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. Specify the server name as a DCE pathname.
 - 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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

bos shutdown(8dfs)

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

You 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 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 where each process is to be started. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.
- 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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.

bos start(8dfs)

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

You 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 **bos restart** 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 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. Specify the server name as a DCE pathname.
- 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged in as **root** to the machine 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 startup** 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

bos startup(8dfs)

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 **bos restart** after this command to start a new instance.

Privilege Required

You 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 (**repsrvr** 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 repsrvr
```

Related Information

Commands: **bos create(8dfs)**, **bos restart(8dfs)**, **bos start(8dfs)**, **bos status(8dfs)**,

File: **BosConfig(4dfs)**

bos status

Purpose Displays the status 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 server name as a DCE pathname.

-**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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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.

bos status(8dfs)

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. Refer to the *OSF DCE Administration Guide* for instructions on using this command to determine the role of a server machine.

Output

The output 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

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** 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 reporting when the command is next scheduled to execute.

The following additional information is displayed 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 shutdown(8dfs)**, **bos start(8dfs)**, **bos startup(8dfs)**, **bos stop(8dfs)**

Files: **BosConfig(4dfs)**

Books: *OSF DCE Administration Guide*

bos stop(8dfs)

bos stop

Purpose 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. Specify the server name as a DCE pathname.
- 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 to remain absent 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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

You 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(8dfs)

bos uninstall

Purpose Uses the former version of a binary file

Synopsis **bos uninstall -server** *machine* **-file** *binary_file...* [**-dir** *alternate_dest*]
[**{-noauth | -localauth}**] [**-help**]

Options **-server** *machine*

Names the server machine where the former (**.BAK**) versions of binary files are to be used. The BOS Server on this machine executes the command. Specify the server name as a DCE pathname.

-file *binary_file*

Names each binary file to be replaced with its **.BAK** version. Only the last element of the pathname is necessary. All specified files must reside in the same directory.

-dir *alternate_dest*

Specifies the complete pathname of the directory in which the specified files reside. This is necessary only if the files reside in a directory other than *dcelocal/bin* (the directory in which the **bosserv** resides).

-noauth

Directs **bos** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer; this identity is not necessarily that of the machine on which the BOS Server is run. The issuer must be logged into the 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 **.BAK** version. If it exists, the current **.OLD** version becomes the **.BAK**

version. Use the **-server** option to specify the name of the server machine where the files reside. This should be the Binary Distribution machine for its CPU/operating system type. If it is not, the binary files will be overwritten the next time the **upclient** process on the machine copies new versions of the binary from the **upserver** process on the Binary Distribution machine. Unless the **-dir** option names a different directory, the files are removed from the *dcelocal/bin* directory.

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 if* the affected processes are 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 getdates** command to check the time stamps on binary files. Use the **bos prune** command to remove old **.BAK** and **.OLD** files from the *dcelocal/bin* directory (the command can also be used to remove core files from the *dcelocal/var/dfs/adm* directory).

Privilege Required

You 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)**

Files: **BosConfig(4dfs)**

bossserver(8dfs)

bossserver

Purpose Initializes the Basic OverSeer (BOS) Server process

Synopsis **bossserver** [-adminlist *filename*] [-noauth] [-help]

Options **-adminlist** *filename*

Specifies the file that contains principals and groups authorized to execute **bossserver** RPCs (usually using **bos** commands). If this option is omitted, the **bossserver** obtains the list of authorized users from the default administrative list file, *dcelocal/var/dfs/admin.bos*.

-noauth Invokes the **bossserver** with DFS authorization checking turned off. In this mode, the **bossserver** process does not check to see whether the issuer is authorized to enter commands.

This option is intended for use when the BOS Server is initially installed on a server machine. Because it starts the **bossserver** with DFS authorization checking turned off, it allows you to add members to the **admin.bos** administrative list and to add a key to the key file on the server machine.

Use this mode sparingly, as it presents a security risk. Using this option also forces all other 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 **bossserver** 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 **bossserver** process, monitors each server process that has a process entry in the **BosConfig** file. The **bossserver** must run on all DFS server

machines. It is usually executed as a background process added to the proper system startup file (*/etc/rc* or its equivalent). The binary file for the **bossserver** process resides in *dcelocal/bin/bossserver*.

The first time the **bossserver** is initialized, it creates several directories, such as the *dcelocal/var/dfs/adm* directory (and any nonexistent directories along this path), sets the owner to **root**, and sets the mode bits. The **bossserver** 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.

When it is started, the **bossserver** also 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 **bossserver** 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 **bossserver** processes in the administrative domain.

When initially installing the BOS Server on a server machine, use the **-noauth** option to initialize the **bossserver** process with DFS authorization checking turned off. Then add members to the **admin.bos** list and add a key to the key file on the server machine. When these steps are complete, use the **bos setauth** command to turn DFS authorization checking on; because running with DFS authorization checking disabled is a serious security risk, enable DFS authorization checking as soon as the previous steps are complete.

Privilege Required

You must be logged in as **root** on the local machine.

Output If problems are encountered during initialization, the **bossserver** prints error messages to the standard error output. The **bossserver** keeps an event log in *dcelocal/var/dfs/adm/BosLog*.

Related Information

Commands: **bos setauth(8dfs)**

Files: **admin.bos(4dfs)**, **BosConfig(4dfs)**, **BosLog(4dfs)**

butc(8dfs)

butc

Purpose Initializes a Tape Coordinator process

Synopsis **butc** [-**tcid** *tc_number*] [-**debuglevel** *trace_level*] [-**bakgroup** *server_group*]
[-**cell** *cellname*] [-**help**]

Options -**tcid** *tc_number*

Specifies the Tape Coordinator ID (TCID) associated with the Tape Coordinator. Use this number to indicate which Tape Coordinator is to execute a command.

Legal values are the integers 0 to 7. 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:

0 (zero) The Tape Coordinator only prompts the issuer to place new tapes in the drive; the process does not report on its activities. This is the default value.

1 The Tape Coordinator reports on its activities as it restores filesets, in addition to prompting for new tapes as necessary.

-**bakgroup** *server_group*

Names a different CDS server group to be contacted by the **butc** process. Do not use this option; it is intended for debugging purposes only.

-**cell** *cellname* Specifies the cell where 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 initializes a Tape Coordinator on a Tape Coordinator machine (a client 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. The binary file for the program 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), the Fileset Location Database (by way of the Fileset Location Server), 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 on the local disk of the Tape Coordinator machine: a tape log file and a tape error file. The program writes information on its execution of operations to a log file named *dcelocal/var/dfs/backup/TL_device_name*, where *device_name* is the device name of the tape drive with which the process is associated. The level of detail to which each operation is described depends on the operation.

The program writes information on any problems it encounters during the execution of operations to an error file named *dcelocal/var/dfs/backup/TE_device_name*, where *device_name* is again the device name of the tape drive with which the process is associated. For example, if you use the **bak dump** command to backup 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 lists the names of only those filesets that were omitted from the dump.

butc(8dfs)

No maintenance is required for the log and error files associated with any tape drive; the files are automatically created or reinitialized each time the **butc** process for a tape drive and Tape Coordinator pair is started. However, the **TL_device_name** and **TE_device_name** 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

Each time it is started, the **butc** process creates the log and error files if they do not already exist, in which case the issuer of the **butc** command must have **write** and **execute** permissions on the *dcelocal/var/dfs/backup* directory. If the two files already exist, the issuer need only the **write** permissions on the files.

Related Information

Files: **TapeConfig(4dfs)**, **TE(4dfs)**, **TL(4dfs)**

cm

Purpose 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 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 **intro(8dfs)** reference page.

Description

The **cm** commands are issued by administrative users to update cached information on local workstations. Specific commands can also be used to set cache parameters and to debug cache manager performance.

All user-level commands are documented in the DCE DFS portion of the *OSF DCE User's Guide and Reference*.

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 pages 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(8dfs)

\$ **cm help** *whereis*

Displays the syntax for a single command.

\$ **cm apropos -topic** *command*

Displays a short description of any commands that match the specified string.

Consult the **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(1dfs)**, **cm checkfilesets(8dfs)**, **cm debug(8dfs)**, **cm flush(8dfs)**, **cm flushfileset(8dfs)**, **cm getcachesize(8dfs)**, **cm getdevok(8dfs)**, **cm getsetuid(8dfs)**, **cm help(1dfs)**, **cm lscellinfo(8dfs)**, **cm lsstores(8dfs)**, **cm resetstores(8dfs)**, **cm setcachesize(8dfs)**, **cm setdevok(8dfs)**, **cm setsetuid(8dfs)**, **cm statsservers(1dfs)**, **cm sysname(8dfs)**, **cm whereis(1dfs)**

Books: User-level commands, designated by **(1dfs)**, are documented in the *OSF DCE User's Guide and Reference*.

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.

Related Information

Commands: **cm flush(8dfs)**, **cm flushfileset(8dfs)**

cm debug(8dfs)

cm debug

Purpose Enables, disables, or reports the status of the Cache Manager debugging trace

Synopsis **cm debug -debug {on | off | report} [-level *number*] [-help]**

Options

-debug	Controls the production of debugging information. Legal values are
on	Directs debugging information into the DFSLog file
off	Stops the recording of information in the DFSLog file
report	Reports whether the debugging trace is available and, if it is, whether it is currently enabled (on) or disabled (off)

-level *number* Determines the type of information the Cache Manager produces about its activities. Legal values are

1	Provides general debugging information
2	Provides RPC-related debugging information
4	Provides replication-related debugging information

The possible values can be added and their sum specified to provide any combination of available debugging information. For example, a value of 7 produces all three types of debugging information. If no level is specified, a value of 0 is used; a value of 0 causes the Cache Manager to produce no output. The debugging level can be changed while debugging is currently enabled.

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

Description

The **cm debug** command enables or disables the Cache Manager debugging trace. If the command is executed with **-debug** specified as **on**, debugging information is directed into the **DFSLog** file, which by default resides in *dcelocal/var/adm/dfs/cache* (the cache directory). However, the location of the cache directory and thus the **DFSLog** file is overridden if a different directory is

named in the second field of the *dcelocal/etc/CacheInfo* file or if the **-cachedir** option is specified with the **dfsd** command.

If the command is executed with **-debug** specified as **off**, debugging information being written to the **DFSLog** file is stopped. When executed with **-debug** set to **report**, the command reports the current status of the trace (**on** or **off**) without changing the setting. The debugging level can be changed while debugging is enabled.

Before using this command, issue the **du** or **df** command to make sure the partition housing the cache directory has enough space to accommodate the **DFSLog** file generated by the trace. This file records trace information; it can become very large and eventually fills the partition if a sizable trace or multiple traces are run. Although you cannot remove the file, you can clear its contents to reduce the amount of disk space it occupies. (If you are using a disk cache, you can free additional space by using the **cm setcachesize** command to reduce the amount of space reserved for the cache by the Cache Manager.)

To read the **DFSLog** file, log in as **root** on the local machine and use the **more** command (or a similar command appropriate to your system). Because the file contains debugging information (for example, **vnode** operation calls or a record of any RPC errors encountered), interpreting the trace requires an understanding of the DFS source code.

Note that the debugging facility may not be available with your system. If it is not available, no output is written to the **DFSLog** file. Use the **report** argument with this command to determine whether the debugging facility is available.

Cautions Make sure you disable the trace when you are done. Otherwise, the **DFSLog** file eventually fills the partition. To avoid this situation, enable the trace, execute a single command, and disable the trace.

Also, never delete the **DFSLog** file. Doing so causes the kernel to panic when the **cm debug** command is issued. If the file is accidentally deleted, reboot the machine to restore normal performance.

Output When **report** is specified as a value with **-debug**, the output displays the following message if the debugging trace is not available:

The debugging facility is not available.

If the debugging trace is available, the command reports whether the trace is on or off. All other debugging information is written to the **DFSLog** file.

cm debug(8dfs)

Examples The following command enables both the general and the replication-related traces:

```
$ cm debug on 5
```

The following command disables the current trace:

```
$ cm debug off
```

The following commands perform a general trace of only the **cm getcachesize** command:

```
$ cm debug on 1; cm getcachesize; cm debug off
```

Related Information

Commands: **dfsd(8dfs)**

Files: **CacheInfo(4dfs)**, **DFSLog(4dfs)**

cm flush

Purpose Forces the Cache Manager to discard data cached from a file or directory

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.

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(8dfs)

cm flushfileset

Purpose Force the Cache Manager to discard data cached from filesets containing 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.

Examples The following command flushes data cached from the fileset containing the current working directory and the directory **reports**, which are both at the same level in the file tree:

```
$ cm flushf ../reports
```

Related Information

Commands: **cm checkfilesets(8dfs)**, **cm flush(8dfs)**

cm getcachesize

Purpose 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 a disk cache and on machines using a memory cache.

This information comes from the kernel of the workstation where the command is issued. On machines using a disk cache, the current cache size may disagree with the default setting specified in the **CacheInfo** file if someone sets the cache size with the **cm setcachesize** command. On machines using either a disk or a memory cache, the size may also disagree with the default setting if it is changed with the **dfsd** process.

Output The 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 in use on the machine (**disk** or **memory**).

cm getcachesize(8dfs)

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)**

cm getdevok

Purpose Shows whether device files 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 you want device file status information for by specifying the name of a file or directory on the fileset with the **-path** option. This information comes from the kernel of the workstation where 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.

Output This 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
Device files from the fileset are honored.

device files not allowed
Device files from the fileset are not honored.

cm: the fileset on which '*pathname*' resides does not exist
The specified *pathname* is invalid.

cm getdevok(8dfs)

Examples The following command indicates that device files from the fileset containing 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 getsetuid

- Purpose** Shows the status of **setuid** programs on 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 on the fileset with the **-path** option. This information comes from the kernel of the workstation where the command is issued.

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.

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:

no setuid allowed

The **setuid** and **setgid** programs from the fileset are disabled.

setuid allowed

The **setuid** and **setgid** programs from the fileset are enabled.

cm: the fileset on which '*pathname*' resides does not exist

The specified *pathname* is invalid.

cm getsetuid(8dfs)

Examples The following command indicates that **setuid** and **setgid** programs from the fileset containing 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 lscellinfo

Purpose 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 where the command is issued.

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**).

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(8dfs)

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 where the command is issued.

When the Cache Manager cannot write data back to a File Server machine (for example, because the File Server machine is down, because disk quota on the fileset containing the data is exceeded, or because the aggregate housing the fileset is full), the Cache Manager returns a failure code to the application program using the data. The Cache Manager then attempts to store the data for three minutes.

The Cache Manager also displays a message on the screen to notify the user as soon as it cannot write data back to a File Server machine. The user then has three minutes to alleviate the problem that prevents the data from being stored (for example, to remove data from the fileset if insufficient quota is preventing unstored data from being written to the fileset). If the problem is alleviated during this period, the Cache Manager stores the data; if the problem still exists after the Cache Manager has continued its attempts for the full three minutes, the Cache Manager discards the unstored data. Other data in the cache is not affected.

To prevent the Cache Manager from retrying the write operation for the full three minutes and to discard the data the Cache Manager has failed to store, issue the **cm resetstores** command.

Related Information

Commands: **cm resetstores(8dfs)**

cm resetstores

Purpose Cancels attempts by the Cache Manager to write data back to a File Server and discard the unstored data

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 write data back to a File Server machine. All data the Cache Manager cannot store is discarded (there is no way to selectively abort retry attempts for individual files or filesets).

When the Cache Manager cannot write data back to a File Server machine (for example, because the File Server machine is down, because disk quota on the fileset housing the data is exceeded, or because the aggregate where the fileset resides is full), the Cache Manager returns a failure code to the application program using the data. The Cache Manager then attempts to store the data for three minutes.

The Cache Manager also displays a message on the screen to notify the user as soon as it cannot write data back to a File Server machine. The user then has three minutes to alleviate the problem that prevents the data from being stored (for example, to remove data from the fileset if insufficient quota is preventing unstored data from being written to the fileset). If the problem is alleviated during this period, the Cache Manager stores the data; if the problem still exists after the Cache Manager has continued its attempts for the full three minutes, the Cache Manager discards the unstored data. Other data in the cache is not affected.

To list the fileset ID numbers of filesets that contain data the Cache Manager cannot write to a File Server machine, issue the **cm lsstores** command.

Related Information

Commands: **cm lsstores(8dfs)**

cm setcachesize

Purpose 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 **-reset**.
- reset** Returns the cache size to the value set when the machine was last booted. Use this option or use **-size**.
- 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 use this command on a machine using a memory cache.*

Do not set the cache size to exceed 90% of the actual disk space available for the cache directory. The cache implementation itself requires a small amount of room on the partition.

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.

To return the cache size to the value set when the machine was last booted, use the **-reset** option instead of the **-size** option. This is the amount specified in the **CacheInfo** file unless the **-blocks** option is used with **dfsd** to override the **CacheInfo** value.

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

You 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)**

cm setdevok

Purpose Specifies whether device files on 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 on 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 startup 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.

The **cm getdevok** command displays whether the Cache Manager honors device files from indicated filesets.

Privilege Required

You must be logged in as **root** on the local machine.

Examples The following command causes device files that reside on the fileset containing the directory `../../abc.com/fs/usr/jlw` to be honored:

```
$ cm setdevok ../../abc.com/fs/usr/jlw
```

Related Information

Commands: **cm getdevok(8dfs)**

cm setsetuid

Purpose Enables or disables setuid programs on 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 on 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 startup 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 to execute with **setuid** permission.

A **setuid** program is indicated when a mode bit associated with a file is set. 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 **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.

The **cm setsetuid** 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

You 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 sysname(8dfs)

cm sysname

Purpose Reports or set 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 unpredictable.)

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 'osf1_pmax'.
```

The following commands set the system type on a machine running AIX 3.1 and use it in a symbolic link from the local machine to the binary file for the **emacs** program stored in the DFS filesystem:

```
# cm sys -new aix31_rs  
# ln -s ../../abc.com/fs/@sys/usr/bin/emacs /usr/bin/emacs  
# cd /usr/bin  
# ls -l emacs  
emacs --> ../../abc.com/fs/aix31_rs/usr/bin/emacs
```

dfsbind

Purpose Resolves CDS pathnames and obtain and maintain user authentication information

Synopsis **dfsbind** [-debug] [-help]

Options **-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 must be run in the background.

