

OS/390®



Distributed File Service DFS Configuring and Getting Started

OS/390®



Distributed File Service DFS Configuring and Getting Started

Note!

Before using this information and the product it supports, be sure to read the general information under Appendix E, "Notices" on page 209.

Seventh Edition (September 2000)

This edition, SC28-1722-06, applies to Version 2 Release 10 of OS/390 Distributed File Service (program number 5647-A01) and to all subsequent releases and modifications until otherwise indicated in new editions. This edition replaces SC28-1722-05.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for reader's comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
Information Development, Dept. G60
1701 North Street
Endicott, NY, 13760-5553
United States of America

FAX (United States & Canada): 1+607+752-2327
FAX (Other Countries):
Your International Access Code +1+607+752-2327

IBMLink (United States customers only): GDLVME(PUBRCF)
Internet e-mail: pubrcf@vnet.ibm.com
World Wide Web: <http://www.ibm.com/s390/os390/>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1996, 2000. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

The following statements are provided by the Open Software Foundation.

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.

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 The Open Group in the United States and other countries.

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

DECstation 3100 and DECnet are trademarks 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.

Ethernet is a registered trademark of Xerox Corporation.

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

MX300i 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

About This Book	xv
Who Should Use This Book	xv
How to Use This Book	xv
Where to Find More Information	xvi
Softcopy Publications	xvi
Internet Sources	xvi
Unsupported Services and Commands	xvi
Conventions Used in This Book	xvii
 Summary of Changes	 xix
 Chapter 1. Introduction to OS/390 DFS	 1
OS/390 DFS Features	1
DFS Client Support	2
DFS Server Support	2
DCE Local File System Support	2
Using the OS/390 Hierarchical File System	3
Using the OS/390 Data Sets	3
Using OS/390 DFS Commands in TSO	3
Using the OS/390 DFS Client (DFSCM)	4
 Chapter 2. Global and Cell Considerations	 5
Planning for Access Control	5
Cell Filespace	5
DFS Administrative Domains	6
DFS Administrative Lists	6
Determining the Roles of DFS Machines	6
Setting Up Filesets	6
 Chapter 3. DFS Client (DFSCM) Considerations	 9
DFSCM Relationship to DCE and DFS Servers	9
DFSCM Address Space and PROCLIB	9
 Chapter 4. DFS Server Considerations	 13
DFS Address Space, Processes, and PROCLIBs	13
DFS Administration Utilities	16
DFS Server Relationship to DCE	16
 Chapter 5. Migration Overview	 17
Steps for Migrating	18
Distributed File Service Migration Actions	19
DCE Cell and Host System Considerations	19
DCE Cell Configuration Considerations	19
Test DCE Cell Considerations	19
DFS Customizable and Configuration Data Set Considerations	20
DFSKERN in a Separate Address Space	20
RACF or Equivalent Security System Entries	20
Year 2000 Support for OS/390 Distributed File Service	20
Migration to New DFS Release	20
Migration Roadmap	22

	OS/390 Version 2 Release 10 Summary	22
	OS/390 Version 2 Release 9 Summary	22
	OS/390 Version 2 Release 8 Summary	22
	OS/390 Version 2 Release 7 Summary	22
	OS/390 Version 2 Release 6 Summary	23
	OS/390 Version 2 Release 5 Summary	23
	OS/390 Version 1 Release 3 Summary	23
	OS/390 Version 1 Release 2 Summary	24
	Version 2 Release 10 Overview	25
	NT LM 0.12 (includes NT LM 0.12 encrypted password)	25
	RFS SMB (SMB support for MVS data sets)	25
	Greater than 32 bit Episode files	25
	Version 2 Release 9 Overview	26
	Encrypted Password Login	26
	Authorized Programs	26
	Version 2 Release 8 Overview	27
	SMB File/Print Server Support	27
	Authorized Programs	27
	Version 2 Release 7 Overview	28
	DFS Server Multi-home Support	28
	Protected RPC	28
	OSF/Transarc compatibility	28
	Version 2 Release 6 Overview	30
	DFSKERN in a Separate Address Space	30
	DFS Backup	30
	File basis translation	30
	Version 2 Release 5 Overview	31
	AIX compatibility	31
	DFS RAS/NLS message support	31
	OSF DCE 1.2.1 enhancements	31
	RFS enhancements	32
	Version 2 Release 4 Overview	33
	Version 1 Release 3 Overview	33
	Record Data	33
	Version 1 Release 2 Overview	34
	DFS client	34
	File Location Server (flserver)	34
	Backup Server (bakserver)	34
	Summary of Interface Changes	36
	Distributed File Service Files	36
	Distributed File Service Commands	36
	Distributed File Service APIs	37
	SMB Files	37
	SMB Commands	37
	Environment Variables	37
	 Chapter 6. Post Installation Considerations	 41
	DFS Installation and Configuration Overview	41
	Using dfs_cpfiles to Create Configuration Files (dfscpfil)	41
	DFS Configuration Considerations	43
	DFS Server Considerations	43
	DFS Client (DFSCM) Considerations	45
	 Chapter 7. Configuring DFS	 47

Overview of DFS Configuration	47
DFS Configuration Steps	48
Using the DFS Configuration Program (DFSCONF)	53
DCE Messaging Subsystem Environment Variables for DFSCONF	53
Configuration Log File for DFSCONF	54
DFSCONF Environment Variables	54
Set Up the DFSCONF Administrator TSO User ID	55
DCE Login Panel	55
DFS Configuration Main Menu	56
Using DFSCONF to Configure DFS Servers	57
Using DFSCONF to Configure the DFS File Server	58
Using DFSCONF to Configure the DFS Backup Server	58
Using DFSCONF to Configure the DFS Fileset Location Database Server	59
Using DFSCONF to Configure the DFS Client	59
Completing the DFS File Server Configuration	61
Completing the boserver Configuration	61
Completing the FLDB Server (flserver) Configuration	61
Creating the DFS File Server FLDB Entry	63
Completing Fileset Server (ftserver) Configuration	63
Completing Replication (repserver) Server Configuration	65
Completing Update Client (upclient) Configuration	65
Completing Update Server (upserver) Configuration	66
Completing the Backup Database Server (bakserver) Configuration	68
Completing the DFS Backup Tape Coordinator Configuration	69
Re-Establish the bakserver Object UUID	71
Completing the DFS Client (DFSCM) Configuration	72
System Parmlib Member BPXPRMxx Entry for the DFSCM	72
DFSCM Cache Configurations	73
DFSCM User .profile File Considerations	77
Enabling DFSCM to Use DCE Single Sign-on	77
DFSCM Environment Variable Considerations	77
Customizing the ioepdcf File	78
Customization of the DFS Server and the DFSCM envar Files	80
Customizing the DFS Server envar Files	80
Customizing the DFSCM envar File	81
Chapter 8. Defining DFS Administrators and Users	83
Creating DCE User Accounts for DFS Administrators and Users	83
Defining a DFS Administrator	84
System Control Machines and Domains	85
DFS Administrator Considerations for a DCE Cell	85
DFS Server Principal Considerations for a DCE Cell	85
Description of DFS Administrative List Files	86
Creating the Initial DFS Administrator(s)	87
Disable/Enable DFS Authorization (NoAuth File)	88
Creating the admin.bos File	89
Adding Principals and Groups to Administrative Lists	89
Example of Updating the admin.bos File	90
Example of Updating the admin.fl File	90
Example of Updating the admin.ft File	91
Example of Updating the admin.bak File	91
Example of Updating the admin.up File	91
Mapping OS/390 User IDs for HFS and RFS Access	92
Registering Users With the OS/390 Security Subsystem	92