-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** and **bos apropos** reference pages for examples of using these commands.

Description

The **dfsbind** command starts the **dfsbind** process on a client machine or a File Server machine. The **dfsbind** process performs two services on a client machine: it resolves CDS pathnames that it receives from the Cache Manager, and it obtains user authentication information for the kernel RPC Runtime. It performs one service on a File Server machine: it maintains user authentication information required by the machine. The binary file for the **dfsbind** process resides in *dcelocal/bin/dfsbind*.

When a user on a client machine requests data, the Cache Manager sends the pathname for the data to **dfsbind**. If the pathname corresponds to a location in a DFS cell, the **dfsbind** process returns a set of Fileset Location Server (FL Server) bindings to the Cache Manager. Each binding exists in the form of an RPC binding handle (machine address, protocol, and endpoint or port on a server machine).

The binding handle identifies the FL Server that stores the location of the data. It also identifies where the FL Server is running and how to communicate with it. The Cache Manager uses the binding to contact the FL Server on the appropriate server machine. The FL Server tells the Cache Manager which File Server machine houses the fileset containing the data requested by the user.

The **dfsbind** process converts bindings to socket addresses before returning them to the Cache Manager. The Cache Manager uses the converted bindings and the

UUID of the FL Server from the appropriate cell to establish contact with that server for further name resolution.

The process also returns one of the following error codes to indicate the result of the name resolution operation:

0 The cell was found and the bindings exist.

EISDIR The cell was found but no bindings exist.

ENOENT The cell was not found.

ETIMEDOUT

Unexpected errors occurred.

If the error code is **0**, the process sends the binding socket address and the UUID of the FL Server in a packet to the Cache Manager. If any other error code is generated, the process returns the error code in the packet to the Cache Manager.

The second service provided for a client machine by the **dfsbind** process is to communicate with the Security Server (**secd**) to obtain authentication information about users of the client machine. The Cache Manager communicates with the kernel RPC Runtime when it needs to create RPC bindings for a File Server machine on behalf of a user. The kernel RPC Runtime then communicates with the **dfsbind** process 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 into an RPC binding packet that can be sent to the File Server machine.

On a File Server machine, the **dfsbind** process maintains user authentication information required by the File Server machine. The File Server machine uses this information to ensure that only authenticated users can access data from the machine. A machine that runs the Fileset Server (**ftserver** process), the File Exporter (which is initialized by the **fxd** process), and the **dfsbind** process is considered a DFS File Server machine.

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 must be run in the background.

The **dfsbind** process must be run on all DFS client and File Server machines. It is usually added to the proper startup file (**/etc/rc** or its equivalent) rather than

dfsbind(8dfs)

entered at the command shell prompt. On a client machine, the **dfsbind** process must be run before the **dfsd** process in a startup file. On a File Server machine, the **dfsbind** process must be run before the **fxd** process in a startup file.

Privilege Required

You must be logged in as **root** on the local machine.

Examples The following line, entered in the appropriate initialization file (*/etc/rc* or its equivalent) on a client machine or a 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 machine or a File Server machine.

```
dfsbind
```

Related Information

Commands: **dfsd(8dfs)**, **fxd(8dfs)**

dfsd

Purpose Initializes the Cache Manager and start 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*] [-**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 measure for block size is always kilobytes.

A disk cache should not exceed 90 percent of the disk space available on the cache partition; a memory cache should not exceed 20 to 25 percent 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.

For a memory cache, do not combine this option with the **-dcache** option.

-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 12.

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 percent 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).

dfsd(8dfs)**-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. Generally used for testing purposes only.

-cachedir *cache_directory*

Names the local disk directory to be used as the cache for disk caching. (It is also the directory the **DFSLog** file is placed in). 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*.

-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*

The name of the cell containing the root fileset. Use this option only with a DCE diskless machine.

-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 a time standard.

-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 daemons.

Increase the number of background daemons if the machine serves more than five users.

-tokenprocs *number_of_token_daemons*

Specifies the number of dedicated background daemons that service incoming token revocation RPC requests. The default is two daemons.

Increase the number of token daemons if users on this machine interact with many File Server machines from multiple cells.

-iopprocs *number_of_I/O_background_daemons*

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 daemons.

Increase the number of I/O daemons if many users use the machine.

-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 with disks).

-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 0 and 20 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 equal to 0 (zero) or

dfsd(8dfs)

greater than 20 sets the chunk size to the appropriate default; values less than 10 (1 kilobyte) are not recommended. 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.

- 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 piped 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 piped 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** and **bos apropos** reference pages for examples using these commands.

Description

The **dfsd** process initializes the Cache Manager on a client machine according to the information specified with the options described previously. More specifically, it

- Transfers information about cell membership and the names and network addresses of Fileset Location Database (FLDB) machines from the CDS namespace 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 **-memcache** is provided.
- 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 **cm setcachesize** 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.)

- 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 chunks in the 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.
- Specifies the directory on the machine's local disk where DFS is mounted. The first field in the **CacheInfo** file specifies the default directory.
- Specifies the read/write fileset that corresponds to the root of the DFS file tree for a diskless machine. For a diskless machine, use the **-rootfileset** option to specify the fileset that serves as the top-level directory. This overrides the default specified in the first field of the **CacheInfo** file.
- Specifies the name of the cell containing the root fileset. For a DCE diskless machine, use the **-rootcell** option to specify the name of the cell that contains the root fileset.
- Selects a random server machine in the local cell as the source of the correct time if **-settime** is provided.
- Creates the **DFSLog** file in the cache directory. Debugging information generated with the **cm debug** command is appended to this file.

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.

dfsd(8dfs)

- One maintenance daemon, which performs routine periodic maintenance tasks such as
 - 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 that of the chosen server machine
- 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. No more than six background daemons are ever necessary.
- One or more token daemons, which handle token revocation RPC requests from the File Exporters on File Server machines. The daemons also respond to periodic probes from the File Exporters to the client machine verifying that the client machine is still active. 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.
- 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 many users are using the machine and the machine begins to experience performance problems.

The default number of daemons is ten (one maintenance daemon, two background daemons, two token daemons, and five I/O daemons). The issuer can alter only the number of background daemons, token daemons, and I/O daemons; **dfsd** always initializes just one maintenance daemon.

The **dfsd** process must be run on all DFS client machines. It is usually added to the proper startup 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 startup file. The binary file for the **dfsd** process resides in *dcelocal/bin/dfsd*.

Privilege Required

You must be logged in as **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 debug(8dfs)**, **cm getcachesize(8dfs)**, **cm setcachesize(8dfs)**, **dfsbind(8dfs)**

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

dfsexport

Purpose Exports DCE LFS aggregates or non-LFS partitions to the DCE namespace

Synopsis **dfsexport** [{**-all** | **-aggregate** *name*}] [**-type** *name*] [**-detach**] [**-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**.

-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**. Use this option or use **-all**.

-type *name*

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 that 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.

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).

Use this option only when no users are accessing data on the aggregates or partitions to be detached or when a serious emergency requires its use.

- 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 of 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. Issue this command with no options to list the aggregates and partitions already exported. 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 **-all** and **-aggregate** options are mutually exclusive.

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 supplied, the **-all** option must also be specified, because the **-type** option is ignored otherwise.

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.

When the command exports an aggregate or partition, it copies the entry that appears for that aggregate or partition from the **dfstab** file to the *dcelocal/var/dfs/dfsatab* file that it creates on the local disk of the machine. Thus, the **dfsatab** file maintains a list of the aggregates or partitions the command has already exported.

The command will not export an aggregate or partition for which an entry already exists in the **dfsatab** file. If the aggregate or partition must be exported a subsequent time, or if this command is included in the machine's initialization file

dfsexport(8dfs)

and the machine needs to be rebooted, entries for all devices to be exported must be deleted from the **dfsatab** file or the **dfsatab** file must be removed entirely. Issuing the **dfsexport** command with no options lists the aggregates and partitions that have entries in the **dfsatab** file.

This 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 listed in the **dfstab** file whenever the machine is rebooted. Typically, a command to remove the **dfsatab** file is included in the initialization file, after which the **dfsexport** 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.

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; it cannot contain data. 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 located, the **fts create** command can be used to create and register filesets on the aggregate, after which the **fts crmount** command must 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 **-all** (and optionally **-type**) or **-aggregate** to specify the devices to be detached, and include the **-detach** option with the command. (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

The issuer must be logged in as **root** on the local machine.

Cautions Before using the **-detach** option with this command, make sure no users are currently accessing data from filesets on the aggregates or partitions to be detached. The command does not verify that a device is not in use before removing it from the namespace. A user who is accessing data housed on an aggregate or partition when it is detached will not be able to save the data back to the device. Any attempt to perform an action that involves a detached aggregate or partition elicits a message reporting that the device is unknown.

Examples The following two command lines are typically added to a machine's initialization file (**/etc/rc** or its equivalent). The first line removes the **dfsatab** file from the machine; the second line exports all of the aggregates and partitions that have entries in the machine's **dfstab** file.

```
rm -f dcelocal/var/dfs/dfsatab
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
```

Related Information

Commands: **fts create(8dfs)**, **fts crfldbentry(8dfs)**, **fts crmount(8dfs)**, **newaggr(8dfs)**

Files: **dfsatab(4dfs)**, **dfstab(4dfs)**

flserver

Purpose Initializes the Fileset Location Server (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, **flserver** uses the default administrative list file, *dcelocal/var/dfs/admin.fl*.

 -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 **flserver** command. See the **bos help** and **bos apropos** reference pages for examples of using these commands.

Description

The Fileset Location Server (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, **flserver** creates the FLDB files in *dcelocal/var/dfs*; all database 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 groups in the **admin.fl** administrative list are authorized to issue commands to create server entries and fileset entries in the FLDB. Because the FLDB is a replicated database, the **admin.fl** administrative list must contain the same principals and groups for all **flserver** processes in an administrative domain.

In addition, when **flserver** 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 respond to.

When it is started, **flserver** creates the *dcelocal/var/dfs/adm/FILog* event log file if the file does not already exist. It then appends messages to the file. If the file exists when **flserver** is started, the process moves it to the **FILog.old** file in the same directory (overwriting the current **FILog.old** file if it exists) before creating a new version to append messages to.

Privilege Required

You must be logged in as **root** on the local machine.

Output If problems are encountered during initialization, the **flserver** prints error messages to the standard error output. The **flserver** process keeps an event log in *dcelocal/var/dfs/adm/FILog*.

Related Information

Files: **admin.fl(4dfs)**, **FILog(4dfs)**

fms(8dfs)

fms

Purpose 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 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 percent before being used in the **TapeConfig** file. It is also recommended that the EOF mark size be increased by 10 to 15 percent 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, you 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 you only need 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: 9320
Finished data capacity test - rewinding
wrote 9230 blocks, 9230 file marks
Finished file mark test
Tape capacity is 151224320 bytes
File marks are 2375680 bytes
```


fms(8dfs)

It writes the following information to the **FMSLog** file:

```
fms test started
wrote 9230 blocks
Tape capacity is 151224320 bytes
File marks are 2375680 bytes
```

The tape drive used in the example uses tapes 151,224,320 bytes in size, and creates end of file marks of size 2,375,680 bytes in size.

Related Information

Files: **FMSLog(4dfs)**, **TapeConfig(4dfs)**

fts

Purpose 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.

- server** Specifies the DCE pathname of the File Server machine (for example, */.../cellname/hosts/hostname*) to use with the command. This option is typically used to provide the name of the File Server machine the fileset or filesets used with the command reside on.
- aggregate** 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.
- fileset** Specifies the fileset to use with the command. You can specify either a fileset name or a fileset ID.
- cell** Specifies that the command is to be run in 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 **anonymous** as the identity of the issuer of the command. If authorization checking has been disabled (with the **bos setauth** command), the identity **anonymous** has the necessary privileges to perform any operation. If you use this option, do not use the **-localauth** option.
- localauth** Directs the **fts** program to use the identity (principal name) of the machine on which the command is issued as the identity of the issuer. The issuer must be logged into the 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 this command. All other valid options specified with this option are ignored. For complete details about receiving help, see the **intro(8dfs)** reference page.

fts(8dfs)**Description**

The user-level commands in the **fts** command suite include the **fts lsquota** command used to list information about filesets. All user-level commands are documented in the *OSF DCE User's Guide and Reference*.

Most **fts** commands are administrative-level commands used by system administrators to contact the Fileset Server and the Fileset Location Server (FL Server). These commands 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 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 they reside on.

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 update(8dfs)

fts zap(8dfs)

FLDB Information

The FLDB is maintained by the FL Server. A master copy of the FLDB is stored on one Fileset Database machine, with copies synchronized on other Fileset Database machines using 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 syncserv** and **fts syncflldb** 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 the *OSF DCE Administration Guide*.

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 the fileset is stored on. The fileset header binds all the files into a logical unit without requiring that they be stored in contiguous memory blocks.

The information in the header of a DCE LFS fileset includes the fileset's name, fileset ID number, type (read/write, read-only, or backup), size, the ID numbers of its parent, clone, and backup versions, its creation date, and the date at which it was last updated.

fts(8dfs)**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 **man** pages for the command suite.

\$ man fts_*command*

Displays the **man** 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 using 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 command*

Displays a short description of any commands that match the specified string.

Consult the **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. Specific privilege information is listed with each command's description.

Related Information

Commands: **fts addsite(8dfs)**, **fts aggrinfo(8dfs)**, **fts apropos(1dfs)**, **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(1dfs)**, **fts lock(8dfs)**, **fts lsaggr(8dfs)**, **fts lsfldb(8dfs)**, **fts lsft(8dfs)**, **fts lsheader(8dfs)**, **fts lsmount(8dfs)**, **fts lsquota(1dfs)**, **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 syncfdb(8dfs), fts syncserv(8dfs), fts unlock(8dfs), fts unlockfdb(8dfs), fts update(8dfs), fts zap(8dfs)

Books: User-level commands, designated by **(1dfs)**, are documented in the *OSF DCE User's Guide and Reference*.

fts addsite

Purpose 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 the replica is to be stored on. Specify the server name as a DCE pathname.

-aggregate *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate the replica is to be stored on. 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 setreplinfo** 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must

be logged into the 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 **-server** 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 **-server**.

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 (an array of pointers to the read/write fileset), 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.

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.

fts addsite(8dfs)

The `DefaultSiteAge` associated with the read/write fileset is used by default if the `-maxsiteage` option is omitted. The option is required 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 agrinfo` 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 a read-only fileset at that site. Use the `fts lsreplicas` command to determine the status of the read-only replica at a site.

Privilege Required

You 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 fileset the replication site is to be added to resides.

Examples The following command defines a read-only site for the `osf1_pmax.bin` fileset. The site is defined as the aggregate whose device name is `/dev/lv01` on the File Server machine named `fs3`.

```
$ fts addsite osf1_pmax.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 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 server name as a DCE pathname.
- 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 about all of the exported aggregates and partitions on the specified machine is provided.
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued on as the identity of the issuer. The issuer must be logged into the 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.

fts aggrinfo(8dfs)**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 the aggregates and partitions reside on. 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 on the specified server is displayed.

Much of the information displayed is specified in the *dcelocal/var/dfs/dfstab* file. The **fts lsaggr** command can also be used to list all of the aggregates and partitions exported from a File Server machine.

For a non-LFS partition, the **fts aggrinfo** command displays the same information as the **df** command available in the UNIX operating system. However, the **df** command cannot be used to display information about DCE LFS aggregates.

Output

The **fts aggrinfo** command displays a separate line for each aggregate or partition. Each line displays the following information:

- The file system type (DCE LFS or non-LFS).
- The aggregate name.
- The device name.
- The number of kilobytes of disk space currently available on the aggregate or partition.
- The total number of kilobytes on the aggregate or partition.
- The number of kilobytes reserved for overdraw by some non-LFS implementations. (Overdraw is disk space reserved in case the allocated quota of the partition is exceeded.) DCE LFS aggregates do not reserve overdraw disk space.

The **fts aggrinfo** and UNIX **df** commands produce the same information about a non-LFS partition.