Mapping DCE User IDs to OS/390 User IDs	92
Using the Identity Mapping File Method	93
Creating the Identity Mapping Input File (dfsidmap)	93
Creating the Identity Mapping File	95
Setting the _IOE_MVS_IDMAP Environment Variable	96
Ensuring the _IOE_MVS_IDMAP_SAF Environment Variable Is Set NO	96
Modifying and Deleting Identity Mapping Entries	97
Allowing Foreign Users Access to the DFS Server	97
Using the RACF Identity Mapping Function	97
Creating the RACF DCEUIDS Profile	97
Creating the RACF DCEUIDS APPLDATA OS/390 ID Entries	98
Ensuring _IOE_MVS_IDMAP_SAF Environment Variable is Set YES	98
Modifying and Deleting RACF DCEUIDS APPLDATA OS/390 ID Entries	98
DFS Anonymous User Considerations	98
 Chapter 9. Exporting Data in DFS	 101
Creating Aggregates in DFS	101
Creating Non-Local File System Aggregates in DFS	101
Creating DCE Local File System Aggregates in DFS	103
Exporting File System Data Using DFS	103
Exporting Non-Local File System Data	104
Exporting DCE Local File System Data	105
Setting Up the root.dfs Fileset	106
Steps to Define a root.dfs Fileset	106
Creating Mount Points for Filesets	109
Steps to Create and Export a DCE Local File System Aggregate	110
Steps to Create and Export an HFS Aggregate	113
Steps to Create and Export an RFS Aggregate	115
 Chapter 10. Deconfiguring DFS	 117
DCE Deconfiguration Considerations for DFS	118
Update (upserver) Deconfiguration Considerations	118
DFS Component Deconfiguration	119
Deconfigure the DFS Client (DFSCM)	119
Deconfigure DFS Servers	119
Deconfigure Backup Database Server (bakserver)	120
Completing the Backup Database Server Deconfiguration	121
Deconfigure the Fileset Location Database Server (flserver)	121
Completing the Fileset Location Database Server Deconfiguration	122
Deconfigure the Fileset Server (ftserver)	123
Deconfigure the DFS File Server	123
Using DFSCONF to Deconfigure DFS Servers	124
Using DFSCONF to Deconfigure the DFS Client	126
Removing DFS Servers from a Host System	127
Remove the DFS Client from a Host System	127
DFS Deconfiguration Steps for DCE Host Name Change	128
Recommended Reconfiguration Steps When Changing DCE Host Name	128
DCE Host Name Change While DFS is Configured	128
 Chapter 11. Starting and Stopping DFS Components	 131
Starting and Stopping the DFS Client (DFSCM)	131
Who Can Start and Stop DFSCM?	131
DFSCM Initialization	131
Starting the DFS Client (DFSCM)	132

Stopping and Restarting the DFSCM	133
DFSCM Relationship to DCE and DFS Servers	133
Starting and Stopping the DFS Servers	133
Who Can Start and Stop DFS Server and boserver Processes?	133
Order of Starting DFS Daemons	134
Starting DFS Server with System START Command	134
Stopping the DFS Server with System STOP Command	134
Controlling DFS Server Processes with System MODIFY Command	135
Starting the boserver Processes from OMVS	136
Stopping All the boserver Processes from OMVS	139
DFS Server Relationship to DCE	139
Appendix A. Example DFS Server Configuration Log File (dfsconf.log)	141
Appendix B. Directories and Files	147
Files and Directories in /opt/dfsglobal	147
Symbolic Links to /etc/dfs Created During Installation	147
Directories in /opt/dfslocal	148
Customizable Files in /opt/dfslocal	149
Appendix C. Environment Variables in DFS	153
Environment Variables	153
Appendix D. DFS/SMB Performance and Tuning Guidelines	173
Who Should Read This	173
Introduction	173
Tuning Considerations and Recommendations	174
QUERY Command	176
Reports Available for the DFS/SMB Server	176
Reports Available for the DFS Client	177
RESET Command	178
Data Normalization	178
DFS Server Tuning	179
Workloads	179
Hardware	179
SMB Workloads	179
SMB Service Threads	182
DCE Workloads	183
DCE Service Threads	184
Token Management	185
Virtual Memory File Cache	187
HFS/RFS Caches	190
HFS	193
LFS (Episode)	194
RFS	196
Locking and Serialization	199
Storage Usage	200
DFS Client	201
Workload and Threading	201
Client Caches	202
RPC I/O Performance	205
Disk Caching Considerations	207
Locking	207
Some Final Considerations	208

Appendix E. Notices	209
Trademarks	210
Bibliography	211
OS/390 Distributed File Service Publications	211
OS/390 DCE Publications	211
SecureWay Security Server Publications	212
Index	213

Figures

1.	DFSCM Address Space	10
2.	DFS Address Space	13
3.	DCE Login Panel	56
4.	DFSCONF Main Menu	56
5.	Configuring DFS Servers Panel	57
6.	Configuring the DFS Client Panel	60
7.	Example Identity Mapping Input File (dfsmap)	94
8.	MAPID Program JCL	96
9.	Deconfiguring DFS Servers Panel	124
10.	Deconfiguring the DFS Client (DFSCM) Panel	126
11.	DFS/SMB Server Components	174