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  
102400
```

```
LFS aggregate lfs2 (/dev/lfs2): 79957 K free out of total  
81920
```

Related Information

Commands: **fts lsaggr(8dfs)**

Files: **dfstab(4dfs)**

fts clone(8dfs)

fts clone

Purpose Creates 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer. The issuer must be logged into the 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.

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.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be backed up. You must remove its local mount point before attempting to clone the fileset.

Privilege Required

You must be listed in the **admin.ft** file on the machine where the read/write copy of the fileset is stored. You must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine where 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(8dfs)

fts clonesys

- Purpose** 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 periods) 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 server name as a DCE pathname. 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

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.

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.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be backed up. You must remove its local mount point before attempting to clone the fileset.

Privilege Required

The issuer must be listed in the **admin.ft** files on all machines where read/write versions of the filesets are stored. The issuer must also be listed in the **admin.fl** files

fts clonesys(8dfs)

on all Fileset Database machines or own the server entry for each machine where a version of any fileset to be backed up resides.

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 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 . (period)
- The - (dash)
- The _ (underscore)

The name must contain at least one alphabetic character or an _ (underscore) to differentiate its name from an ID number. The name can be no longer than 102 characters. This does not include the **.readonly** or **.backup** extension, which is added automatically when a read-only or backup fileset is created.

-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 server name as a DCE pathname.

-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.

fts create(8dfs)

- noauth** Directs **fts** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 installed, the **fts create** command can be 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. The *OSF DCE Administration Guide* provides instructions for configuring the root fileset to support replication.

Privilege Required

You must be listed in the **admin.ft** file on the machine specified by **-server**. You must also be listed on 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)**

Books: *OSF DCE Administration Guide*

fts crfldbentry(8dfs)

fts crfldbentry

Purpose 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 . (period)
- The - (dash)
- The _ (underscore)

The name must contain at least one alphabetic character or an _ (underscore) to differentiate it from an ID number. The name 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.)

-server *machine*

Names the File Server machine where the fileset resides. Specify the server name as a DCE pathname.

-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, the specified number must not already be in use in the **dfstab** file on the **-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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 **-fname** option to specify a name for the fileset according to the guidelines presented with the description of the **-fname** option. Use the **-server** option to indicate the File Server machine the aggregate or partition housing the fileset resides on. 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 also appear in the entry for the aggregate or partition in the **dfstab** file on **-server**.

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, the partition on which the fileset resides must be exported with the **dfsexport** command, and the fileset it contains must be mounted with the **fts crmount** command, before the fileset is

fts crfldbentry(8dfs)

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 the fileset resides on.

Privilege Required

You 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 Creates a mount point for a fileset

Synopsis **fts crmount -dir** *directory_name* [-fileset {*name* | *ID*} | -global] [-cell *cellname*]
[-rw] [-fast] [-help]

Options **-dir** *directory_name*
Names the point in the file tree where the root directory of the fileset is to be mounted (also known as the mount point). The specified name must not already exist. However, the parent directory of the mount point must exist in the DCE namespace.

-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 **-dir**. 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 is global. When preparing a fileset for use as the top-level fileset for diskless machines, use this option to create a mount point for the global namespace under the top-level diskless directory. The mount point for the global namespace is specified with **-dir**. Use this option only when preparing a fileset for use as the top-level diskless fileset. Use this option or use the **-fileset** option.

-cell *cellname* Specifies the cell where the fileset to be mounted is located. This option is necessary only if the fileset resides in a different cell from the parent directory of the mount point, in which case this option directs the command, and a Cache Manager requesting data from the fileset, to the other cell's Fileset Location Database (FLDB). A mount point of this type is referred to as a "cellular" mount point. If this option is omitted, the fileset to be mounted is assumed to reside in the same cell as the parent directory of the mount point.

-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 name the read/write version of the fileset. When the Cache Manager encounters a mount point during its search for a directory or file, it determines which fileset is associated

fts crmount(8dfs)

with the mount point. When it finds the root of that fileset, it traverses any paths leading through directories or other mount points in that fileset until it finds the indicated directory or file. The type of the mount point determines the version of a fileset through which the Cache Manager searches for the desired directory or file.

When the Cache Manager encounters a regular mount point, it checks the version of the fileset the mount point indicates. If the mount point indicates a read-only or backup version, the Cache Manager accesses that version. If it indicates the read/write version, the Cache Manager attempts to access a read-only fileset if the fileset in which the mount point resides is read-only.

If the regular mount point for 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.

If the regular mount point for a read/write fileset resides in a read-only fileset, the Cache Manager attempts to access a read-only version of the fileset first. If no read-only versions exist, the Cache Manager attempts to access the read/write version of the fileset. If one or more read-only versions exist but all are unavailable (perhaps because of one or more machine outages), the Cache Manager cannot access the fileset, because it does not attempt to access the read/write version.

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 where 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 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. 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 can still attempt 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 exist.

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. More information about using this command with the **-rw** option to create a read/write mount point for **root.dfs** is provided later in this reference page.

Note that cellular mount points function differently from regular and read/write mount points. If a cellular mount point indicates a read-only or backup version of a fileset, the Cache Manager accesses that version of the fileset in the other cell. However, if a cellular mount point indicates a read/write fileset, the Cache Manager attempts to access a read-only version of the fileset, if one exists, even if the mount point is located in a read/write fileset.

- fast** Indicates that the Fileset Location Server (FL Server) need not check if there is an FLDB entry for the fileset to be mounted. By default, the FL Server checks the FLDB and prints a warning if no FLDB entry exists for the fileset. The mount point is created in any case.
- help** Prints the online help for this command. All other valid options specified with this option are ignored.

Description

This command creates a mount point for the fileset specified by **-fileset** at the location in the file tree specified by **-dir**. The mount point makes the contents of the specified fileset visible to other users. Once this command is used to mount a fileset, the fileset never needs to be mounted again; it is mounted automatically.

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, it accesses the root directory of the fileset indicated by the mount point, after which it can access the files or directories in the associated fileset.

By default, every mount point is a regular mount point. However, if the **-rw** option is included with the command, the new mount point is a read/write mount point. Furthermore, the mount point and the fileset for which it serves as a mount point are assumed to reside in the same cell unless the **-cell** option is used to name a different cell; if the **-cell** option is used, the mount point is a cellular mount point. More information about the **-rw** and **-cell** options and their implications is provided earlier in this reference page.

During a cell's installation, 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

fts crmount(8dfs)

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 filesystem. 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 **dfsd** process automatically mounts it at the top-level of the cell's DFS filesystem (`/.../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 this command, the read/write version of **root.dfs** cannot be accessed in the cell. The *OSF DCE Administration Guide* provides instructions for configuring the root fileset to support replication.

Privilege Required

If the parent directory of the mount point (the directory where the mount point is to be created) is in a DCE LFS fileset, you must have write, execute, and insert permissions on the directory. If the directory is in a non-LFS fileset, you must have write and execute permissions on the directory.

Cautions It is recommended that a fileset not be mounted 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.

Examples The following command creates a regular mount point (the default type of mount point) for a read/write fileset named **user.jlw** at the directory named `/.../abc.com/fs/usr/jlw`:

```
$ fts crmount /.../abc.com/fs/usr/jlw user.jlw
```

This next command creates a regular mount point for a read/write fileset named **user.vijay**, which resides in the **def.com** cell, at the directory named `/.../abc.com/fs/usr/vijay` in the **abc.com** cell. The **-cell** option indicates that the fileset is located in a foreign cell, in this case **def.com**. When a user accesses a file

or directory under `../abc.com/fs/usr/vijay` (in the fileset named `user.vijay`), the Cache Manager contacts the FL Server in the cell named `def.com` to reach the fileset.

```
$ fts crmount ../abc.com/fs/usr/vijay user.vijay -cell def.com
```

The next two commands create mount points for the read/write fileset `diskless.root` and the global namespace under the top-level directory of the fileset. These steps are necessary if the fileset is to serve as the top-level diskless fileset (to be used as the root directory by diskless machines). The first command creates a mount point for the top-level diskless fileset at `../abc.com/fs/diskless.root`. The second command creates a mount point for the global namespace (`/...`) beneath the top-level directory of the diskless fileset.

```
$ fts crmount ../abc.com/fs/diskless.root diskless.root
```

```
$ fts crmount ../abc.com/fs/diskless.root/... -global
```

The final command creates a read/write mount point for the `root.dfs` fileset in the `abc.com` cell. The fileset is mounted at `.rw`, below the top-level of the cell's DFS filesystem.

```
$ fts crmount ../abc.com/fs/.rw root.dfs -rw
```

Related Information

Commands: `dfsd(8dfs)`, `fts create(8dfs)`, `fts delmount(8dfs)`, `fts lsmount(8dfs)`

Books: *OSF DCE Administration Guide*

fts crserverentry(8dfs)

fts crserverentry

Purpose Creates a server entry in the FLDB

Synopsis **fts crserverentry -server** {*machine* | *address*} **-principal** *name* [**-quota** *entries*]
[**-owner** *group*] [**-fxdid** *uuid*] [**-cell** *cellname*] [{**-noauth** | **-localauth**}] [**-verbose**]
[**-help**]

Options **-server** *machine*

Specifies the DCE pathname or network address of the server machine for which an entry is to be created in the Fileset Location Database (FLDB).

-principal *name*

Specifies the abbreviation for the DFS server principal to be registered in the FLDB for the machine. 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 group that is the owner of the server entry. If this option is omitted, a **NULL** value is used, meaning no group owns the server entry.

-fxdid *uuid* Specifies the object identifier for the server. If this option is omitted, a **NULL** value is used, meaning no object identifier is associated with the machine.

This option is not yet implemented.

-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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must

be logged into the 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.

fts crserverentry(8dfs)

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 DFS server principal of the machine is specified with the **-principal** option as **fs1**. Because they are omitted, the **-quota** and **-owner** options receive the default values of 0 (zero) and **NULL**, respectively.

```
$ fts crserverentry ../abc.com/hosts/fs1 hosts/fs1
```

Related Information

Commands: **fts delserverentry(8dfs)**, **fts edserverentry(8dfs)**, **fts lsserverentry(8dfs)**

fts delete

- Purpose** 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 to remove the fileset from. Specify the server name as a DCE pathname.
 - aggregate** *name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate to remove the fileset from. 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

fts delete(8dfs)**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 a read/write or backup version of a fileset is removed from is decremented once for each deleted version of the fileset.

Before you remove the read/write (and backup) version of a fileset, use the **fts rmsite** command to remove the read-only replicas of the fileset. If Release Replication was used for the fileset, use the **fts rmsite** command to remove the read-only replica stored at the read/write fileset's site as well. After removing a read/write or backup 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.

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.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be deleted. You must remove its local mount point before attempting to delete the fileset.

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(8dfs)

fts delfldbentry

Purpose 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 periods 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 server name as a DCE pathname. 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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

fts delfldbentry(8dfs)

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 syncflldb(8dfs)**, **fts syncserv(8dfs)**, **fts zap(8dfs)**

Files: **dfstab(4dfs)**

fts delmount

Purpose 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, you must have write, execute, and delete permissions on the directory it resides in. If **-dir** resides in a directory in a non-LFS fileset, you must have write and execute permissions on the directory it resides on.

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 Deletes a server entry from the FLDB

Synopsis **fts delserverentry** **-server** {*machine* | *address*} [**-cell** *cellname*]
[**{-noauth** | **-localauth**}] [**-verbose**] [**-help**]

Options

- server** {*machine* | *address*}
Specifies the DCE pathname or network address of the server machine whose entry is to be removed from the Fileset Location Database (FLDB).
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 lserverentry** 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

You 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 lserverentry(8dfs)**

fts dump

Purpose Converts a fileset to a bytestream format and place it in a file

Synopsis **fts dump -fileset** {*name* | *ID*} {-**time** {*date* | 0} | -**version** *number*} [-**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 used.

-**time** {*date* | -**version** *number*}

Specifies a full or incremental dump. There are three legal values:

- Causes a full dump of the current version of the fileset.
- (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 **10/7/89**. Only files with modification timestamps equal to or later than the specified date and time are dumped.
- An exact time and date value causes an incremental dump from the specified time on the indicated date. Only files with modification timestamps 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 “**10/7/89 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.

- file *filename*** Specifies the complete pathname of the file the dump is to be written to. 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer. The issuer must be logged into the 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 options on this command 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.

fts dump(8dfs)

- 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 does not affect the status of a fileset in the Fileset Location Database (FLDB) or at the site from which it is dumped. However, it does make a fileset inaccessible during the duration of the dump operation. Therefore, you may want to dump the backup version of a fileset to prevent the read-write and read-only versions from being inaccessible for an extended time.

The **fts restore** command can be used to restore a fileset dumped with this command.

Privilege Required

You must be listed in the **admin.ft** file on the machine the fileset is stored on. In addition, you must have the write, execute, and insert permissions on the directory the dump file is to reside in.

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 timestamps 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 restore(8dfs)**

fts edserverentry

Purpose Edits a server entry in the FLDB

Synopsis **fts edserverentry** **-server** {*machine* | *address*} [{**-rmaddr** | **-addaddr** *address* | **-changeaddr** *address*}] [**-principal** *name*] [**-quota** *entries*] [**-owner** *group*] [**-fxdid** *uuid*] [**-cell** *cellname*] [{**-noauth** | **-localauth**}] [**-verbose**] [**-help**]

Options **-server** {*machine* | *address*}

Specifies the DCE pathname or network address of the server machine whose entry in the Fileset Location Database (FLDB) is to be edited. Specify the network address if the **-rmaddr**, **-addaddr**, or **-changeaddr** option is used with the command.

-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.

fts edserverentry(8dfs)

- 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. The machine's principal name in the Registry Database must match this name. If this option is omitted, the 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. If this option is omitted, the group that currently owns the server entry retains ownership.
- fxdid** *uuid* Changes the object identifier for the server. If this option is omitted, the object identifier presently associated with the machine remains unchanged.
- This option is not yet implemented.*
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 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.

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 lserverentry** 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

You 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
```

fts edserverentry(8dfs)

Related Information

Commands: **fts crserverentry(8dfs)**, **fts delserverentry(8dfs)**, **fts lserverentry(8dfs)**

fts lock

Purpose 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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer. The issuer must be logged into the 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

fts lock(8dfs)

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

You must be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine a version of the fileset to be locked resides on.

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 unlockflldb(8dfs)**

fts lsaggr

Purpose 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 server name as a DCE pathname.
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

Enter the **dfsexport** command with no options on a File Server machine to list the aggregates and partitions with entries in the *dcelocal/var/dfs/dfsatab* file on that machine. If the file is properly maintained, the command should display an accurate listing of all exported aggregates and partitions. Use the **fts aggrinfo**

fts lsaggr(8dfs)

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.

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: **dfsatab(4dfs)**, **dfstab(4dfs)**

fts lsflldb

Purpose Shows information from fileset entries in the FLDB

Synopsis **fts lsflldb** [-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 server name as a DCE pathname. 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 on the server machine with locked FLDB entries. 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 on the aggregate with locked FLDB entries.

-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

fts lsfldb(8dfs)

- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

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 (with a **.readonly** or **.backup** extension, if appropriate).
- The fileset IDs of the read-write, read-only, and backup versions.
- 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 where 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) 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 where a version of the fileset resides, the name of the group that owns the server entry for the machine (or **<nil>** if no group owns the server entry), and the object identifier for the machine (or **<nil>** if no identifier is associated with the machine).

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**).

fts lsfldb(8dfs)

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
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  objid
fs3.abc.com  RW,BK  lfs1  0:00:00 hosts/fs3 <nil> <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 Lists fileset information from both the fileset header and the FLDB entry

Synopsis **fts lsft** [{**-path** {*filename* | *directory_name*} | **-fileset** {*name* | *ID*}}]
[**-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 containing 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 used.

Append the **.backup** or **.readonly** extension to the name of the fileset to list information about one of those versions of the fileset instead of the read-write version. This is useful if the read-write version no longer exists, in which case 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 containing the current working directory.
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

fts lsft(8dfs)

-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 lsfdb** 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 containing 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.

Use the **fts lsheader** command to display information from fileset headers. Use the **fts lsfdb** command to display information from fileset entries in the FLDB.

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**)
- Its size in kilobytes
- Additional information about the fileset and its access status
- Its status (**On-line**, **Off-line**, or an error indicator)
- The File Server machine, aggregate name, and aggregate ID number where 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
- The maximum quota, in kilobytes, of the read-write version 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)

It then displays the following information from the fileset's entry in the FLDB:

- The fileset's name (with a **.readonly** or **.backup** extension, if appropriate).
- The fileset IDs of the read-write, read-only, and backup versions.
- 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.
- 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 where a version of the fileset resides, the name of the group that owns the server entry for the machine (or <nil> if no group owns the server entry), and the object identifier for the machine (or <nil> if no identifier is associated with the machine).

fts lsft(8dfs)

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 5071 K states 0x4009 accStatus 0x0 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
MaxQuota 15000 K
Creation Tue Oct 15 16:45:16 1991
Last Update Fri Nov 22 11:36:00 1991
```

```
user.terry
```

```
readWriteID 0,,196953 valid
readOnlyID 0,,196594 invalid
backupID 0,,196595 valid
```

```
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	objid
fs3.abc.com	RW,BK	lfs1	0:00:00	hosts/fs3	<nil>	<nil>

Related Information

Commands: **fts lsfldb(8dfs)**, **fts lsheader(8dfs)**

fts lsheader

Purpose 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 server name as a DCE pathname. 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

fts lsheader(8dfs)

-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 lsflldb** 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 lsflldb** command.

Privilege Required

You 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 following information is displayed for DCE LFS filesets. Because non-LFS filesets do not have DCE LFS fileset headers, this 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 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)
- Its fileset ID number
- Its type (**RW** for read-write, **RO** for read-only, or **BK** for backup)
- Its size in kilobytes
- Its status (**On-line**, **Off-line**, or an error indicator)
- The total number of filesets on-line, the total number of filesets off-line, and the total number of filesets busy

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
- Additional information about the fileset and its access status
- 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 maximum quota, in kilobytes, of the read-write version 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(8dfs)

```

$ 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 On-line
user.wvh            0,,196956 RW      4955 K 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 5071 K states 0x4009 accStatus 0x0 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
    MaxQuota 15000 K
    Creation Tue Oct 15 16:45:16 1991
    Last Update Fri Nov 22 11:36:00 1991

user.wvh 0,,196956 RW LFS 4955 K states 0x9 accStatus 0x0 On-line
.
.
Total filesets on-line 15; total off-line 1;
total busy 0

```

Related Information

Commands: **fts lsfdb(8dfs)**, **fts lsft(8dfs)**

Files: **dfstab(4dfs)**

fts lsmount

Purpose 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 **-dir** 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.