Tables

1.	Summary of OS/390 Distributed File Service Updates for OS/390 V2R10	22
2.	Summary of OS/390 Distributed File Service Updates for OS/390 V2R9	22
3.	Summary of OS/390 Distributed File Service Updates for OS/390 V2R8	22
4.	Summary of OS/390 Distributed File Service Updates for OS/390 V2R7	23
5.	Summary of OS/390 Distributed File Service Updates for OS/390 V2R6	23
6.	Summary of OS/390 Distributed File Service Updates for OS/390 V2R5	23
7.	Summary of OS/390 Distributed File Service Updates for OS/390 V1R3	23
8.	Summary of OS/390 Distributed File Service Updates for OS/390 V1R2	24
9.	Summary of New and Changed Distributed File Service Files	36
10.	Summary of New and Changed Distributed File Service Commands	36
11.	Summary of New and Changed Distributed File Service APIs	37
12.	Summary of New and Changed SMB Files	37
13.	Summary of New and Changed SMB Commands	37
14.	Summary of New and Changed Environment Variables	38
15.	Directory Locations for envvar Files on OS/390 DFS	80
16.	Files and Directories in /opt/dfsglobal	147
17.	Symbolic Links Created During Installation	147
18.	Directories in /opt/dfslocal	148
19.	OS/390 DFS Customizable Files	150
20.	Environment Variables	153

About This Book

This book helps system and network administrators configure OS/390 Distributed File Service (DFS). This book is used after the successful installation of OS/390 DFS. Installation is described in the *OS/390 Program Directory*.

OS/390 Distributed File Service includes a DFS** function that is based on the OSF DCE Distributed File Service component. The OSF DCE Distributed File Service includes distributed file system capabilities based on the OSF DCE RPC protocols and security.

Who Should Use This Book

This book is intended for system administrators who understand the basic concepts of the Distributed Computing Environment (DCE). A knowledge of TCP/IP communications also helps administrators to use this book more effectively.

How to Use This Book

This book provides overview information on configuring and deconfiguring OS/390 DFS. It takes you through the steps you will perform to prepare for configuration by running the DFS Server Configuration Program (**DFSCONF**) from TSO to configure or deconfigure DFS.

Note: The chapters in this book describe procedures that should be performed in the order that they appear.

In addition, there are appendices:

- Appendix A, "Example DFS Server Configuration Log File (dfsconf.log)" on page 141, shows the contents of a log file created after a typical configuration on an OS/390 DFS host system using the OS/390 DFS Server Configuration Program (**DFSCONF**).
- Appendix B, "Directories and Files" on page 147, lists the important files and subdirectories shipped as part of the OS/390 DFS product. A listing of symbolic links created during the installation process is also provided.
- Appendix C, "Environment Variables in DFS" on page 153, lists and describes some of the DCE environment variables and all of the OS/390 DFS environment variables.
- Appendix D, "DFS/SMB Performance and Tuning Guidelines" on page 173, describes information about tuning the server and/or client for optimal performance for administrators of the Distributed File Service (DFS/SMB).

Any reference to DCE in this book is understood to specifically mean DCE for the IBM OS/390 operating system, unless otherwise noted.

Where to Find More Information

Where necessary, this book references information in other books using shortened versions of the book title. For complete titles and order numbers of the books for all products that are part of OS/390, see the *OS/390 Information Roadmap*, GC28-1727.

To understand most of the topics that are covered in this book, refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

For information about installing DFS components, refer to the *OS/390 Program Directory*.

Information about administrative commands and syntax is available in the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

For information concerning Distributed File Service-related messages, refer to the *OS/390 Distributed File Service Messages and Codes* book.

Information about DFS configuration on other IBM systems can be found in the configuration guide for those systems.

Softcopy Publications

The OS/390 Distributed File Service library is available on the following CD-ROMs. The CD-ROM online library collections include the IBM Library Reader, which is a program that enables you to view the softcopy books.

SK2T-6718 *OS/390 PDF Library Collection*

This collection contains the set of unlicensed books for the current release of OS/390 in Portable Document Format (PDF) files. You can view or print these files with the Adobe Acrobat reader.

SK2T-6700 *Online Library Omnibus Edition OS/390 Collection*

This softcopy collection contains a set of unlicensed books for OS/390 and related products. The collection contains the publications for multiple releases of these products.

Internet Sources

- **OS/390 Online Library**

Softcopy OS/390 publications are also available for web-browsing and PDF versions of the OS/390 publications for viewing or printing using Adobe Acrobat Reader at the following URL:

<http://www.ibm.com/s390/os390/>

You can also provide comments about this book and any other OS/390 documentation by visiting that URL. Your feedback is important in helping to provide the most accurate and high-quality information.

Unsupported Services and Commands

The following commands and services are not supported in OS/390 DFS:

- The **TapeConfig** configuration file.
- The **FMSLog** log file.
- The following **bos** commands are not supported in OS/390 DFS:

- **bos getdates** command
- **bos install** command
- **bos prune** command
- **bos uninstall** command.

In addition, the **-newbinary** option is not available in OS/390 DFS for any **bos** commands.

- The following **bak** commands cannot be issued against an OS/390 **bakserver**. These commands may be issued from OS/390 against non-OS/390 **bakserver** processes:
 - **bak labeltape** command
 - **bak scantape** command
 - **bak readlabel** command.
- The **fms** command.
- The **dfsgw** gateway function is provided, see the *OS/390 Distributed File Service DFS Administration Guide and Reference* for complete details. However, the following **dfsgw** commands cannot be issued from OS/390:
 - **dfsgw add** command
 - **dfsgw apropos** command
 - **dfsgw delete** command
 - **dfsgw help** command
 - **dfsgw list** command
 - **dfsgw query** command.
- The **dfstrace** command.

Commands and services not directly supported in OS/390 DFS may be run or requested from other non-OS/390 DFS systems. Exceptions are noted where applicable.

Conventions Used in This Book

This book uses the following typographic conventions:

Bold	Bold words or characters represent system elements that you must enter into the system literally, such as commands.
<i>Italic</i>	<i>Italicized</i> words or characters represent values for variables that you must supply.
Example Font	Examples and information displayed by the system are printed using an example font that is a constant width typeface.
[]	Optional items found in format and syntax descriptions are enclosed in brackets.
{ }	A list from which you must choose an item found in format and syntax descriptions are enclosed by braces.
	A vertical bar separates items in a list of choices.
< >	Angle brackets enclose the name of a key on a keyboard.
...	Horizontal ellipsis points indicate that you can repeat the preceding item one or more times.
\	A backslash is used as a continuation character when entering commands from the shell that exceed one line (255 characters). If the command exceeds one line, use the backslash character \ as the last non-blank character on the line to be continued, and continue the command on the next line.