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'
```

In the output, *directory_name* is the name of a directory you specified with the **-dir** option; *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

A cellname and : (colon)

Precede *fileset_name* if *directory_name* is a cellular mount point; a # (pound sign) or % (percent sign) precedes the cellname

fts lsmount(8dfs)

! (exclamation point)

Replaces *fileset_name* if the directory is a global mount point

The **fts lsmount** command displays the following message for each directory that is not a mount point:

'directory_name' is not a mount point.

Privilege Required

You must have read permission on each **-dir** used in the command, regardless of whether the directory in which **-dir** resides is in a DCE LFS or non-LFS fileset.

Related Information

Commands: **fts delmount(8dfs)**

fts lsreplicas

Purpose Displays the status 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**
Names a specific File Server machine where replicas of **-fileset** are to be checked. Specify the server name as a DCE pathname. 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 status 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:

fts lsreplicas(8dfs)

- 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.

Examples The following command displays the status of each replica of the read/write fileset named **osf1_pmax.bin**:

```
$ fts lsr osf1_pmax.bin -a
```

Related Information

Commands: **fts addsite(8dfs)**, **fts release(8dfs)**, **fts rmsite(8dfs)**, **fts statrepserver(8dfs)**, **fts update(8dfs)**

fts lsserverentry

Purpose 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 server name as a DCE pathname. 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

fts lserverentry(8dfs)

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.

Examples The following command displays the server entry from the FLDB for a server machine named **fs1**:

```
$ fts lserverentry ../../abc.com/hosts/fs1
```

Related Information

Commands: **fts crserverentry(8dfs)**, **fts delserverentry(8dfs)**, **fts edserverentry(8dfs)**

fts move

Purpose 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 a read/write fileset to be moved.
- fromserver** *source_machine*
Names the File Server machine where the fileset currently resides. Specify the server name as a DCE pathname.
- fromaggregate** *source_name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate the fileset currently resides on. 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 move the fileset to. Specify the server name as a DCE pathname.
- toaggregate** *dest_name*
Specifies the device name, aggregate name, or aggregate ID of the aggregate to move the fileset to. 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the machine as **root** for this option to work. If you use this option, do not use the **-noauth** option.

fts move(8dfs)

- 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 **-fromserver** and **-fromaggregate**) to the destination site (specified with **-toserver** and **-toaggregate**). The command decrements the number of fileset entries recorded as residing on **-fromserver** in the Fileset Location Database (FLDB) entry for the File Server machine, and it increments the number of fileset entries recorded as residing on **-toserver** in the FLDB entry for that File Server machine. It also automatically removes the backup copy of the fileset from the current site, if it exists. To create a new backup 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 and issuing the **fts clone** command.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be moved. You must remove its local mount point before attempting to move the fileset.

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.

Privilege Required

You must be listed in the **admin.ft** files on both the source and destination machines. You 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 replicas of the fileset reside on. 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(8dfs)

fts release

Purpose Initiates Release Replication by placing a read-only version of a read/write DCE LFS fileset at the local site

Synopsis `fts release -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 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.
- noauth** Directs **fts** to use the unprivileged identity **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 Release Replication process. It places a read-only copy of the read/write DCE LFS fileset specified with the **-fileset** option at the local replication site defined at the same File Server machine as the read/write fileset. The Replication Servers (**repserver** processes) at each of the fileset's replication sites (specified File Server machines and aggregates) then place read-only replicas of the copy at the sites on their respective machines.

Before this 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 when the **fts setrepinfo** command is used to define the fileset's replication parameters. The **fts addsite** command must also be used to define the replication sites for the read/write fileset. (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 alternative to Release Replication is Scheduled Replication, which automatically places replicas of a read/write fileset at its replication sites according to the fileset's replication parameters. (Additional replication parameters must be set with the **fts setrepinfo** command if Scheduled Replication is to be used.) The **fts update** command can be used to request an immediate update of read-only replicas, regardless of the type of replication used (Release or Scheduled).

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.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be replicated. You must remove its local mount point before attempting to replicate the fileset.

Privilege Required

You must be listed in the **admin.ft** file on the machine where the source read/write fileset is stored. You must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entries for the machine the source fileset resides on and all machines the read-only replicas are to reside on.

Examples The following command initiates Release Replication for the read/write fileset named **osf1_pmax.bin**:

```
$ fts release osf1_pmax.bin
```

Related Information

Commands: **fts addsite(8dfs)**, **fts lsreplicas(8dfs)**, **fts setrepinfo(8dfs)**, **fts statrepserver(8dfs)**, **fts update(8dfs)**

fts rename(8dfs)

fts rename

Purpose 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 . (period)
- The - (dash)
- The _ (underscore)

The name must contain at least one alphabetic character or an _ (underscore) to differentiate its name from an ID number. The name can be no longer than 102 characters. This does not include the **.readonly** or **.backup** extension, which is added automatically when a read-only or backup fileset is created.

-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 **anonymous** 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 identity (principal name) of the machine on which the command is issued as the identity of the issuer. The issuer must be logged into the 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 **fts delmount** and creating a replacement for each with **fts crmount**. (These commands require that the issuer have the delete, write, and insert permissions on any directories that mount points are removed from or that they are created in.)

Privilege Required

You must be listed in the **admin.ft** file on the machine where the read/write fileset resides. You must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine a version of the fileset to be renamed resides on.

Examples The following command changes the incorrect fileset name **osf1.bin** to the correct fileset name **osf1_pmax.bin**:

```
$ fts rename osf1.bin osf1_pmax.bin
```

Related Information

Commands: **fts crmount(8dfs)**, **fts delmount(8dfs)**

fts restore(8dfs)

fts restore

Purpose Converts a file from a 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 . (period)
- 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.

-server *machine*

Specifies the File Server machine the file is to be restored to. Specify the server name as a DCE pathname.

-aggregate *name*

Specifies the device name, aggregate name, or aggregate ID of the aggregate or partition on **-server** the file is to be restored to. 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 sparingly and with great care.*
- 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. Use this option only to overwrite a previously restored version of a fileset with a file containing 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.
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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**.

fts restore(8dfs)

The fileset contained in the dump file can be restored as a new read/write fileset by specifying a new name and, optionally, a new site for the 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 rather than overwriting the existing fileset.

The fileset to be overwritten must initially have been restored as a new read/write fileset from a full dump. Also, the dump file to be restored must be an incremental dump. (A full dump of a fileset cannot be restored to overwrite an existing fileset, even one of the same name.) Both the incremental and the full dump that initially produced the read/write fileset to be overwritten must be dumps of the same fileset.

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.

A fileset ID number can be assigned to any 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 fileset retains its current ID number, or the FL Server allocates a new ID number for a new 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.

A fileset can be restored to a non-LFS file system different from the one it was dumped from. However, a non-LFS fileset cannot be restored as a DCE LFS fileset, and vice versa.

If an existing 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. 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.

Cautions Ensure that all of the conditions discussed earlier are met before you restore 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.

Privilege Required

You must be listed in the **admin.ft** file on the machine specified by **-server** and must have the read permission on the dump file. You must also be listed in the **admin.fl** files on all Fileset Database machines or own the server entry for each machine where a version of the fileset is recorded as residing in the FLDB (generally only **-server** unless an existing fileset is to be overwritten).

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

fts restore(8dfs)

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 server name as a DCE pathname.
 - 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.
 - 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

fts rmsite(8dfs)**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 removes the definition of the replication site from the Fileset Location Database (FLDB) entry for the fileset. It also removes the replica of the read/write fileset stored at the removed site and decrements the number of fileset entries recorded as residing on the File Server machine specified with **-server** in the FLDB entry for the server.

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 the **fts delete** command is used to remove the read/write (and backup) version of a fileset, use the **fts rmsite** command to remove the read-only replicas of the fileset. If Release Replication was used for the fileset, use the **fts rmsite** command to remove the read-only replica stored on the same File Server machine as the read/write fileset as well.

Replication sites are added with the **fts addsite** command.

Privilege Required

You 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 that houses a version of the fileset for which the replication site and replica are to be removed.

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 **osf1_pmax.bin**. A replica of **osf1_pmax.bin** that resides at the site is also removed.

```
$ fts rmsite osf1_pmax.bin /.../abc.com/hosts/fs5 /dev/lv01
```

Related Information

Commands: **fts addsite(8dfs)**, **fts delete(8dfs)**, **fts setrepinfo(8dfs)**

Files: **dfstab(4dfs)**

fts setquota

- Purpose** 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 the fileset can occupy. Specify the value in 1-kilobyte blocks. (A value of 1024 kilobytes is 1 megabyte.) By default, every newly created fileset has a maximum 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 maximum size of the quota 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

fts setquota(8dfs)

indicated by specifying the name of a file or directory on the fileset with the **-path** option or by indicating the fileset directly with the **-fileset** option.

By default, every newly created fileset has a maximum quota of 5000 kilobytes. This command increases or decreases a fileset's maximum quota to be 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 maximum quota for a fileset.

Privilege Required

You must be listed in the **admin.ft** file on the machine the fileset is stored on.

Examples The following command sets the quota for the fileset containing the directory named **/usr/terry** to be 15,000 kilobytes:

```
$ fts setq /usr/terry 15000
```

Related Information

Commands: **fts lsft(8dfs)**, **fts lsheader(8dfs)**, **fts lsquota(8dfs)**

fts setrepinfo

Purpose 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 **-fileset**. When initially defining a fileset's replication parameters, use this option or use **-scheduled**. 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 **-fileset**. When initially defining a fileset's replication parameters, use this option or use **-release**. 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 **-fileset** 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

fts setrepinfo(8dfs)

Scheduled. Omit this option when specifying **-release** or **-scheduled** to initially set a fileset's replication type. Also omit it when changing a fileset's replication parameters without changing its replication type.

-maxage interval

Specifies the maximum amount of time for which the Cache Manager distributes data without verifying that it is still current (MaxAge). The default is 2 hours. An effective value must be greater than or equal to 2 minutes. This is applicable to Release and Scheduled Replication.

-failage interval

Specifies how long the Cache Manager continues to provide data before it fails to do so because the data is outdated (FailAge). The Cache Manager fails to use cached data if the data is older than this value and the Cache Manager cannot obtain more current data. The default is 1 day or twice the MaxAge, whichever is larger. An effective value must be greater than or equal to **-maxage**. This is 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; that is, those not referenced by any 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. The Cache Manager sends keep-alive messages more frequently than the ReclaimWait interval to prevent the File Exporter from reclaiming storage space occupied by deleted files. The default is 18 hours. An effective value must be greater than 2 hours; do not specify a value less than 90 minutes. This is 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 currency

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. This is 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`. This is 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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

fts setrepinfo(8dfs)

be defined for the fileset with the **fts addsite** command and before the **fts release** or **fts update** commands 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 closely tracked.
- 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** parameter.

The **-maxage**, **-failage**, **-reclaimwait**, **-minrepdelay**, and **-defaultsiteage** options are used to set the corresponding replication parameters for a read/write fileset. Information on the replication parameter each option sets is provided earlier in this reference page. The following lists describe the dependencies between the different options when they are used to set the replication parameters for either Release or Scheduled Replication.

The following options are used to define the parameters for Release Replication. (The **-minrepdelay** and **-defaultsiteage** options do not apply for Release Replication.)

-maxage Required only if **-failage** is specified. Otherwise, the default is 2 hours.

- failage** Optional. If it is specified, both **-maxage** and **-reclaimwait** are required. The default is 1 day or twice the MaxAge, whichever is larger.
- reclaimwait** Required only if **-failage** is specified. Otherwise, the default is 18 hours.

The following options are used to define the parameters for Scheduled Replication:

- maxage** Required only if **-failage**, **-minrepdelay**, or **-defaultsiteage** is specified. Otherwise, the default is 2 hours.
- failage** Required only if **-minrepdelay** or **-defaultsiteage** is specified. Otherwise, the default is 1 day or twice the MaxAge, whichever is larger.
- reclaimwait** Required only if **-failage**, **-minrepdelay**, or **-defaultsiteage** is specified. Otherwise, the default is 18 hours.
- minrepdelay** Required only if **-failage** or **-defaultsiteage** is specified. Otherwise, the default is 5 minutes or one-quarter of the DefaultSiteAge, whichever is smaller.
- defaultsiteage** Always optional. The default is one-quarter of the MaxAge. However, if the other four options are specified and **-defaultsiteage** is not, the **-maxsiteage** option must be specified when defining replication sites for the read/write fileset with the **fts addsite** command.

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 **-defaultsiteage** 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.

fts setrepinfo(8dfs)

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 the unit abbreviations (**d**, **h**, **m**, and **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 lsfdb** or **fts lsft** command to display the replication parameters associated with a read/write fileset. Use the **fts lsreplicas** command to display the status of replicas at replication sites. Use the **fts statrepserver** command to display the status of the Replication Server on a File Server machine.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be replicated with either method of replication. You must remove its local mount point before attempting to replicate the fileset.

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 when the cell was configured; and **root.dfs** is replicated. Refer to the *OSF DCE Administration Guide* for information on configuring **root.dfs** to support replication.

Privilege Required

You must be listed in the **admin.ft** file on the machine where the fileset resides. You 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.

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 **osf1_pmax.bin**. The default replication parameters are used for the fileset.

```
$ fts setrepinfo -fileset osf1_pmax.bin -release
```

The following command changes the replication type for the **osf1_pmax.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 osf1_pmax.bin -scheduled -change -clear
```

The following command clears the current replication parameters used for the **osf1_pmax.bin** fileset, replacing them with parameters specified by the issuer of the command:

```
$ fts setr osf1_pmax.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 lsfdb(8dfs)**, **fts lsft(8dfs)**, **fts lsreplicas(8dfs)**, **fts release(8dfs)**, **fts statrepserver(8dfs)**, **fts update(8dfs)**

Books: *OSF DCE Administration Guide*

fts statftserver(8dfs)

fts statftserver

Purpose 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 server name as a DCE pathname.
- 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 **anonymous** 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 identify (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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

You must be listed on the **admin.ft** file on the machine specified by **-server**.

Related Information

Commands: **ftserver(8dfs)**

fts statrepsrver

Purpose Displays the status of a Replication Server process

Synopsis **fts statrepsrver** *-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 server name as a DCE pathname.
- long** Specifies that more detailed information about the Replication Server is to be displayed.
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 statrepsrver** command displays information about the status of the Replication Server (**repsrver** 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. Use the **fts lsreplicas** command to check the statuses of fileset replicas.

Related Information

Commands: **fts lsreplicas(8dfs)**, **repsvr(8dfs)**

fts syncfldb

Purpose Synchronizes FLDB entries with fileset headers at a particular site

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 server name as a DCE pathname.
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 filesets residing on the File Server machine specified by **-server**. It checks either all of the filesets on **-server** or only the filesets on the optionally specified **-aggregate**. It checks that the FLDB correctly records every fileset whose fileset header is marked *On-line*. It changes any necessary FLDB entries to be consistent with the status of each fileset on **-server** and optionally **-aggregate**.

This 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 removes the backup from the site.
- If it finds a fileset ID number that is larger than the value of the counter used by the File 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.

It is recommended that the **fts syncserv** command be run on all File Server machines in a cell *after* the **fts syncfldb** command is run on all File Server machines in the cell. However, nothing prohibits the commands from being executed in 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 proper commands then need 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**). You

fts syncfdb(8dfs)

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.

Related Information

Commands: **fts addsite(8dfs)**, **fts setrepinfo(8dfs)**, **fts syncserv(8dfs)**

Files: **dfstab(4dfs)**

fts syncserv

- Purpose** Synchronizes fileset headers at a particular site with 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 server name as a DCE pathname.
 - 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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.

fts syncserv(8dfs)**Description**

This command finds and inspects the FLDB entries for filesets (read-write, read-only, and backup) residing on the File Server machine specified by **-server**. It checks either all of the filesets on **-server** or only the filesets on the optionally specified **-aggregate**. It checks that the FLDB entry for each fileset is consistent with the state of the fileset at the specified site, including the reported sites of all copies (even though that requires checking filesets on server machines other than **-server**). It also increments or decrements the number of fileset entries recorded as residing on File Server machines in the FLDB entries for the servers, as necessary.

It is recommended that the **fts syncfdb** 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 syncfdb** 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 setreinfo** and **fts addsite** commands.

Because non-LFS filesets do not have fileset headers, the **fts syncserv** and **fts syncfdb** 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 an error 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

You 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**). You 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 syncfdb(8dfs)**

Files: **dfstab(4dfs)**

fts unlock

Purpose 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 unlockfdb** command to unlock multiple filesets at one time.

Privilege Required

You 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 unlockfdb(8dfs)**

fts unlockfldb

Purpose 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 server name as a DCE pathname. 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** where 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 **anonymous** 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 identity (principal name) of the machine where the command is issued as the identity of the issuer. The issuer must be logged into the 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 lock 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

You 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(8dfs)

fts update

- Purpose** Requests an immediate update of replicas of a read/write DCE LFS fileset
- 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. The fileset can use Release or Scheduled Replication.
 - all**
Specifies that all replicas of **-fileset** are to be updated. Use this option or use **-server**.
 - server** *machine*
Names a specific File Server machine on which the replica of **-fileset** to be updated is stored. Specify the server name as a DCE pathname. 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 **anonymous** 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 identity (principal name) of the machine where the command is issued as the identity of the issuer. The issuer must be logged into the 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. It

can be used to update replicas created with Release or Scheduled Replication. Use the command's options to update replicas of **-fileset** as follows:

- To update the replica stored on a specific File Server machine, specify the name of the machine with the **-server** option.
- To update all replicas, specify the **-all** option.

When this command is used to update replicas of a fileset that uses Scheduled Replication, it requests that the Replication Server perform an immediate update of the specified replicas based on the current read/write version of the fileset. When it is used to update replicas of a fileset that uses Release Replication, it forces the Replication Server to copy the read-only replica stored on the same File Server machine as the read/write fileset to the replicas at the specified replication sites. This command does not change the replication type and parameters defined for the read/write fileset.

Before this 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. If Release Replication is used, the first replication site created with the **fts addsite** command must be on the same File Server machine as the read/write fileset, after which the **fts release** command is used to place a replica at the site.

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.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be replicated with either method of replication. You must remove its local mount point before attempting to replicate the fileset.

Examples The following command requests an immediate update of the replica of the read/write fileset named **osf1_pmax.bin** at the replication site defined at the File Server machine named **fs3**:

```
$ fts update osf1_pmax.bin /.../abc.com/hosts/fs3
```

Related Information

Commands: **fts addsite(8dfs)**, **fts lsreplicas(8dfs)**, **fts release(8dfs)**, **fts setrepinfo(8dfs)**, **fts statrepserver(8dfs)**

fts zap

Purpose 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 valid.
- server** *machine* Names the File Server machine from which to remove the fileset. Specify the server name as a DCE pathname.
- 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 **anonymous** 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 identity (principal name) of the machine the command is issued from as the identity of the issuer. The issuer must be logged into the 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 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 where it is important to delete the fileset but, for some reason, the FLDB is unreachable; for example, if the FL Server is unavailable. The issuer can remove the FLDB entry later with the **fts rmsite** or **fts delfdbentry** commands, or it can be removed automatically by running the **fts syncserv** and **fts syncfdb** commands.

A DCE LFS fileset that is mounted locally (as a file system on its File Server machine) cannot be deleted. You must remove its local mount point before attempting to delete the fileset.

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 delfdbentry(8dfs)**, **fts rmsite(8dfs)**, **fts syncfdb(8dfs)**, **fts syncserv(8dfs)**

Files: **dfstab(4dfs)**

ftserver(8dfs)

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 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 **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 **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 prints error messages to the standard error output. The **ftserver** keeps an event log file in *dcelocal/var/dfs/adm/FtLog*.

Related Information

Commands: **dfsbind(8dfs)**, **fts statftserver(8dfs)**, **fxd(8dfs)**

Files: **admin.ft(4dfs)**, **FtLog(4dfs)**

fxd

Purpose Initializes the File Exporter and start associated kernel daemons

Synopsis **fxd -admingroup** *group* [**-mainprocs** *number_of_background_daemons*]
 [**-tokenprocs** *number_of_token_daemons*] [**-verbose**] [**-help**]

Options **-admingroup** *group*
 Specifies 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. 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. File Exporter kernel daemons are responsible for receiving and servicing RPC requests from DFS clients. 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. File Exporter kernel daemons are responsible for issuing token revocation RPC requests to DFS clients. If this option is omitted, two main kernel daemons issue token revocation requests.

-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 of 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, that the File Exporter requires. It also passes connection-state information (such as cell name, Fileset Database machine location, and server key information) to the File Exporter. The File Exporter uses this information to communicate with other processes such as Fileset Location Servers.

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 **-mainprocs** and **-tokenprocs** options can be used to alter the default number of main kernel daemons running on the server machine as follows. On most system types, these daemons appear as nameless entries in the output of the **ps** command.

- mainprocs** Specifies the number of main kernel daemons that run on the machine to service RPC requests 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 if the machine is to support an unusually large number of DFS clients.
- tokenprocs** Option specifies the number of main kernel daemons dedicated to issuing token revocation RPC requests to DFS clients. The daemons also issue messages to client machines to verify that the client machines are still active. The default number of main 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 daemons dedicated to token revocation accordingly.

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

fxd(8dfs)

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 **-adminingroup** 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 **-adminingroup** options of all **fxd** commands issued in a domain.

When it is issued, the **fxd** command creates the *dcelocal/var/dfs/adm/FileLog* event log if the file does not already exist. The File Exporter then appends to the file any messages it generates. If the file exists when the **fxd** command is run, the command moves it to the **FileLog.old** file in the same directory (overwriting the current **FileLog.old** file if it exists) before creating a new version the File Exporter can append messages to.

Privilege Required

You must be logged in as **root** on the local machine.

Output If problems are encountered during initialization, the **fxd** process prints error messages to the standard error output. The File Exporter keeps an event log in *dcelocal/var/dfs/adm/FileLog*.

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
```

Related Information

Commands: **dfsbind(8dfs)**, **ftserver(8dfs)**

Files: **FileLog(4dfs)**

growaggr

Purpose 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. A relative pathname specified with this option is taken relative to */dev/*. 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 where 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 it resides on.

-noaction

Used without the **-aggrsize** option, this option directs the command to display the total number of 1024-byte blocks on the disk partition the specified aggregate resides on. 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.

growaggr(8dfs)

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 of 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 the aggregate resides on. 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 the aggregate resides on.
- 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 the aggregate resides on 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, you must be logged in as **root** on the local machine or must be able to write to the device (disk partition) on which the specified aggregate resides. If the **-noaction** option is included with the command, you only need to be able to read the device the aggregate resides on.

Related Information

Commands: **newaggr(8dfs)**

Files: **dfstab(4dfs)**

newaggr

Purpose Initialize 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 percent 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. 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.

-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.

newaggr(8dfs)

-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 of 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 in the partition where 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.

The DCE LFS manages blocks and fragments as variable-length containers for the storage of user and system data. The DCE LFS 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. The 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, one percent 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.

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.

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.

newaggr(8dfs)

Cautions Do not use the **newaggr** command to create non-LFS aggregates. Also, 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. Finally, do not use the command on a locally mounted partition; doing so causes the kernel to panic.

Privilege Required

If the **-noaction** option is not included with the command, you must be logged in as **root** on the local machine or must be able to write to the device (disk partition) to be initialized as a DCE LFS aggregate. If the **-noaction** option is included with the command, you only need to be able to read the device.

Related Information

Commands: **dfsexport(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*] [-**verbose**] [-**help**]
- Options**
- mainprocs** *number_of_background_daemons*
 Specifies the number of background daemons to run on the machine. These daemons are responsible 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 incoming token revocation RPC requests from DFS servers. If this option is omitted, four background daemons service token revocation requests.
 - 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 **repserver** command. See the **bos help** and **bos apropos** reference pages for examples of 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*.

repsvr(8dfs)

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 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 incoming token revocation RPC requests from DFS servers. 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, use the **-tokenprocs** option to increase the number of background daemons dedicated to token revocation.

When it is started, the **repsvr** 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 the **repsvr** 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 append messages to.

Use the **fts statrepsvr** 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, the **repsvr** prints error messages to the standard error output. The **repsvr** keeps an event log in *dcelocal/var/dfs/adm/RepLog*.

Related Information

Commands: **fts lsreplicas(8dfs)**, **fts statrepsvr(8dfs)**

Files: **RepLog(4dfs)**

salvage

Purpose Uses the DFS Salvager to verify, recover, or repair the structure of an aggregate

Synopsis `salvage -aggregate name [-verify] [-recover] [-norecover] [-verbose] [-nosalvage] [-help]`

Options

- aggregate *name*** Specifies the device name or aggregate name of the DCE LFS aggregate to be repaired or verified. These names are specified in the first and second fields of the entry for the aggregate in the *dcelocal/var/dfs/dfstab* file.
- verify** Directs the Salvager to examine the structure of the specified aggregate to determine if it contains any inconsistencies. The Salvager reports any inconsistencies it finds, but it does nothing to correct them. Use this option with **-recover** to run recovery on the aggregate and then examine its structure without attempting to repair any inconsistencies.
- recover** Directs the Salvager to run the recovery procedure on the log it finds on the aggregate before beginning any other operation. Recovery is the replaying of the log on the aggregate. Use this option with **-verify** to run recovery on the aggregate and then examine its structure without attempting to repair any inconsistencies.
- norecover** Directs the Salvager not to run recovery on the aggregate. Use this option alone to salvage the aggregate by repairing any inconsistencies on it.
- verbose** Directs the Salvager to produce detailed information about the aggregate. The information is useful primarily for debugging purposes. It is displayed on standard output (**stdout**) unless it is piped elsewhere. Use this option alone or with any other combination of options.
- nosalvage** Directs the Salvager not to salvage the aggregate; the Salvager does not attempt to repair any inconsistencies on the aggregate. Use this option alone to run recovery on the aggregate without attempting to determine or repair any inconsistencies.
- help** Prints the online help for this command. All other valid options specified with this option are ignored.

salvage(8dfs)

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 of using these commands.

Description

The **salvage** command invokes the DFS Salvager to analyze or repair any internal inconsistencies found in the structure of the DCE LFS aggregate specified with the **-aggregate** option. The DCE LFS log mechanism can usually return consistency to a file system after a system failure. However, the Salvager needs to be used if problems are detected in the basic structure of the DCE LFS aggregate or if the log mechanism is damaged. The binary file for the **salvage** command resides in *dcelocal/bin/salvage*.

This command can be used to verify, recover, or repair the structure of an aggregate. Specify the command's **-verify**, **-recover**, **-norecover**, and **-nosalvage** options as follows to indicate the operations the Salvager is to perform on the specified aggregate:

- verify** To determine whether the structure of the aggregate contains any inconsistencies without running recovery (replaying the log) or attempting to repair any inconsistencies found on the aggregate. Use this option to assess the extent of the damage to the aggregate.
- verify and -recover** To run recovery on the aggregate and then analyze the structure of the aggregate without repairing any inconsistencies. 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.
- nosalvage** To run recovery on the aggregate without attempting to determine or repair any inconsistencies found on the aggregate. Use this option to quickly return consistency to an aggregate that does not need to be salvaged.
- norecover** To attempt to repair any inconsistencies found in the structure of the aggregate without first running recovery. Use this option if you believe replaying the log will not return consistency to the aggregate and may in fact further damage it.
- No options To run recovery on the aggregate and then attempt to repair any inconsistencies found in the structure of the aggregate. Specify no options if you believe the log should be replayed before attempts are made to repair any inconsistencies found on the aggregate.

The basic function of the Salvager is similar to that of the UNIX **fsck** program. However, the Salvager prompts for additional information only if it believes the specified aggregate does not contain a DCE LFS file system; the operation can then be canceled or continued. If it is continued and the aggregate does not contain a DCE LFS aggregate, the Salvager exits with an error code of at least 16 without analyzing the partition. Otherwise, the Salvager continues with the requested operation without prompting for any additional information or pausing to verify any changes before it makes them.

The Salvager analyzes and repairs problems detected in the structure of a DCE LFS aggregate. It does not verify or repair the format of the data contained in the files on the aggregate. If changes are required, 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 blocks and anodes in each fileset on that aggregate. An anode is an area on the disk that provides information used to locate data and other anodes. By following the links between the various types of anodes, the Salvager can make reasonable assumptions about whether the organization of an aggregate and the filesets it contains is correct.

Not all aggregates can be salvaged. In cases of extensive damage to the structure of the metadata on the aggregate or damage to the physical disk housing the aggregate, the Salvager cannot repair the inconsistencies. Also, the Salvager cannot verify or repair damage to data on the aggregate. The Salvager cannot detect problems that modified the contents of a file without damaging the structure of the aggregate or changing the metadata logged for 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.

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 filesets, the Salvager marks as "unsafe" anodes where it encounters problems. The Salvager later attempts to determine the actual fileset pathnames associated with these anodes and include the pathnames in the lists. When it has finished salvaging, the Salvager displays any nonempty lists.

The Salvager also maintains a log file in *dcelocal/var/dfs/adm/SalvageLog* to which it writes output about the aggregate as it executes. The Salvager also displays one of a number of informative exit codes depending on the

salvage(8dfs)

inconsistencies it found and the repairs it made. More information about the lists and exit codes displayed by the Salvager is provided later in this reference page.

The internal structures maintained by the Salvager require a minimum of 1 megabyte of swap space. However, the 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

If only the **-verify** option is included with the command, you only need the **read** permission for the specified device (aggregate). If the **-recover**, **-norecover**, or **-nosalvage** option is included with the command, or if all options are omitted from the command, you must be logged in as **root** on the local machine or must own the specified device (aggregate).

Cautions

Never attempt to salvage an aggregate that houses mounted filesets or that is currently exported. In most cases, filesets cannot be mounted on an aggregate in need of salvaging. However, verify that the filesets on an aggregate are unmounted, and unmount them if necessary, before attempting to salvage the aggregate. Similarly, detach an aggregate from the global namespace before salvaging it.

The Salvager can be used to salvage only DCE LFS aggregates. If it is executed on a non-LFS partition, it exits with an error code of at least 16 without analyzing the disk. Use the UNIX **fsck** program or its equivalent to verify or restore consistency to non-LFS disk partitions.

Output

The Salvager displays output on the screen and sends the same output to the *dcelocal/var/dfs/adm/SalvageLog* file. When the Salvager is executed, it displays the name of the aggregate it is salvaging. If the **-verbose** option is specified with the command, the Salvager also 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

When the Salvager has finished executing, it displays any of the following lists that contain the names of one or more files:

Ought to Restore

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.

May Restore Files to which modifications were made (for example, files whose ACLs or property lists were changed). The user should verify the contents of these files, 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.

dupAnode Files that referenced fragments or blocks of data that were also referenced by other files.

badACLs Files whose ACLs were modified or deleted.

badPlists Files whose property lists (plists) were modified or deleted.

zeroLinkCnt Files that referred to files whose link counts should be 0 (zero). This list is displayed only if the zero link list is incorrect.

badLinkCnts Files whose link counts were inconsistent with the number of references found to the file.

In addition, the Salvager returns one of various exit codes summarizing its actions and findings. Four bits are used to indicate the state of the aggregate. The larger the exit code, the greater the severity of the problems on the aggregate. An exit code made up of the following bits can be returned. (The following exit codes are displayed in hexadecimal format.)

0x0 (all bits off)

The command found no problems. The message Done is also displayed.

0x1

The command found one or more problems. The messages Some inconsistencies found and Done are also displayed. Run the command on the aggregate without the **-verify** option to attempt to correct the problems.

salvage(8dfs)

- 0x3** The command found one or more problems and fixed them. The messages `Some inconsistencies found while salvaging` and `Done` are also displayed.
- At least **0x10** Some serious problem (for example, the aggregate on which the salvage was attempted is not a DCE LFS aggregate) prevents the Salvager from running on the aggregate. Attempt to determine the cause of the problem.
- At least **0x8** The command found the aggregate to be irreparably damaged. Use the **newaggr** command to reinitialize the aggregate, and reconstruct the data from existing backups if possible.
- At least **0x4** The command found one or more problems and fixed some of them. However, some of the problems were more severe and require a subsequent salvage to be repaired. Run the command on the aggregate without the **-verify** option again to attempt to correct the problems.

Including the **-verbose** option with the command produces more detailed information about the aggregate being salvaged. However, the additional information is useful primarily for debugging purposes.

Examples The following command instructs the Salvager to analyze the structure of the aggregate whose device name is `/dev/lv01` 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 -norecover
```

Related Information

Commands: **newaggr(8dfs)**

Files: **dfstab(4dfs)**, **SalvageLog(4dfs)**

scout

Purpose Initializes the Scout program

Synopsis **scout -server** *machine...* [**-basename** *common_prefix*] [**-host**]
 [**-frequency** *seconds*] [**-attention** *stat/threshold_pair...*] [**-debug** *filename*] [**-help**]

Options **-server** *machine*

Names each File Server machine whose File Exporter is to be monitored. If **-basename** is not specified, provide the DCE pathname of each machine (for example, */.../abc.com/hosts/fs1*). If **-basename** is specified, provide only the unique suffix of each machine name (for example, *fs1*).

-basename *common_prefix*

Specifies the DCE pathname prefix (for example, */.../abc.com/hosts*) common to the File Server machines specified with **-server**. Do not include the / (slash) separating the prefix from the unique part of each machine name. If a basename is specified with this option, it is displayed in the banner line.

-host

Displays the name of the machine running **Scout** 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 **Scout** 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). **Scout** 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 the File Exporter can have open to client machines before the value is highlighted.

fetch *bytes_fetched*

The maximum number of bytes of data clients can

scout(8dfs)

fetch from the File Exporter before the value is highlighted. The number of bytes fetched is accumulated from the time the File Exporter is started. The highlighting is removed when the File Exporter is restarted, at which time the value returns to 0 (zero). Enter a threshold for this statistic in 64 kilobyte units. For example, to have **Scout** highlight this value when it equals or exceeds 128 kilobytes, specify a threshold of 2.

store bytes_stored

The maximum number of bytes of data clients can send to the File Exporter for storage before the value is highlighted. The number of bytes stored is accumulated from the time the File Exporter is started. The highlighting is removed when the File Exporter is restarted, at which time the value returns to 0 (zero). Enter a threshold for this statistic in 64 kilobyte units. For example, to have **Scout** highlight this value when it equals or exceeds 128 kilobytes, specify a threshold of 2.

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.

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 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. 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 **scout**. 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 **-server**. 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 *dcshared/bin/scout*.

To change attention settings (statistic and threshold pairs), you must stop and restart **Scout**. 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. **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 **Scout**, enter the interrupt command (<Ctrl-c> or its equivalent) for your system in the **Scout** window.

Scout 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.

scout(8dfs)

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.

Examples The following **scout** command causes the program to monitor the File Exporters on File Server machines **fs1** and **fs2** in the **abc.com** cell. **Scout** probes the File Exporters every 30 seconds and prints 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. **Scout** highlights an entry in the **Fetch** column if more than one megabyte (1024 kilobytes) of data are fetched from a File Exporter, and it highlights an entry in the **Store** column if more than 512 kilobytes are stored by a File Exporter.

```
$ scout -server fs1 fs2 -b ../../abc.com/hosts -attention fetch 16 store 8
```

upclient

Purpose 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 on the local disk of **-server** to be periodically checked. 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 **-log**) and ignored.

Specify a / (slash) with this option to check all files and directories on the local disk of **-server**.

-time *frequency*

Specifies in number of seconds how often the **upclient** process is to check each file or directory specified with **-path** for changes. 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.

-help

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

upclient(8dfs)

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 **-server** to be sure they match the corresponding files and directories on the local machine (the machine running **upclient**). If a file on **-server** 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 **-server**. 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. To check all files and directories on **-server**, specify a / (slash) with the **-path** option. When specifying 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 (if a log file is named with **-file**) 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.

Multiple **upclient** processes can be run on one host machine. Therefore, a filename specified with the **-file** option needs to be distinct.

Finally, the machine running the **upclient** process must be named in the **admin.up** file on the machine running the **upserver** process. Otherwise, its **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 **/osf1_pmax/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 **upclient**.

```
$ upclient -s ././abc.com/hosts/fs1 -p /osf1_pmax/bin -t 180 -l /tmp/fs1/UpclientLog
```

Related Information

Commands: **upserver(8dfs)**

Files: **admin.up(4dfs)**

upserver

Purpose 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.

Specify a / (slash) with this option 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** 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 which of the files and directories on a machine's local disk the **upserver** can distribute. To allow all of a machine's files and directories to be distributed, specify a / (slash) with the **-path** option. When specifying multiple files and directories, the paths must be disjoint (nonintersecting). Paths are examined from left to right, and any path that intersects with a previous path is logged as an error and ignored.

If the **-path** option names an entire directory, an **upclient** process can request and receive any file from that directory. However, if the **-path** option names only a single file from a directory, an **upclient** process can request and receive only that file. If, in the latter case, an **upclient** process requests the entire directory the file resides in, the request fails; the **upclient** process receives no files. Error messages are written to the appropriate log files.

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.

When it is started, the **upserver** creates the *dcelocal/var/dfs/adm/UpLog* event log file if the file does not already exist. It then appends messages to the file. If the file exists when the **upserver** 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 append messages to.

Privilege Required

You must be logged in as **root** on the local machine.

Examples The following command specifies that files from the directories **/osf1_pmax/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 /osf1_pmax/bin /usr/mike
```

The following command specifies that files from the directories **/osf1_pmax/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.

```
$ upserver -path /osf1_pmax/bin /usr/mike/public /usr/mike
```

upserver(8dfs)

Had `/usr/mike` been specified before `/usr/mike/tmp` in the previous command, `/usr/mike/tmp` would have been logged as an error in the **UpLog** file and ignored; `/usr/mike` would have been exported as intended.

Related Information

Commands: **upclient(8dfs)**

Files: **admin.up(4dfs)**, **UpLog(4dfs)**

Chapter 6

DCE Commands

dce_config(8dce)

dce_config

Purpose Installs, configures, and starts up DCE

Synopsis **dce_config**

Description

The **dce_config** shell command invokes a menu-driven interface that installs, configures, and starts up DCE. The **dce_config** command displays a hierarchy of menus, beginning with an installation menu. The **dce_config** command invokes individual installation and configuration routines, according to the selection entered by the user.

Installation routines store the binaries required for the server installation that is selected into **\$DCELOCAL**. Binaries required for a client installation are stored on every machine. The configuration menu consists of initial cell configuration, additional server configuration, and DCE client configuration. The security server and the first CDS server constitute initial cell configuration.

Privilege Required

You must have **root** authority to run the **dce_config** command.

Environment Variables

BIND_PE_SITE=1
DCEROOT=/opt
DCELOCAL=\$DCEROOT/dcelocal
DCEINSTDIR=\$DCEROOT/dce1.0
DCESHARED=\$DCEROOT/dce
SUBSYSDIR=subsys/dce
SECURITYDIR=security
DFSDIR=dfs
HOSTNAME=hostname

Exit Values

In case of an error, this command repeats requests for correct input. The user can exit the program from any menu.

Related Information

Books: *OSF DCE Administration Guide*

Index

A

- abbreviating, CDS control program
 - commands, 2–19
- abbreviating command names, 5–48
- access control list, permissions for
 - RPC control program, 1–13
- access permissions, 2–109
- ACL, permissions for RPC control program, 1–13
- acting courier role characteristic, 3–22
- adding
 - server encryption key, 5–123
 - Tape Coordinator, 5–65
 - user, group, or server to
 - administrative file, 5–121
- admin.bak file, 5–33
 - modifying, 5–33
- admin.bos file, 5–35
 - modifying, 5–35
- admin.fl file, 5–37
 - modifying, 5–37
- admin.ft file, 5–39
 - modifying, 5–39
- admin.up file, 5–41
 - modifying, 5–41
- administration
 - functions, 2–121
 - mask, 2–123
 - program, 2–113, 2–121
 - security, 4–27
- administrative lists, 5–144, 5–154
 - admin.bak, 5–55, 5–116
 - admin.bak file, 5–33
 - admin.bos, 5–120, 5–176
 - admin.bos file, 5–35
 - admin.fl, 5–55, 5–116, 5–220, 5–228
 - admin.fl file, 5–37
 - admin.ft, 5–55, 5–116, 5–228, 5–334
 - admin.ft file, 5–39
 - admin.up file, 5–41
- advertise command, 3–3
- advertisement interval
 - characteristic, 3–22
- aggregate
 - displaying information, 5–233
 - listing, 5–273
 - repairing, 5–349
 - verifying consistency, 5–349
- aggregates
 - exported, 5–43
 - exporting, 5–216
 - increasing size, 5–339

- initializing, 5–342
- restoring, 5–92
- anodes, fileset, 5–350
- anonymous user, 2–113
- attribute groups, 3–9
- attributes
 - Authentication Failures, 2–87
 - Cache Bypasses, 2–87
 - Cache Hits, 2–87
 - cached clearinghouse, 2–74
 - CCS_LinkTimeout, 2–70
 - CDS_AllUpTo, 2–84, 2–89, 2–97
 - CDS_Cellname, 2–84
 - CDS_CHDirectories, 2–84
 - CDS_CHLastAddress, 2–85
 - CDS_CHName, 2–85
 - CDS_CHState, 2–85
 - CDS_Class, 2–95
 - CDS_ClassVersion, 2–95
 - CDS_Convergence, 2–89, 2–97
 - CDS_CTS, 2–82, 2–85, 2–90, 2–93, 2–95, 2–98
 - CDS_DirectoryVersion, 2–85, 2–90, 2–98
 - CDS_Epoch, 2–90
 - CDS_InCHName, 2–98
 - CDS_LastSkulk, 2–90, 2–98
 - CDS_LastUpdate, 2–90, 2–98
 - CDS_LinkTarget, 2–93
 - CDS_LinkTarget, 2–70
 - CDS_LinkTimeout, 2–93
 - CDS_ObjectUUID, 2–82, 2–90, 2–95, 2–98
 - CDS_ParentPointer, 2–90, 2–98
 - CDS_Replicas, 2–82, 2–91, 2–99
 - CDS_ReplicaState, 2–91, 2–99
 - CDS_ReplicaType, 2–91, 2–99
 - CDS_ReplicaVersion, 2–85, 2–91, 2–99
 - CDS_RingPointer, 2–91, 2–99
 - CDS_UTS, 2–83, 2–85, 2–91, 2–93, 2–96, 2–99
 - Child Update Failures, 2–101
 - Creation Time, 2–87, 2–101
 - Crucial Replicas, 2–101
 - Data Corruption, 2–85
 - Enable Counts, 2–85
 - Future Skew Time, 2–101
 - Known Clearinghouses, 2–101
 - LCDS_Epoch, 2–98
 - Miscellaneous Operations, 2–74, 2–87
 - Name, 2–76
 - Read Access, 2–85
 - Read Operations, 2–74, 2–87, 2–101
 - References Returned, 2–85
 - Security Failures, 2–102
 - Skulk Failures, 2–85
 - Skulks Completed, 2–102
 - Times Clearinghouse Entry Missing, 2–86
 - Times Lookup Paths Broken, 2–102
 - Times Root Not Reachable, 2–86

- Towers, 2-74, 2-76
- Upgrades Failed Counts,
2-86
- Write Accesses, 2-86
- Write Operations, 2-74,
2-88, 2-102
- Authenticated RPC, choosing
server principal name, 1-12
- Authentication Failures attribute,
2-87
- authorization checking, disabling,
5-158
- automatic tdf change
characteristic, 3-23

B

- backup courier, 3-19
- Backup Database, 5-116
 - creating backup copy,
5-104
 - listing information, 5-76
 - restoring backup copy, 5-90
 - verifying status, 5-114
- backup fileset, removing, 5-255
- Backup Server
 - generated messages, 5-4
 - initializing, 5-116
- Backup System, 5-51
 - adding Tape Coordinator,
5-65
 - Backup Database, 5-76,
5-90, 5-114
 - backup tapes, 5-70, 5-80,
5-88, 5-105
 - backups, 5-74
 - commands, 5-51
 - creating fileset family, 5-63
 - defining fileset family entry,
5-59
 - dump hierarchy, 5-70, 5-82
 - dump level, 5-56, 5-70,
5-109
 - dump set, 5-70, 5-76,
5-105
 - dumping fileset family,
5-70
 - fileset family, 5-85
 - fileset family entry, 5-100
 - initializing Tape
Coordinator, 5-178
 - listing machines running
tape drives, 5-86
 - removing dump level, 5-99
 - removing dump set, 5-68
 - removing fileset family,
5-101
 - restoring aggregates, 5-92
 - restoring date format, 5-96
 - restoring date-specific
restore, 5-96
 - restoring individual filesets,
5-95
 - Tape Coordinator, 5-102,
5-111
- Backup Tape Coordinator, 5-51
- backup tapes, 5-70
 - creating labels, 5-80
 - extracting dump set
information, 5-105
 - reading label information,
5-88
 - scanning contents, 5-105

- bak adddump command, 5–56
- bak addftentry command, 5–59
- bak addftfamily command, 5–63
- bak addhost command, 5–65
- bak apropos command, 5–67
- bak commands, 5–51
 - interactive mode, 5–52
- bak deletedump command, 5–68
- bak dump command, 5–70
- bak dumpinfo command, 5–74
- bak ftinfo command, 5–76
- bak help command, 5–78
- bak jobs command, 5–52
- bak kill command, 5–53
- bak labeltape command, 5–80
- bak lsdumps command, 5–82
- bak lsftfamilies command, 5–85
- bak lshosts command, 5–86
- bak quit command, 5–54
- bak readlabel command, 5–88
- bak restoredb command, 5–90
- bak restoredisk command, 5–92
- bak restoreft command, 5–95
- bak rmdump command, 5–99
- bak rmftentry command, 5–100
- bak rmftfamily command, 5–101
- bak rmhost command, 5–102
- bak savedb command, 5–104
- bak scantape command, 5–105
- bak setexp command, 5–109
- bak status command, 5–111
- bak verifydb command, 5–114
- BakLog file, 5–4
- bakserver command, 5–116
- bakserver process, 5–33
- Basic OverSeer Server, adding
 - server encryption key, 5–123, 5–134
- Basic Overseer Server, admin.bos file, 5–119
- Basic OverSeer Server
 - administrative file, 5–121
 - administrative list, 5–35
 - administrative lists, 5–144, 5–154
 - binary files, 5–174
- Basic Overseer Server, BosConfig file, 5–118
- Basic OverSeer Server
 - BosConfig file, 5–125, 5–128
 - disabling authorization checking, 5–158
 - displaying server encryption key, 5–146
 - examining log file, 5–138
 - files, BosConfig, 5–165
 - garbage collection, 5–132
 - generated messages, 5–8
 - initializing processes, 5–176
 - installing new binary file version, 5–142
- Basic Overseer Server
 - introduction to, 5–118
 - key files, 5–119
- Basic OverSeer Server
 - listing process status, 5–169
 - listing restart times, 5–140
 - listing timestamps, 5–136
 - removing files, 5–149
 - removing server encryption key, 5–132, 5–156
 - restarting processes, 5–151, 5–160
 - server encryption key number, 5–147
 - setting authorization, 5–158

- starting processes, 5–165, 5–167
- stopping processes, 5–163, 5–172
- Basic OverSeer Server, NoAuth, file, 5–20
- batch mode, 2–114
- binary file, installing new version, 5–142
- binary trace entries, 2–108
- bos addadmin command, 5–121
- bos addkey command, 5–123
- bos commands, introduction to, 5–118
- bos create command, 5–125
- bos delete command, 5–128
- bos exec commands, 5–130
- bos gckey command, 5–132
- bos genkey command, 5–134
- bos getdates command, 5–136
- bos getlog command, 5–138
- bos getrestart command, 5–140
- bos install command, 5–142
- bos lsadmin command, 5–144
- bos lskeys command, 5–146
- bos prune command, 5–149
- bos restart command, 5–151
- bos radmin command, 5–154
- bos rmkey command, 5–156
- bos setauth command, 5–158
- bos setrestart command, 5–160
- bos shutdown command, 5–163
- bos start command, 5–165
- bos startup command, 5–167
- bos status command, 5–169
- bos stop command, 5–172
- bos uninstall command, 5–174

- BosCofig file, 5–151
- BosConfig file, 5–5
 - creating process entries, 5–125
 - removing processes, 5–128
- BosLog file, 5–8
- bosserv command, 5–176
- bosserv process, 5–35
- buffering factor, 2–108
- butc command, 5–51, 5–178
- butc process, 5–24, 5–26, 5–28

C

- C-Stub process type, 2–110
- cache
 - DSA, 2–124
 - DUA, 2–113, 2–124
- Cache Bypasses attribute, 2–87
- Cache Hits attribute, 2–87
- Cache Manager
 - cache size, setting, 5–198
 - CacheInfo files, 5–198
 - discarding cached
 - information, 5–187, 5–188
 - displaying cache size, 5–189
 - fileset, listing, 5–196
 - honoring device files, 5–200
 - initial configuration, 5–9
 - initializing, 5–209
 - listing available device files, 5–191
 - listing database server machines, 5–195

- listing setuid permission, 5-193
 - resolving DCE pathnames, 5-206
 - setting CPU/operating system type, 5-204
 - setting setuid permission, 5-202
 - starting related daemons, 5-209
 - storing data, 5-197
 - updating cached information, 5-183
 - using debugging trace, 5-184
- Cache Manager, in DFS, listing setgid permission, 5-193
- cache size
 - displaying, 5-189
 - setting, 5-198
- Cached clearinghouse entity, 2-18
- cached clearinghouse entity, 2-74
- cached information
 - discarding, 5-187, 5-188
 - updating, 5-183
- cached server entity, 2-18, 2-76
- CacheInfo file, 5-9
- CacheItems file, 5-11
- calls
 - t_attach, 2-108
 - t_error, 2-107
- CDS commands, 2-2
- CDS control program, 2-12
 - attributes, 2-19
 - command verbs, 2-17
 - editing commands in, 2-19
 - entities, 2-18
 - invoking commands from shell, 2-17
 - starting, 2-17
 - wildcards, 2-20
- CDS control program commands
 - abbreviating commands, 2-19
 - continuing to a second line, 2-19
 - disable server, 2-44
 - editing, 2-19
 - invoking from shell, 2-17
 - list link, 2-54
 - quit, 2-19
 - reading a file of, 2-17
 - redirecting output to a file, 2-20
 - using comment character, 2-19
 - wildcards in, 2-20
- CDS permissions, 2-20
- cdsadv command, 2-9
- cdsbrowser command, 2-10
- cdsclerk command, 2-12
- CDS_AllUpTo attribute, 2-84, 2-89, 2-97
- CDS_Cellname attribute, 2-84
- CDS_CHDirectories attribute, 2-84
- CDS_CHLastAddress attribute, 2-85
- CDS_CHName attribute, 2-85
- CDS_CHState attribute, 2-85
- CDS_Class attribute, 2-95
- CDS_ClassVersion attribute, 2-95
- CDS_Convergence attribute, 2-65, 2-89, 2-97
- CDS_CTS attribute, 2-82, 2-85, 2-90, 2-93, 2-95, 2-98
- CDS_DirectoryVersion attribute, 2-85, 2-90, 2-98