Note: When you enter a command from this book that uses the backslash character (\) make sure you immediately press the Enter key and then continue with the rest of the command. In most cases, the backslash has been used for ease of readability.

\$ A dollar sign that appears before a command indicates that the example is shown in shell mode, that is, commands are entered from the shell.

A number sign is used for the command prompt in instances where *root* authority is needed (in OS/390 DFS, **root** refers to a user with a **UID = 0**).

This book uses the following keying convention:

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

Entering commands

When instructed to enter a command, type the command name and then press <Return>.

Summary of Changes

Summary of Changes

for SC28-1722-06

OS/390 Version 2 Release 10

This book contains information previously presented in Release 8 of the *OS/390 Distributed File Service DFS Configuring and Getting Started* book, SC28-1722.

New Information

- An Appendix has been added describing the performance and tuning guidelines for the Distributed File Service for DFS and SMB.

Changed Information: The following information has changed:

- **dfs_cpfiles** program.

This book includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of Changes

for SC28-1722-05

OS/390 Version 2 Release 8

This book contains information previously presented in Release 6 of the *OS/390 Distributed File Service Configuring and Getting Started* book, SC28-1722-04.

Changed Information

- The OS/390 Distributed File Service now provides support for the SMB File/Print Server as well as the DFS client and server support for a DCE.
- A new environment variable, **_IOE_PROTOCOL_RPC**, has been added to allow the Distributed File Service server to start with the DCE RPC support disabled.
- The DFS customizable files reside in the path **/etc/dfs** instead of a separate HFS dataset mounted at **/usr/lpp/dfs/local**.

This book includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Summary of Changes

for SC28-1722-04

OS/390 Version 2 Release 6

This book contains information previously presented in Release 5 of the *OS/390 Distributed File Service Configuring and Getting Started* book, SC28-1722-03.

Changed Information

- As part of the name change of OpenEdition to OS/390 UNIX System Services, occurrences of OpenEdition have been changed to OS/390 UNIX System Services or its abbreviated name, OS/390 UNIX. OpenEdition may continue to appear in message, panel text, and other code locations.
- The DFSKERN process of the OS/390 DFS File Server can be run in a separate address space. This can provide better recovery characteristics in DFSKERN failure situations.

- The DFS customizable files reside in the path **/etc/dfs** instead of a separate HFS dataset mounted at **/usr/lpp/dfs/local**.

**Summary of Changes
for SC28-1722-03
OS/390 Version 2 Release 5**

This book contains information previously presented in Release 3 of the *OS/390 Distributed File Service Configuring and Getting Started* book, SC28-1722-02. This version of the book has been reorganized to help you better explain the steps required for configuring DFS.

**Summary of Changes
for SC28-1722-02
OS/390 Version 1 Release 3**

This book contains information previously presented in Release 2 of *OS/390 OpenEdition DCE Distributed File Service Configuring and Getting Started* book, SC28-1722-01.

The following summarizes the changes to that information.

New Information:

- OS/390 DFS configures using ISPF dialogs. Chapter 7, “Configuring DFS” on page 47 provides this information.
- The DFS Server exports OS/390 record data. Chapter 9, “Exporting Data in DFS” on page 101 provides this information.

**Summary of Changes
for SC28-1722-01
OS/390 Version 1 Release 2**

This book contains information previously presented in Release 1 of *OS/390 OpenEdition DCE DFS: Configuration and Release Notes*, SC28-1722-00.

The following summarizes the changes in this addition.

New Information: Information on the following processes now supported in Release 2 of OS/390 has been added:

- DFS Client (**DFSCM**)
- Fileset Location Server (**flserver**)
- Backup Server (**bakserver**).

Information on configuring (and deconfiguring) the above processes has been added to Chapter 7, “Configuring DFS.”

Planning information has been added regarding the DFS Client in Chapter 3, “DFS Client (DFSCM) Considerations.”

Information on how to start and stop the DFS Client in OS/390 has been added to “Starting and Stopping the DFS Client (DFSCM).”

Changed Information: Information on configuring DFS components has been changed in “Using the DFS Configuration Program (DFSCONF)” to reflect support for the DCE **dcecp** administrative interface.

Updates to the following sections have been made:

- Appendix B, “Directories and Files”
- Appendix C, “Environment Variables in DFS.”

Information has been changed throughout to reflect changes made by upgrading to Open Software Foundation's Revision 1.1 level.

**Summary of Changes
for SC28-1722-00
OS/390 Version 1 Release 1**

This book contains information previously presented in the *OpenEdition DCE DFS for MVS/ESA: Configuration and Release Notes*, SC24-5723-00, which was a feature of MVS/ESA System Product Version 5 Release 2.2.

Chapter 1. Introduction to OS/390 DFS

The OS/390 Distributed File Service supports the client/server model for distributed file systems. The Distributed File Service provides DCE based OS/390 DFS client and server support. It also provides OS/390 SMB File/Print Server support which is not dependent on DCE. The OS/390 Distributed File Service server running on OS/390 can provide support for either the DFS file serving environment for DCE, the SMB File/Print Server environment or both environments simultaneously. HFS file data exported from the OS/390 Distributed File Service server for workstation user access can be accessed and shared by both DFS clients, SMB workstation users, and OS/390 UNIX System Services users and applications.

For further information on the OS/390 SMB File/Print Server support, refer to the *OS/390 Distributed File Service SMB Administration Guide and Reference*.

IBM's OS/390 Distributed File Service (DFS) allows users to access and share data in a distributed environment across a wide range of IBM and non-IBM platforms. OS/390 is based on the source code developed by the Open Software Foundation (OSF). In the environment, OS/390 DFS provides access to the Hierarchical File System (HFS), Record File System (RFS), and OSF's DCE Local File System. The benefits of these file systems are detailed in this chapter.

This chapter discusses the following topics:

- "OS/390 DFS Features"
- "Using the OS/390 Hierarchical File System" on page 3
- "Using the OS/390 Data Sets" on page 3
- "Using OS/390 DFS Commands in TSO" on page 3
- "Using the OS/390 DFS Client (DFSCM)" on page 4.

OS/390 DFS Features

The Distributed File Service (DFS) allows users to access and share files stored on a File Server anywhere on the network, without having to know the physical location of the file. Files are part of a single namespace. Therefore, no matter where in the network a user is, the file can be found using the same name.

OS/390 DFS includes functionality to configure DFS in a DCE cell and export the OS/390 Hierarchical File System (HFS) and the Record File System (RFS). For further information on DCE cells, see the *OS/390 DCE Administration Guide*, SC28-1584.

Note: IBM OS/390 DCE Base Services must be installed and running before installing OS/390 DFS.