- CDS_Epoch attribute, 2–90, 2–98
- CDS_InCHName attribute, 2–90, 2–98
- CDS_LastSkulk attribute, 2–90, 2–98
- CDS_LastUpdate attribute, 2–90, 2–98
- CDS_LinkTarget attribute, 2–70, 2–93
- CDS_LinkTimeout attribute, 2–70, 2–93
- CDS_ObjectUUID attribute, 2–82, 2–85, 2–90, 2–95, 2–98
- CDS_ParentPointer attribute, 2–90, 2–98
- CDS_Replicas attribute, 2–82, 2–91, 2–99
- CDS_ReplicaState attribute, 2–91, 2–99
- CDS_ReplicaType attribute, 2–91, 2–99
- CDS_ReplicaVersion attribute, 2–85, 2–91, 2–99
- CDS_RingPointer attribute, 2–91, 2–99
- CDS_UTS attribute, 2–83, 2–85, 2–91, 2–93, 2–96, 2–99
- cell, 1–9
- cell name, 1–10
- cell-relative name, 1–10
- change command, 3–4
- characteristic, DTS version, 3–23
- characteristics, 3–9
 - acting courier role, 3–22
 - advertisement, 3–22
 - automatic tdf change, 3–23
 - check interval, 3–23
 - clock adjustment rate, 3–23
 - clock resolution, 3–23
 - courier role, 3–19, 3–23
 - epoch number, 3–24
 - error tolerance, 3–19, 3–24
 - global set timeout, 3–19, 3–26
 - local servers, 3–24
 - local set timeout, 3–20, 3–24
 - maximum clock drift, 3–24
 - maximum inaccuracy, 3–20, 3–24
 - next tdf change, 3–25
 - query attempts, 3–20, 3–25
 - server entry name, 3–20, 3–25
 - server group name, 3–25
 - server principal name, 3–20, 3–25
 - servers required, 3–20, 3–25
 - synchronization, 3–25
 - synchronization hold down, 3–21
 - time differential factor, 3–26
 - time provider present, 3–26
 - time representation version, 3–26
- check interval characteristic, 3–19, 3–23
- child entity, 2–18, 2–82
- Child Update Failures attribute, 2–101
- clear cached server command, 2–24
- clear clearinghouse command, 2–25
- clearinghouse entity, 2–18, 2–84

- clerk entity, 2-18, 2-87
- client system, 2-122
- client/server system, 2-122
- clock adjustment rate
 - characteristic, 3-23
- clock resolution characteristic,
 - 3-23
- clock settings counter, 3-26
- cm checkfilesets command, 5-183
- cm commands, introduction to,
 - 5-181
- cm debug command, 5-184
- cm debug commands, 5-12
- cm flush command, 5-187
- cm flushfileset command, 5-188
- cm getcachesize command, 5-189
- cm getdevok command, 5-191
- cm getsetuid command, 5-193
- cm lscellinfo command, 5-195
- cm lsstores command, 5-196
- cm resetstores command, 5-197
- cm setcachesize command, 5-198
- cm setdevok command, 5-200
- cm setsetuid command, 5-202
- cm sysname command, 5-204
- CMXL log file, 2-108
- CMXTRACE, 2-108
- CMXTRACE environment
 - variable, 2-107
- command names, abbreviating
 - DFS, 5-48
- command suites, in DFS, 5-47
- Commands, v5srvtab, 4-7
- commands
 - add directory, 2-4
 - add element, 1-15
 - add object, 2-6
 - advertise, 3-3
 - backup, 5-51
 - bak, 5-51
 - bak adddump, 5-56
 - bak addftentry, 5-59
 - bak addftfamily, 5-63
 - bak addhost, 5-65
 - bak apropos, 5-67
 - bak deletedump, 5-68
 - bak dump, 5-70
 - bak dumpinfo, 5-74
 - bak ftinfo, 5-76
 - bak help, 5-78
 - bak jobs, 5-52
 - bak kill, 5-53
 - bak lsdumps, 5-82
 - bak lsftfamilies, 5-85
 - bak lshosts, 5-86
 - bak quit, 5-54
 - bak readlabel, 5-88
 - bak restoredb, 5-90
 - bak restoredisk, 5-92
 - bak restoreft, 5-95
 - bak rmdump, 5-99
 - bak rmftentry, 5-100
 - bak rmftfamily, 5-101
 - bak rmhost, 5-102
 - bak savedb, 5-104
 - bak scantape, 5-105
 - output, 5-106
 - bak setexp, 5-109
 - bak status, 5-111
 - bak verifydb, 5-114
 - bakserver, 5-116
 - Basic Overseer Server,
 - 5-118
 - bos, 5-118
 - bos addadmin, 5-121
 - bos addkey, 5-123
 - bos create, 5-125
 - bos delete, 5-128

- bos exec, 5–130
- bos gckey, 5–132
- bos genkey, 5–134
- bos getdates, 5–136
- bos getlog, 5–138
- bos getrestart, 5–140
- bos install, 5–142
- bos lsadmin, 5–144
- bos lskeys, 5–146
- bos prune, 5–149
- bos restart, 5–151
- bos radmin, 5–154
- bos rmkey, 5–156
- bos setauth, 5–158
- bos setrestart, 5–160
- bos shutdown, 5–163
- bos start, 5–165
- bos startup, 5–167
- bos status, 5–169
- bos stop, 5–172
- bos uninstall, 5–174
- bosserv, 5–176
- butc, 5–51, 5–178
- Cache Manager, 5–181
- cdsadv, 2–9
- cdsbrowser, 2–10
- cdsclerk, 2–12
- cdscp, 2–12
- clear cached server, 2–24
- clear clearinghouse, 2–25
- cm, 5–181
- cm checkfilesets, 5–183
- cm debug, 5–12, 5–184
- cm flush, 5–187
- cm flushfileset, 5–188
- cm getcachesize, 5–189
- cm getdevok, 5–191
- cm getsetuid, 5–193
- cm lscellinfo, 5–195
- cm lsstores, 5–196
- cm resetstores, 5–197
- cm setcachesize, 5–198
- cm setdevok, 5–200
- cm setsetuid, 5–202
- cm sysname, 5–204
- create, 3–5
- create child, 2–26
- create clearinghouse, 2–27
- create directory, 2–29
- create link, 2–30
- create object, 2–32
- create replica, 2–33
- dce_config, 6–2
- dce_login, 2–9, 2–11
- define cached server, 2–34
- delete, 3–6
- delete child, 2–36
- delete clearinghouse*O,
2–37
- delete directory, 2–39
- delete link, 2–40
- delete object, 2–41
- delete replica, 2–42
- dfsbind, 5–206
- dfsd, 5–209
- dfsexport, 5–216
- disable, 3–7
- disable clerk, 2–43
- do filename, 2–17
- DTS, 3–2
- dtscp, 3–8
- dtscp help, 3–16
- dtstd, 3–11
- dump clerk cache, 2–45
- enable, 3–13
- exit, 3–15
- fileset, 5–225
- flserver, 5–220

fms, 5-14, 5-222
fts, 5-225
fts addsite, 5-230
fts aggrinfo, 5-233
fts clone, 5-236
fts clonesys, 5-238
fts create, 5-241
fts crfldbentry, 5-244
fts crmount, 5-247
fts crserverentry, 5-252
fts delfldbentry, 5-258
fts delmount, 5-261
fts delserverentry, 5-262
fts dump, 5-264
fts edserverentry, 5-267
fts lock, 5-271
fts lsaggr, 5-273
fts lsfdb, 5-275
fts lsft, 5-279
fts lsheader, 5-283
fts lsmount, 5-287
fts lsreplicas, 5-289
fts lserverentry, 5-291
fts move, 5-293
fts release, 5-296
fts rename, 5-298
fts restore, 5-300
fts rmsite, 5-305
fts setquota, 5-307
fts setrepinfo, 5-309
fts statftserver, 5-316
fts statrepserver, 5-318
fts syncfdb, 5-320
fts syncserv, 5-323
fts unlock, 5-326
fts unlockfdb, 5-328
fts update, 5-330
fts zap, 5-332
ftserver, 5-334
fxd, 5-336
gdad, 2-46
gdscacheadm, 2-105
gdscmxl, 2-107
gdsdirinfo, 2-110
gdsditadm, 2-113
gdsstep, 2-119
gdssysadm, 2-121
growaggr, 5-339
list child, 2-48
list clearinghouse, 2-50
list directory, 2-52
list object, 2-56
newaggr, 5-342
passwd_export, 4-9
passwd_import, 4-11
passwd_override, 4-3
quit, 2-17, 3-17
remove directory, 2-58
remove link, 2-60
remove object, 2-61
repserver, 5-347
rgy_edit, 4-15
rpccp, 1-4
rpccp add element, 1-15
rpccp add entry, 1-18
rpccp add member, 1-20
rpccp export, 1-22
rpccp help, 1-26
rpccp import, 1-28
rpccp remove element, 1-31
rpccp remove entry, 1-34
rpccp remove group, 1-35
rpccp remove mapping,
1-36
rpccp remove member, 1-40
rpccp remove profile, 1-42
rpccp show entry, 1-43
rpccp show group, 1-45

- rpccp show mapping, 1–47
- rpccp show profile, 1–50
- rpccp show server, 1–54
- rpccp unexport, 1–56
- rpcd, 1–59
- salvage, 5–23, 5–349
- scout, 5–355
- secd, 4–32
- security, 4–8
- sec_admin, 4–27
- sec_clientd, 4–29
- sec_create_db, 4–30
- set, 3–18
- set cdsdp confidence, 2–63
- set cdsdp preferred
 - clearinghouse, 2–64
- set directory, 2–65
- set directory to new epoch,
 - 2–67
- set directory to skulk, 2–69
- set link, 2–70
- set object, 2–72
- show, 3–22
- show cached clearinghouse,
 - 2–74
- show cached server, 2–76
- show cdsdp confidence,
 - 2–78
- show cdsdp preferred
 - clearinghouse, 2–79
- show cell, 2–80
- show child, 2–82
- show clearinghouse, 2–84
- show clerk, 2–87
- show directory, 2–89
- show link, 2–93
- show object, 2–95
- show replica, 2–97
- show server, 2–101
- synchronize, 3–30
- unadvertise, 3–31
- upclient, 5–359
- update, 3–32
- upserver, 5–362
- commands, bak labeltape, 5–80
- commnads, change, 3–4
- configuration
 - data, 2–121
 - mode, 2–121
 - type, 2–122
- connection redirection, 2–107
- connection setup and termination,
 - 2–107
- continuing CDS commands to a second line, 2–19
- control program, 3–8
 - RPC, 1–4
- counters, 3–9
 - for show command, 3–26
- courier, 3–19
- courier role characteristic, 3–19,
 - 3–23
- CPU/operating system type,
 - setting, 5–204
- create child command, 2–26
- create clearinghouse command,
 - 2–27
- create command, 3–5
- create directory command, 2–29
- create link commands, 2–30
- create object command, 2–32
- create replica command, 2–33
- creating
 - backup copy of Backup Database, 5–104
 - backup read/write fileset,
 - 5–236, 5–238
 - BosConfig file, 5–5

- entries for non-LFS filesets, 5-244
- fileset, read/write, 5-241
- fileset family, 5-63
- fileset mount point, 5-247
- labels for backup tapes, 5-80
- password files, 4-9
- process entries, 5-125
- registry database, 4-30
- registry database entries, 4-11
- server entry in FLDB, 5-252

Creation Time attribute, 2-74, 2-87, 2-101

creation time counter, 3-27

cron process, 5-125

Crucial Replicas attribute, 2-101

current time status, 3-28

D

daemon

- security, 4-32
- Security Client, 4-29

daemons

- background, 5-212
- callback, 5-212
- maintenance, 5-212
- starting, 5-209
- starting kernel, 5-336

Data Corruptions attribute, 2-85

data exchange, 2-107

database server machines, listing, 5-195

date format in Backup System, 5-96

dce_config command, 6-2

dce_login command, 2-9, 2-11

debugging trace, 5-184

define cached server command, 2-34

delete child command, 2-36

delete clearinghouse command, 2-37

delete command, 3-6

delete directory command, 2-39

delete link command, 2-40

delete object command, 2-41

delete replica command, 2-42

deleting, server entry from FLDB, 5-262

delta update, 2-122

device files

- honoring, 5-200
- listing availability, 5-191

DFS commands, 5-47

- fts delete, 5-255

DFS help commands, 5-49

dfsatab file, 5-43

dfsbind command, 5-206

dfsd command, 5-209

dfsexport command, 5-216

DFSLog file, 5-12

dfstab file, 5-45

dialog mode, 2-105, 2-114

different epochs detected counter, 3-27

Dir-User process type, 2-110

directory

- database, 2-113
- system, 2-121

- directory administration input file, 2–105
 - directory administration output file, 2–106
 - directory entity, 2–18, 2–89
 - directory pathname, 1–10
 - Directory System Agent
 - See also* DSA
 - DIRID process type, 2–110
 - disable clerk command, 2–43
 - disable command, 3–7
 - disable directives completed counter, 3–27
 - disk cache, data, 5–31
 - Distinguished Name, 2–113
 - DN
 - See also* Distinguished Name
 - DSA, 2–113
 - See also* Directory System Agent
 - cache, 2–124
 - default, 2–113
 - DTS, commands, 3–2, 3–8
 - DTS control program. *See* dtscp
 - DTS daemon, restarting, 3–11
 - DTS version characteristic, 3–23
 - dtscp
 - attribute groups, 3–9
 - attribute specifiers for show command, 3–22
 - attributes, 3–9
 - characteristics for show command, 3–22
 - counters for show command, 3–26
 - status for show command, 3–28
 - dtscp commands, 3–8
 - dtscp help command, 3–16
 - dtscd command, 3–11
 - DUA, cache, 2–113, 2–124
 - DUA-Cache process type, 2–110
 - dump clerk cache command, 2–45
 - dump hierarchy, 5–70, 5–82
 - dump level, 5–70
 - absolute expiration date, 5–56
 - defining, 5–56
 - expiration date, 5–109
 - names, 5–57
 - relative expiration date, 5–56
 - removing, 5–99
 - dump set, 5–70
 - clone date, 5–76
 - creation date, 5–76
 - dump ID, 5–76
 - extracting information, 5–105
 - level, 5–76
 - parent ID, 5–76
 - removing, 5–68
 - tape name, 5–76
 - dumping
 - fileset, 5–264
 - fileset family, 5–70
- ## E
- editing
 - CDS control program commands, 2–19
 - detailed, 2–107

- limited, 2–107
- registry database, 4–15
- server entry in FLDB, 5–267
- enable command, 3–13
- Enable Counts attribute, 2–85
- enable directives completed
 - counter, 3–27
- endpoint, 1–2
- endpoint map, 1–2, 1–59
- Endpoint Map Service, 1–2, 1–59
- environment variables
 - CMXTRACE, 2–107
 - NLSPATH, 1–8
 - RPC user defined, 1–7
 - RPC_DEFAULT_ENTRY,
 - 1–9
 - RPC_DEFAULT_ENTRY_SYNTAX,
 - 1–8
- EOF mark, determining size,
 - 5–222
- epoch changes completed counter,
 - 3–27
- epoch number characteristic, 3–24
- errno, 2–109
- error messages, 2–123
- error tolerance characteristic,
 - 3–19, 3–24
- event handling, 2–107
- examples, in DFS, TL file, 5–26
- exporting
 - aggregates, 5–216
 - partitions, 5–216

F

- faulty servers detected counter,
 - 3–27
- File Exporter, 5–16
 - initializing, 5–336
 - monitoring, 5–355
- FileLog file, 5–16
- files
 - admin.bak, 5–33, 5–55,
 - 5–116
 - admin.bos, 5–35, 5–119,
 - 5–176
 - admin.fl, 5–37, 5–55, 5–116,
 - 5–220, 5–228
 - admin.ft, 5–39, 5–55,
 - 5–116, 5–228, 5–334
 - admin.up, 5–41
 - administrative, 5–121
 - BakLog, 5–4
 - binary, 5–174
 - BosConfig, 5–5, 5–118,
 - 5–128, 5–151, 5–163,
 - 5–165, 5–167, 5–172,
 - 5–176
 - BosLog, 5–8
 - CacheInfo, 5–9, 5–31,
 - 5–198, 5–209
 - CacheItems, 5–11, 5–31
 - CMXL log, 2–108
 - converting to fileset format,
 - 5–300
 - creating password, 4–9
 - DFS, 5–2
 - dfsatab, 5–43
 - DFSLog, 5–12, 5–184
 - dfstab, 5–45
 - end of file mark, 5–222
 - EOF mark, 5–222

- FileLog, 5–16
 - FilesetItems, 5–17
 - FILog, 5–18
 - FMSLog, 5–14, 5–222
 - FtLog, 5–19
 - honoring device, 5–200
 - installing new binary
 - version, 5–142
 - key file, 5–119, 5–123
 - listing timestamps, 5–136
 - log, 5–138
 - NoAuth, 5–20
 - password, 4–11
 - removing, 5–149
 - RepLog, 5–22
 - SalvageLog, 5–23
 - stdio.h, 2–108
 - TapeConfig, 5–28, 5–178
 - TE, 5–24, 5–179
 - TL, 5–179
 - UpLog, 5–30
 - V file, 5–11, 5–209
 - V files, 5–31
 - v5srvtab, 4–7
 - Vn files, 5–31
- fileset
- aggregate, listing, 5–273
 - anodes, 5–350
 - commands, 5–225
 - creating backup read/write,
 - 5–238
 - creating mount point, 5–247
 - creating non-LFS entries,
 - 5–244
 - displaying entry
 - information, 5–275
 - displaying FLDB entry
 - information, 5–279
 - displaying header
 - information, 5–279,
 - 5–283
 - displaying replicas status,
 - 5–289
 - dumping, 5–264
 - dumping, time format,
 - 5–264
 - initializing Fileset Location Server, 5–220
 - initializing Fileset Server,
 - 5–334
 - listing, 5–196
 - listing FLDB server entry,
 - 5–291
 - locking entry in FLDB,
 - 5–271
 - mount point, 5–287
 - moving, 5–293
 - placing read-only at local sites, 5–296
 - producing replica, 5–330
 - read-only, 5–230
 - read/write
 - creating, 5–241
 - creating backup
 - version, 5–236
 - removing, 5–332
 - removing backup, 5–255
 - removing FLDB entry,
 - 5–305
 - removing mount point,
 - 5–261
 - removing read/write, 5–255
 - removing varying numbers,
 - 5–259
 - renaming, 5–298
 - restoring, 5–95
 - setting maximum quota,
 - 5–307
 - setting replication
 - parameters, 5–309