For information regarding other corequisites and prerequisites for OS/390 DFS, see the *OS/390 Program Directory*.

Using OS/390 DFS with HFS and RFS file systems provides the following benefits:

- File location transparency. DFS keeps track of the location of files at all times so users do not have to.
- Data consistency for distributed data through the use of advanced token management.
- Access to the OS/390 Hierarchical File System and Record File System.
- Uniform access to enterprise-wide data through unique naming of files.
- Improved security based on Kerberos authentication.

- Increased manageability through the use of distributed databases to track file location and authentication.

There are no DFS application programming interfaces. Access to DFS resources is transparent to the application.

DFS Client Support

OS/390 DFS now provides support for the Distributed File Service Client (**DFSCM**). This support allows an OS/390 user or application to access directories and/or files in the DFS global namespace. (Note that the terms DCE global namespace and DFS global namespace can be used interchangeably.) The DFS global namespace includes any directories and files exported by DFS servers running on either the local OS/390 system, a remote OS/390 system, or on non-OS/390 systems. OS/390 Hierarchical File System (HFS), Record File System (RFS), or DCE Local File System files exported by DFS on a remote OS/390 system running an OS/390 DFS server are part of this namespace. The DFS servers can be in the same or different DCE cells.

DFSCM uses the DCE Remote Procedure Call (RPC) security mechanisms based on Kerberos authentication. Support for directory and file authorization based on DCE Access Control Lists (ACLs) is included with **DFSCM**. Optimal performance through memory or local disk caching is also included with **DFSCM**.

Refer to the “DFSCM Address Space and PROCLIB” on page 9 for more information on the OS/390 implementation of the DFS client.

DFS Server Support

OS/390 DFS provides support for all the Distributed File Server daemons. In OS/390, the term daemons, server processes, and servers are used interchangeably. The DFS servers export file data for access by DFS clients. Refer to the “DFS Address Space, Processes, and PROCLIBs” on page 13 for more information on the OS/390 implementation of the DFS servers.

DCE Local File System Support

OS/390 DFS also includes support for the DCE Local File System. The DCE Local File System is part of the Open Software Foundation DFS product. DCE Local File System is a high-performance, log-based file system. It provides enhanced performance and reliability over traditional file systems by providing improved data storage and management. The DCE Local File System provides support for the extended DFS fileset management functionality of setting fileset quotas, moving filesets, creating backup filesets, and limiting access to DCE Local File System data by supporting DCE Access Control Lists (ACLs).

DCE Local File System support provided with OS/390 DFS includes:

- OS/390 kernel extension for the DCE Local File System.
- Greater reliability through data replication across multiple servers.
- Better availability of data through data replication.
- The following DCE Local File System processes and commands:
 - Fileset Replication server (**repserver**)
 - DCE Local File System salvager (**salvage**)
 - DCE Local File System aggregate initialization (**newaggr**, **growaggr**).

Using the OS/390 Hierarchical File System

OS/390 Hierarchical File Systems (HFS) can be exported to the DFS filesystem. In the DFS documentation this is referred to as a non-Local File System fileset. The entire HFS file system is registered in the Fileset Location Database machine as one DFS fileset. Additional DFS filesets cannot be created within that HFS file system.

An HFS file system has to be locally mounted on the DFS file server machine before it is exported to the DFS filesystem. An HFS data set is mounted at its root.

See the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information about exporting HFS file systems (non-Local File System partitions) to the OS/390 DFS filesystem.

There are some limitations when using HFS file systems with DFS. The fileset management functions of creating backup filesets, moving filesets, and fileset replication are not supported for HFS filesets. In addition, only UNIX mode bits can be used to protect HFS files that have been exported to the OS/390 DFS filesystem.

An inconsistent DFS token state can occur if there is any local file system activity occurring on an HFS file system before it is exported to the DFS filesystem. To avoid this state, when configuring the DFS File Server on OS/390 to export HFS file systems, ensure that DFS is started at system start time (see “Using the DFS Configuration Program (DFSCONF)” on page 53). It is not recommended that **/**, **/tmp**, **/user**, **/bin**, **/etc**, or **/var** be exported since the operating system generates activity on these systems at system restart time. If **/home** is exported with DFS, it should be exported before local system logins are permitted. Consequently, the DFS File Exporter should be started as early in the OS/390 startup sequence as is possible.

Using the OS/390 Data Sets

OS/390 data sets can be exported to the DFS filesystem. In the DFS documentation an exported OS/390 data set is referred to as a non-Local File System fileset, an RFS (Record File System) fileset, or simply as an RFS (Record File System). The entire RFS is registered in the Fileset Location Database machine as one DFS fileset. Additional DFS filesets cannot be created within that RFS.

There are some limitations when using RFS filesets with DFS. The fileset management functions of creating backup filesets, moving filesets, and fileset replication are not supported for RFS filesets.

See the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information about exporting RFS filesets (non-Local File System partitions) to the OS/390 DFS filesystem and using OS/390 data sets from DFS clients.

Using OS/390 DFS Commands in TSO

In OS/390 DFS, most commands can be run from Time Sharing Option Extensions (TSO/E), the shell, or submitted as batch jobs. To ensure TSO/E users can run these commands, the SIOEEXEC library containing the OS/390 DFS REXX EXECs for TSO/E must be either allocated to SYSEXEC or concatenated with existing SYSEXEC libraries. This is normally done during the installation process (for details, see the *OS/390 Program Directory*).

The SIOEEXEC library is supplied with a record format of Fixed Block. Ensure that all libraries in the SYSEXEC concatenation have the same record format.

To ensure that the SYSEXEC library SIOEEXEC is available to be searched, execute the following TSO/E command:

```
EXECUTIL SEARCHDD(YES)
```

You may also, optionally, add the SIOEEXEC library to SYSPROC instead of SYSEXEC. To do so, ensure all libraries in the SYSPROC concatenation have the same record format. For further information, see the *OS/390 Program Directory*.

Using the OS/390 DFS Client (DFSCM)

The OS/390 DFS client (**DFSCM**) can be used to access directories and files in the DFS global namespace from OS/390.

The DFS global namespace is identified by the pathname prefix, */...* Any directory or file with this pathname prefix is considered a part of the DFS global namespace. In addition, any directories or files exported by DFS servers on any system platform are part of the DFS global namespace. The DFS namespace includes any Hierarchical File System (HFS) or Record File System (RFS) (also referred to as a non-Local File System) or DCE Local File System files exported by a DFS server running on any OS/390 system. This includes either the local OS/390 system where **DFSCM** is running or a different OS/390 system.

In addition, any OS/390 command that references a pathname beginning with */...* uses **DFSCM** services to access the requested directory or file information.

Chapter 2. Global and Cell Considerations

The purpose of this chapter is to assist you in planning for the installation and configuration of OS/390 DFS. "Using the DFS Configuration Program (DFSCONF)" on page 53 describes the configuration process for setting up and configuring DFS servers.

This chapter discusses the following topics:

- "Planning for Access Control"
- "Cell Filespace"
- "DFS Administrative Domains" on page 6
- "DFS Administrative Lists" on page 6
- "Determining the Roles of DFS Machines" on page 6
- "Setting Up Filesets" on page 6.

Note: After a DCE cell is configured and named you cannot change the name after DFS is configured and used.

Planning for Access Control

In planning for access control, it is important to keep the level of access control in your cell restrictive enough to ensure that security is maintained. A special set of individuals or a special group can be given permission to create accounts and groups in the root directory of the Security namespace. The **acct-admin** group is created when you configure DCE. **acct-admin** is the only group that can create accounts and groups in the root directory of the Security space.

While maintaining an adequate level of security in your cell, you also need to consider the requirements of administrators who are maintaining DCE services when you set access control levels. For example, if one person is responsible for administration of DFS in your cell, that person may need to add servers to the Security and CDS namespaces. On the other hand, an administrator responsible for the Security Service manages the Security server but does not control the DFS filepace.

Following are some of the groups created when you configure DFS:

- | | |
|------------------------|---|
| dfs-admin | This group administers DFS File Servers and related DFS functions. |
| dfs-fs-servers | This group contains abbreviated forms of the DFS server principal of all Fileset Database machines. |
| dfs-bak-servers | This is the security group to which all database servers belong. |

Cell Filespace

This section contains guidelines for planning your cell's filepace.

The filepace begins under the cell root at the **/./fs** junction to DFS from the CDS namespace. The notation **/./** is set up by default to be equivalent to **/../cellname**. The notation **/:** is set up by default to be equivalent to **/./fs**. Thus, the notation **/:/usr/user_name** is equivalent to **/./fs/usr/user_name** which in turn is equivalent to **/../cellname/fs/usr/user_name**.

Some parts of DFS run in the host machine's kernel. This kernel function must be present on your machine before you run DFS. On OS/390, this function is implemented as MVS kernel extensions that are loaded when you configure DFS on the machine.

DFS Administrative Domains

A DFS administrative domain is a collection of machines in the same cell configured for administration as a single unit. In a single cell, you can have one or many administrative domains, depending on the size of your organization. Organizing DFS server machines into different administrative domains simplifies the management of the cell filesystem by creating smaller units for administration.

DFS Administrative Lists

DFS administrative lists are files that define the principals and groups that can perform actions affecting specific server processes on a server machine. There is one DFS administrative list for each DFS server process running on a machine. For example, a server's **admin.bos** file defines who has administrative rights to the BOS Server (**boserver**), and thus determines who can manipulate and maintain server processes on that one server. Groups, as well as individual users, can be placed in an administrative list. Each server machine stores administrative lists for its processes on its local HFS file system. A process automatically creates its initial administrative list when it is started if the list does not already exist.

Determining the Roles of DFS Machines

Follow recommendations when you assign roles to the DFS machines in your cell. The first DFS machine that you configure needs to function as a System Control Machine. The System Control machine runs the Update Server (**upserver**) that is responsible for distributing the DFS configuration. Next, you configure a Fileset Location server, the server that maintains the fileset location database. Then, you configure a DFS File Server machine. Usually, the first DFS File Server is the machine on which you plan to put the root fileset (**root.dfs**). After configuring the DFS File Server, you should create the **root.dfs** fileset. Then configure a machine as a DFS client (if a machine is not already a DFS client). “Setting Up Filesets” contains further information about **root.dfs**.

Machines that you configure as DFS servers run the processes required to be File Servers. Be sure the machine you choose has enough space to store DCE Local File System filesets. The amount of free space you need depends on how much data you plan to store in DCE Local File System filesets. Filesets on File Servers can store DFS client binaries in addition to user files. These filesets can also be distributed on other File Server machines in your cell. In addition, if your domain has only one server machine, this machine must run all processes and fill all required machine roles. For example, in addition to being a System Control machine, this machine must be a File Server and a Fileset Location server. An odd number of DFS Fileset Location machines is recommended. Additionally, an odd number of DFS Backup Database Servers is also recommended.

Setting Up Filesets

Consider the following recommendations and restrictions when you set up filesets:

- Fileset names must be limited to 111 characters or less.
- Every cell must include **root.dfs**. The root fileset can be a DCE Local File System fileset or it can be an HFS fileset. If you plan to use replication, your **root.dfs** must be a DCE Local File System fileset and you need to follow the steps described in the *OS/390 Distributed File Service DFS Administration Guide and Reference*, which explains how to create **root.dfs** as a DFS Local File System fileset and create a read-write mount point for the fileset below the top level of the cell's filesystem. See also Chapter 9, “Exporting Data in DFS” on page 101 for information on creating **root.dfs**.

- You should use a common prefix when naming related filesets. This aids in manipulating and grouping related filesets. It also relates the fileset's name to its mount point.
- You can group filesets on the same partition of a File Server machine. This can localize the effects of an outage, but you also need to consider factors such as number of File Server machines and load balancing before grouping filesets.
- You can replicate filesets for load balancing and to make fileset contents more available. Replication is appropriate for filesets that are read much more often than they are written, such as filesets containing installed executable files. Replication is not supported for non-Local File System filesets.
- Consider the disk space a fileset requires before setting up filesets.
- You should create the initial Access Control Lists (ACLs) for the filesets you want to set up. For further information on creating ACLs, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Chapter 3. DFS Client (DFSCM) Considerations

This chapter describes the OS/390 DFS client machine (**DFSCM**). The term Cache Manager is also used to refer to the **DFSCM**. On OS/390, the **DFSCM** is an independent part of the Distributed File Service.

In DFS, the **DFSCM** is responsible for the local caching of file and directory data on machines used as DFS clients. The **DFSCM** enables users on an OS/390 system to access directories and files in the DFS global namespace.

This chapter discusses the following topics:

- “DFSCM Relationship to DCE and DFS Servers”
- “DFSCM Address Space and PROCLIB.”

DFSCM Relationship to DCE and DFS Servers

To configure and use the **DFSCM**, DCE must be running on the same host system. **DFSCM** waits for DCE to start before completing initialization. If DCE is stopped while the **DFSCM** is running, the **DFSCM** cannot continue to run successfully.