- synchronizing header with FLDB, 5-320
 - synchronizing with FLDB, 5-323
- fileset family
 - creating, 5-63
 - dumping, 5-70
 - listing, 5-85
 - removing, 5-101
- fileset family entry
 - defining, 5-59
 - removing, 5-100
- Fileset Location Database, 5-226
 - creating server entry, 5-252
 - deleting server entry, 5-262
 - displaying entry
 - information, 5-279
 - displaying fileset entry
 - information, 5-275
 - listing server entry, 5-291
 - locking fileset entry, 5-271
 - read-only fileset, 5-230
 - removing entry, 5-258
 - removing read-only site
 - definition, 5-305
 - server entry, editing, 5-267
 - synchronizing with fileset header, 5-320
 - synchronizing with filesets, 5-323
 - unlocking entry, 5-326, 5-328
- Fileset Location Server, 5-18, 5-226
 - administrative list, 5-37
 - initializing, 5-220
- Fileset Server, 5-19, 5-226
 - administrative list, 5-39
 - displaying activity, 5-316
 - initializing, 5-334
- FilesetItems file, 5-17
- FL Server, 5-226
- FLDB, 5-226
 - read-only fileset, adding site definition, 5-230
- FILog file, 5-18
- flow control, 2-107
- flserver command, 5-220
- fms command, 5-14, 5-222
- FMSLog file, 5-14
- FtLog file, 5-19
- fts, introduction to, 5-225
- fts addsite command, 5-230
- fts agrinfo command, 5-233
- fts clone command, 5-236
- fts clonesys command, 5-238
- fts create command, 5-241
- fts crfldbentry command, 5-244
- fts crmount command, 5-247
- fts crserverentry command, 5-252
- fts delete command, 5-255
- fts delfldbentry command, 5-258
- fts delmount command, 5-261
- fts delserverentry command, 5-262
- fts dump command, 5-264
- fts edserverentry command, 5-267
- fts lock command, 5-271
- fts lsaggr command, 5-273
- fts lsfdb command, 5-275
- fts lsft command, 5-279
- fts lsheader command, 5-283
- fts lsmount command, 5-287
- fts lsserverentry command, 5-291
- fts move command, 5-293
- fts release command, 5-296
- fts rename command, 5-298

fts restore command, 5–300
 fts rmsite command, 5–305
 fts setquota command, 5–307
 fts setrepinfo command, 5–309
 fts statftserver command, 5–316
 fts statrepserver command, 5–318
 fts syncfdb command, 5–320
 fts syncserv command, 5–323
 fts unlock command, 5–326
 fts unlockfdb command, 5–328
 fts update command, 5–330
 fts zap command, 5–332
 ftserver command, 5–334
 ftserver process, 5–37, 5–39
 Future Skew Time attribute, 2–101
 fxd command, 5–336

G

garbage collection, 5–132
 gdad command, 2–46
 GDS commands, 2–104
 gdscacheadm command, 2–105
 gdscmxl command, 2–107
 gdsdirinfo command, 2–110
 gdsditadm command, 2–113
 gdsstep command, 2–119
 gdssysadm command, 2–121
 global servers, 3–9
 global servers status, 3–28
 global set timeout characteristic,
 3–19, 3–26
 group, RPC, 1–9
 growaggr command, 5–339

H

help screens, bak apropos, 5–67
 help topics, 3–16
 help, in DFS, 5–49
 honoring, device files, 5–200

I

input, fields, 2–123
 insufficient resources detected
 counter, 3–27
 interactive mode, bak commands,
 5–52
 interface identifier, RPC definition
 of, 1–16
 invalid messages detected counter,
 3–27
 IPCID process type, 2–110

J

jobs, in bak command suite, 5–52

K

key, master, 4–27
 kill, in bak command suite, 5–53

Known Clearinghouses attribute,
2-101
krb5/v5srvtab file, 4-7

L

last synchronization status, 3-28
lb_admin program, 1-60
leaf name, 1-11
link entity, 2-18, 2-93
list child command, 2-48
list clearinghouse command, 2-50
list directory command, 2-52
list object command, 2-56
listing

- aggregates, 5-273
- Backup Database
 - information, 5-76
- backup information, 5-74
- database server machines,
5-195
- device file availability,
5-191
- file timestamps, 5-136
- fileset families, 5-85
- filesets, 5-196
- from administrative list,
5-144
- machines running tape
drives, 5-86
- process status, 5-169
- server entry in FLDB, 5-291
- server process restart times,
5-140
- Tape Coordinator
 - operations, 5-111

LLB database, 1-60
llbd, with RPC daemon, 1-60
local faults detected counter, 3-27
local servers, 3-9
local servers characteristic, 3-24
local servers status, 3-29
local set timeout characteristic,
3-20, 3-24
log, transport system interface
calls, 2-108
log file, examining for server
process, 5-138
Logon Menu Mask, 2-106, 2-114
lsreplicas command, 5-289

M

mask, administration, 2-123
master registry, 4-27
maximum clock drift rate
characteristic, 3-24
maximum inaccuracy
characteristic, 3-20, 3-24
Menu Mask

- Logon, 2-106, 2-114

Miscellaneous Operations

- attribute, 2-74, 2-87

mode, configuration, 2-121
Monitor process type, 2-110
monitoring, File Exporters, 5-355
mount point

- creating, 5-247
- removing, 5-261

N

- Name attribute, 2–76
- name service entries
 - See also* RPC NSI attributes
 - structure of entry names, 1–9
- names
 - guidelines for choosing server principal names, 1–12
 - guidelines for RPC name service entries, 1–11
- newaggr command, 5–342
- next tdf change characteristic, 3–25
- NLSPATH, RPC control program requirements, 1–8
- no global servers detected counter, 3–27
- NoAuth file, 5–20
- non-LFS partitions, exported, 5–43
- noncourier, 3–19
- NotRun status flag, 5–5
- NSI, 1–2
 - ACL permissions for control program, 1–13
 - attributes, 1–9
 - group, 1–9
 - name service names, 1–11
 - profile, 1–9
 - server entry, 1–9
- NSI commands, 1–9

O

- object entity, 2–18, 2–95

P

- parent directory, 1–10
- partitions, exporting, 5–216
- passwd_export command, 4–9
- passwd_import command, 4–11
- passwd_override command, 4–3
- password files, creating, 4–9
- passwords, encrypting, 4–27
- pathnames, resolving DCE, 5–206
- permissions, 2–20
 - listing setuid, 5–193
 - setgid, 5–193
 - setting setgid, setting, 5–202
 - setting setuid, 5–202
 - setuid, 5–193
- PID process type, 2–110
- process types, GDS, 2–110
- processes, 5–125
 - bakserver, 5–5, 5–33
 - bossserver, 5–35
 - butc, 5–24, 5–28
 - command parameters, 5–5
 - cron, 5–5
 - DFS authorization checking, 5–20
 - flserver, 5–5
 - ftserver, 5–5, 5–37, 5–39
 - initializing BOS, 5–176
 - initializing Replication Server, 5–347
 - initializing Tape Coordinator, 5–178

- kernel, 5-336
- listing server restart times, 5-140
- listing status, 5-169
- NoAuth file, 5-20
- removing, 5-128
- Replication Server, 5-318
- repserver, 5-5
- restarting, 5-151, 5-160
- restarting date and time, 5-160
- setting authorization, 5-158
- simple, 5-5
- starting, 5-165, 5-167
- stopping, 5-163, 5-172
 - bakserver, 5-90
- upclient, 5-5, 5-30, 5-41
- upserver, 5-5, 5-30, 5-41
- processes, in DFS, butc, 5-26
- profile, RPC, 1-9
- protocol mismatches detected counter, 3-27

Q

- query attempts characteristic, 3-20, 3-25
- quit command, 3-17
- quit, in bak command suite, 5-54

R

- Read Accesses attribute, 2-85
- Read Operations attribute, 2-74, 2-87, 2-101
- read-only fileset, adding site definition to FLDB, 5-230
- read/write fileset
 - creating, 5-241
 - creating backup version, 5-236, 5-238
 - removing, 5-255
- reading file of CDS commands, 2-17
- redirecting, CDS command output to a file, 2-20
- References Returned attribute, 2-85
- registry database
 - creating, 4-30
 - creating entries, 4-11
 - editing, 4-15
- remove directory command, 2-58
- remove lind command, 2-60
- remove object command, 2-61
- remove profile command, 1-42
- removing
 - dump level, 5-99
 - dump set, 5-68
 - files, 5-149
 - fileset, 5-332
 - fileset family, 5-101
 - fileset family entry, 5-100
 - fileset mount point, 5-261
 - filesets, varying numbers, 5-259
 - FLDB entry, 5-258
 - from administrative list, 5-154

- processes from BosConfig file, 5–128
 - read-only site definition
 - from FLDB entry, 5–305
 - read/write fileset, 5–255
 - server encryption key, 5–132, 5–156
 - Tape Coordinator entry, 5–102
- replica entity, 2–18, 2–97
- Replication Server
 - displaying process information, 5–318
 - initializing processes, 5–347
 - messages, 5–22
- RepLog file, 5–22
- repsrver command, 5–347
- restarting, DTS daemon, 3–11
- restoring
 - aggregates, 5–92
 - Backup Database, 5–90
 - date format, 5–96
 - individual filesets, 5–95
- rgy_edit command, 4–15
- RPC
 - ACL permissions for control program, 1–13
 - cell, 1–9
 - cell name, 1–10
 - cell-relative name, 1–10
 - daemon, 1–2
 - directory pathname, 1–10
 - endpoint, 1–2
 - endpoint map, 1–2
 - Endpoint map Service, 1–2
 - group, 1–9
 - leaf name, 1–11
 - name service entries, 1–9
 - NLSPATH environment variable, 1–8
 - NSI, 1–2
 - NSI attributes, 1–9
 - profile, 1–9
 - RPC_DEFAULT_ENTRY environment variable, 1–9
 - RPC_DEFAULT_ENTRY_SYNTAX environment variable, 1–8
 - server entry, 1–9
 - server instance, 1–2
 - server principal name, 1–12
 - user-defined environment variables, 1–7
- RPC commands, 1–2
- RPC control program, 1–4
- RPC daemon, 1–59
- RPCCP. *See* RPC control program
- rpccp, add entry, 1–18
- rpccp add member command, 1–20
- rpccp commands, 1–4
 - quotation marks with, 1–6
 - scope of, 1–7
 - wildcards with, 1–6
- rpccp export command, 1–22
- rpccp help command, 1–26
- rpccp import command, 1–28
- rpccp remove element command, 1–31
- rpccp remove entry command, 1–34
- rpccp remove group command, 1–35
- rpccp remove mapping command, 1–36

- rpccp remove member command, 1–40
- rpccp show entry command, 1–43
- rpccp show group command, 1–45
- rpccp show mapping command, 1–47
- rpccp show profile command, 1–50
- rpccp show server command, 1–54
- rpccp unexport command, 1–56
- rpcd command, 1–2, 1–59
- RPC_DEFAULT_ENTRY
 - environment variable, 1–9
- RPC_DEFAULT_ENTRY_SYNTAX environment variable, 1–8
- Run status flag, 5–5

S

- S-Stub process type, 2–110
- salvage command, 5–23, 5–349
- SalvageLog file, 5–23
- Salvager
 - repairing aggregate, 5–349
 - verifying aggregate consistency, 5–349
- Scout
 - display environment, 5–357
 - monitoring screen, 5–357
- scout command, 5–355
- secd command, 4–32
- security commands, 4–8
- security data medium, 2–123
- Security Failures attribute, 2–102
- Security Server, 4–27, 4–29, 4–32
- Security Service, name service names, 1–12
- sec_admin command, 4–27
- sec_clientd command, 4–29
- sec_create_db command, 4–30
- server encryption key
 - adding, 5–123, 5–134
 - displaying, 5–146
 - removing, 5–132, 5–156
 - version number, 5–147
- server entity, 2–19, 2–101
- server entry
 - creating in FLDB, 5–252
 - deleting from FLDB, 5–262
 - editing, 5–267
 - RPC, 1–9
- server entry name characteristic, 3–20, 3–25
- server group name characteristic, 3–25
- server instance, RPC, 1–2
- server machine, restarting processes, 5–151
- server principal name characteristic, 3–20, 3–25
- servers not responding counter, 3–28
- servers required characteristic, 3–20, 3–25
- set cdsdp confidence command, 2–63
- set cdsdp preferred clearinghouse command, 2–64
- set command, 3–18
- set directory command, 2–65
- set directory to new epoch command, 2–67

- set directory to skulk command, 2-69
 - set link command, 2-70
 - set object command, 2-72
 - setgid permission
 - listing, 5-193
 - setting, 5-202
 - setting
 - cache size, 5-198
 - CPU/operating system type, 5-204
 - dump level expiration date, 5-109
 - setgid permission, 5-202
 - setuid permission, 5-202
 - setuid permission
 - listing, 5-193
 - setting, 5-202
 - show cached clearinghouse command, 2-74
 - show cached server command, 2-76
 - show cdscp confidence command, 2-78
 - show cdscp preferred clearinghouse command, 2-79
 - show cell command, 2-80
 - show child command, 2-82
 - show clearinghouse command, 2-84
 - show clerk command, 2-87
 - show command, 3-22
 - show directory command, 2-89
 - show link command, 2-93
 - show object command, 2-95
 - show replica commands, 2-97
 - show server commands, 2-101
 - simple process, 5-125
 - Skulk Failures attribute, 2-85
 - Skulks Completed attribute, 2-102
 - Skulks Initiated attribute, 2-102
 - starting
 - daemons, 5-209
 - kernel processes, 5-336
 - processes, 5-165, 5-167
 - state status, 3-29
 - status, 3-9
 - status flags
 - NotRun, 5-5
 - Run, 5-5
 - stderr, 2-109
 - stdio.h header file, 2-108
 - stopping
 - bakserver process, 5-90
 - processes, 5-163, 5-172
 - synchronization hold down
 - characteristic, 3-21, 3-25
 - synchronizations completed
 - counter, 3-28
 - synchronize command, 3-30
 - system
 - client, 2-122
 - client/server, 2-122
 - system errors detected counter, 3-28
- ## T
- Tape Coordinator
 - adding, 5-65
 - initializing, 5-178

- listing operations, 5-111
- removing entry, 5-102
- Tape Coordinator ID, 5-51
- Tape Coordinator machine, 5-51
 - configuration parameters, 5-28
 - error file, 5-24
- Tape Coordinator machine, in DFS,
 - tape drives, log file, 5-26
- TapeConfig file, 5-28, 5-178
 - parameters, 5-28
- TCID, 5-51
- TE file, 5-24
- time differential factor
 - characteristic, 3-26
- time provider failures detected
 - counter, 3-28
- time provider present
 - characteristic, 3-26
- time representation version
 - characteristic, 3-26
- time representation version mismatches counter, 3-28
- Times Clearinghouse Entry
 - Missing attribute, 2-86
- Times Lookup Paths Broken
 - attribute, 2-102
- Times Root Not Reachable
 - attribute, 2-86
- TL file, 5-26
- TL file, in DFS, example, 5-26
- too few servers detected counter, 3-28
- too many servers detected counter, 3-28
- Towers attribute, 2-74, 2-76
- trace evaluation, 2-119

- trace information, 2-107
- trace mechanism, 2-107
- transport, system interface, 2-107

U

- uid status, 3-29
- unadvertise command, 3-31
- upclient command, 5-359
- upclient process, 5-30, 5-41
- update command, 3-32
- Update Server
 - administrative list, 5-41
 - server portion, 5-30
- updates initiated counter, 3-28
- Upgrades Failed Counts attribute, 2-86
- UpLog file, 5-30
- upserver command, 5-362
- upserver process, 5-30, 5-41
- user, anonymous, 2-113
- using DFS commands, 5-48

V

- V file, 5-11
- V files, 5-31
- Vn files, 5-31

W

- wildcard characters, in CDS
 - control program commands,
2–20
- wildcards, 5–60
- Write Accesses attribute, 2–86
- Write Operations attribute, 2–74,
2–88, 2–102

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

Notes

OPEN SOFTWARE FOUNDATION™

INFORMATION REQUEST FORM

Please send to me the following:

- OSF™ Membership Information
- OSF™ DCE License Materials
- OSF™ DCE Training Information

Contact Name _____

Company Name _____

Street Address _____

Mail Stop _____

City _____ State _____ Zip _____

Phone _____ FAX _____

Electronic Mail _____

MAIL TO:

Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142

Attn: OSF™ DCE

For more information about OSF™ DCE call OSF Direct Channels at 617 621 7300.

OSF™ DCE

OSF™ DCE Administration Reference

TITLES IN THE OSF™ DCE SERIES:

Introduction to OSF™ DCE

OSF™ DCE User's Guide and Reference

OSF™ DCE Administration Reference

OSF™ DCE Application Development Guide

OSF™ DCE Application Development Reference

Application Environment Specification (AES)
— Distributed Computing

Printed in the U.S.A.

Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142

Prentice Hall, Inc.

ISBN 0-13-643818-0



9 780136 438182

DISTRIBUTED SYSTEMS