The **DFSCM** does not require the installation and running of DFS servers on the same OS/390 system. The OS/390 system, where the **DFSCM** is run, must be part of a DCE cell. The **DFSCM** can then access file data exported by DFS file servers running in the same or different DCE cell.

DFSCM Address Space and PROCLIB

The DFS Client in OS/390 DFS is implemented as an a physical file system running as a colony address space named **DFSCM**.

The **DFSCM** address space is started when OS/390 is initialized. The OS/390 system parmlib member **BPXPRMxx**, read during initialization, identifies the physical file systems.

The variable, *xx*, is an installation specific suffix used to uniquely identify system parmlib members. For more information, see “System Parmlib Member BPXPRMxx Entry for the DFSCM” on page 72.

- | Additionally, the proclib member **IOEP0002** (alias **DFSCM**) is supported with JCL to run the DFSCM. Note
- | that the proclib members to run the DFS server processes are described in “DFS Address Space,
- | Processes, and PROCLIBs” on page 13. Refer to the *OS/390 Program Directory* for proclib member customization considerations for Wave 2 elements.

Refer to the *OS/390 UNIX System Services File System Interface Reference*, SC28-1909, for more information on the physical file system and colony address space.

The **DFSCM** daemons **ioecmini**, **ioedfsd**, and **ioellogin** run as individual processes in the **DFSCM** address space as shown in Figure 1 on page 10.

OS/390 DFS Client Environment

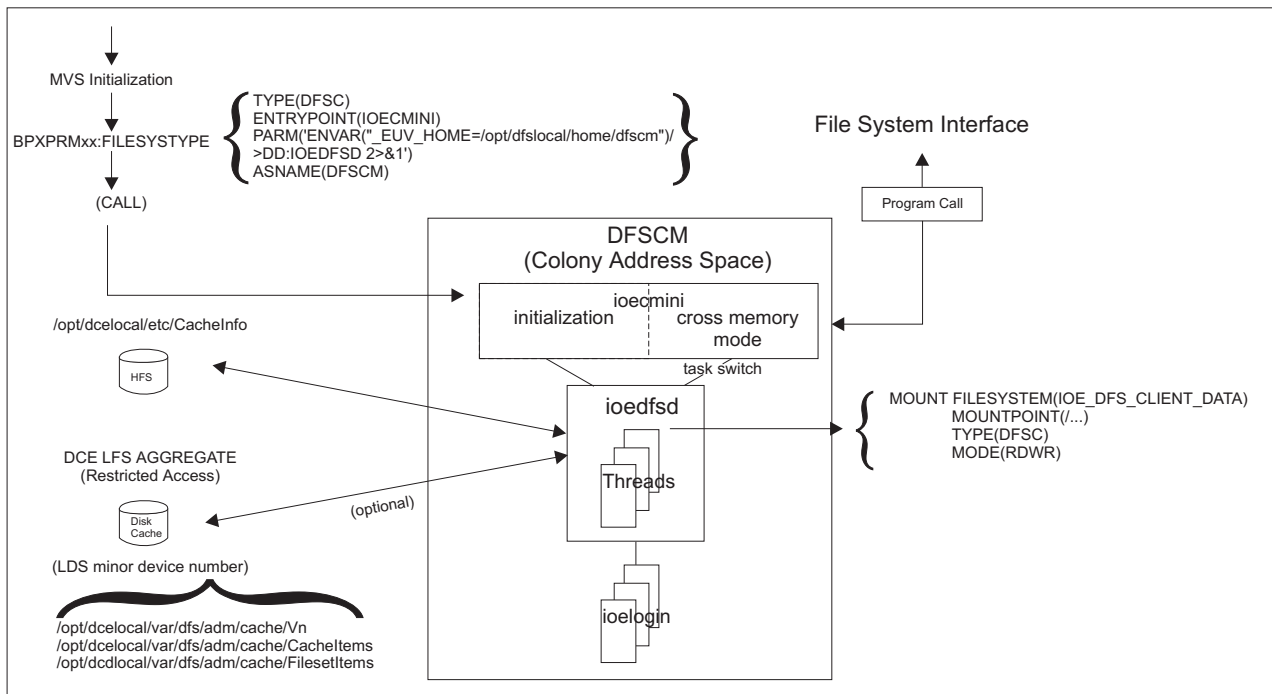


Figure 1. DFSCM Address Space

The **DFSCM** is composed of three daemons: **ioecmini**, **ioedfsd**, and **ioellogin**. These processes are controlled at the address space level and can not be stopped or started independently. A description of the function performed by each process is:

ioedfsd The **ioedfsd** process controls the Cache Manager on OS/390 DFS. The functions provided by the DFS processes, **dfsd** and **dfsbind**, are combined in the **ioedfsd** process.

ioecmini The **ioecmini** process is invoked during **OMVS** initialization to perform **DFSCM** initialization services. These services include starting the other **DFSCM** daemons: **ioedfsd** and **ioellogin**. After returning to allow **OMVS** initialization to continue, **ioecmini** waits for DCE initialization to complete before mounting the HFS file system name that identifies the DFS global namespace at `/...`. The **ioecmini** process also queues work for the **ioedfsd** process.

ioellogin The **ioellogin** process is invoked when a DCE single sign-on is required.

Important Note to Users

The **ioedfsd** process is also unique in OS/390 in that it combines the functions of the traditional Cache Manager **dfsd** and **dfsbind** processes.

Before running the **DFSCM** on an OS/390 host system you must configure it using **DFSCONF**, see Chapter 7, “Configuring DFS” on page 47 for more information.

Control and customization of several **DFSCM** features is accomplished on OS/390 by the following:

- Modifying the local HFS `/opt/dcelocal/etc/CacheInfo` file;
- Adding Cache Manager initialization parameters to the `_IOE_CM_PARMS` parameter in the `/opt/dfslocal/home/dfscm/envvar` file;
- Adding parameters to the **FILESYSTYPE TYPE(DFSC)** entry in the system parmlib member **BPXPRMxx**; or

- Enabling **DFSCM** to use DCE single sign-on.

Refer to “Completing the DFS Client (DFSCM) Configuration” on page 72 for more information.

On OS/390 DFS, the DFS Cache Manager is invoked when OS/390 is started. It can be stopped and restarted without stopping OS/390. Cache Manager customization and control parameters can be specified in the local HFS **/opt/dcelocal/etc/CacheInfo** file or in the **/opt/dfslocal/home/dfscm/envvar** file. The parameters are effective when the Cache Manager is started or restarted. Refer to Chapter 11, “Starting and Stopping DFS Components” on page 131 for more information.

Chapter 4. DFS Server Considerations

On OS/390, DFS servers run as processes in the DFS address space. DFS servers are configured to run a certain set of software. This software is made up of at least one daemon and, in some cases, one or more additional programs that comprise the server side of a DFS component. DFS servers also run the software that makes up the DFS Client configuration.

This chapter discusses the following topics:

- “DFS Address Space, Processes, and PROCLIBs”
- “DFS Administration Utilities” on page 16
- “DFS Server Relationship to DCE” on page 16.

DFS Address Space, Processes, and PROCLIBs

OS/390 DFS supports all types of DFS servers. DFS servers can assume different roles as described below. The DFS system requirements vary, depending on the DFS servers that are configured and in use.

After DFS has been configured and initialized, the DFS server address space (**DFS**) can be started. The DFS daemons run as individual processes in a single **DFS** address space; or as individual processes in the DFS server and DFSKERN address space as illustrated in Figure 2.

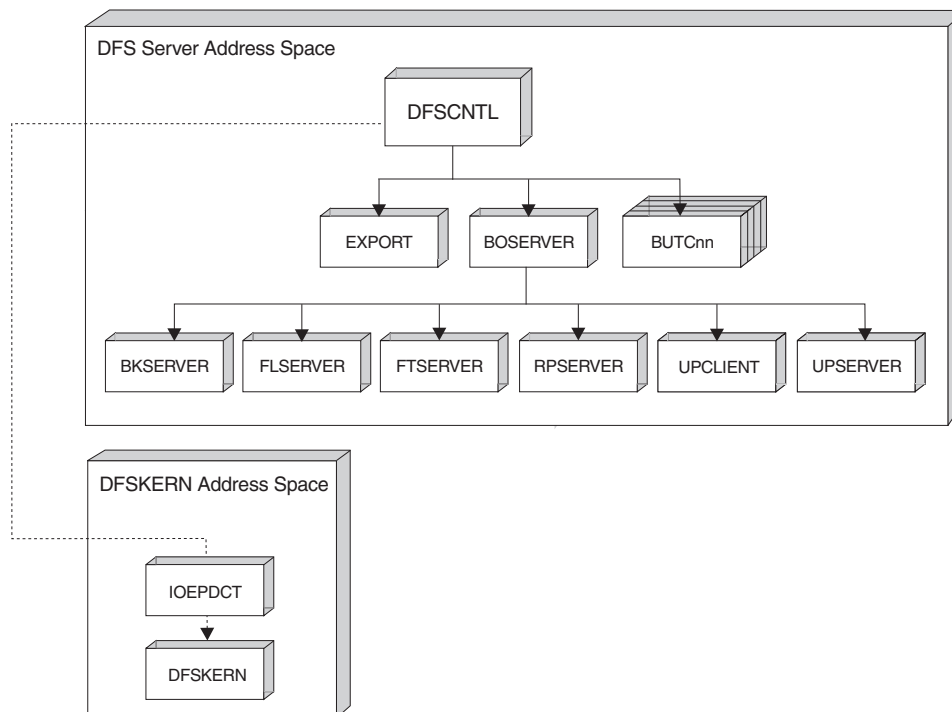


Figure 2. DFS Address Space

- | The proclib member **IOEP0001** (alias **DFS**) is supplied with JCL to run the DFS server address space.
- | The proclib member **IOEP0003** (alias **DFSKERN**) is supplied with additional JCL to run the DFSKERN processes in a separate address space when the environment variable, **_IOE_DAEMONS_IN_AS=DFSKERN**, is specified in the file **/opt/dfslocal/home/dfscntl/envvar**. Note that the proclib member **IOEP0002** (alias **DFSCM**) has JCL to run the DFS client as described in “DFSCM

- Address Space and PROCLIB” on page 9. Refer to the *OS/390 Program Directory* for proclib member customization considerations for Wave 2 elements. It is recommended that the DFSKERN processes be run in an address space separate from the other DFS server processes to take full advantage of the system termination processing.

Important Note to Users

In DFS, there are two types of processes: **simple** and **cron**. A **simple** process is defined as a continuous process that runs independently of any other processes on a server machine. All standard DFS processes are **simple** processes. A **cron** process refers to a process that runs independently of any other processes; however, unlike a **simple** process, a **cron** process runs periodically, not continuously.

In OS/390 DFS, **simple** processes are not identified by complete path names to the binary files for the processes. All **simple** process commands in OS/390 DFS are members of the SIOELMOD data set created during installation (for further information, refer to the *OS/390 Program Directory*).

Also, as all OS/390 process names and PDS member names are limited to eight or less characters, the **bakserver**, **repserver**, and **boserver** process member names are shortened to **bkserver**, **rpserver**, and **boserver**, respectively.

The DFS Control Task, **dfscntl**, is the *parent process* to all the DFS server processes including those that may be running in a separate DFSKERN address space. All requests to **DFS** are directed to **dfscntl** to perform the requested action.

Once DFS is configured, all servers that have been configured can be started (see Chapter 7, “Configuring DFS” on page 47). DFS servers can be started and stopped using the OS/390 system commands **start** and **modify** as described later in this chapter.

The content of the **/opt/dfslocal/etc/ioepdcf** file (also referred to as the daemon configuration file) is used to determine the DFS server processes that are started by **dfscntl** (see “Customizing the ioepdcf File” on page 78). Initializing DFS on OS/390 automatically starts all processes as specified in the **ioepdcf** file. The DFS processes identified in the **ioepdcf** file are:

dfscntl The DFS Control Task which controls the **dfskern**, **export** (and **unexport**), **boserver**, and **butcnn** processes.

The **dfscntl** and all of its child processes are controlled by the OS/390 system command, **modify**. All options for this command are specified in the DFS **ioepdcf** file.

The name, **dfscntl**, is an alias in the SIOELMOD load library for the load module **IOEDFSCL**.

dfskern The **dfskern** process is the DFS kernel program which includes sub-processes that:

- Provide support for DFS file locking and data sharing
- Allow the DFS server to export OS/390 file data to DFS clients
- Provide support for the DCE Local File System
- Allow the DFS server to export OS/390 sequential, PDS(E), and VSAM data sets.

In OS/390 DFS, the **fxd** command is run as part of the **dfskern**. process. All options of this command are specified in the **ioepdcf** file.

The name, **dfskern**, is an alias in the SIOELMOD library for the load module **IOEDFSKN**.

ioepdct The **ioepdct** process is used to initialize the DFSKERN process when DFSKERN is defined to run in a separate address space from the other server processes.

export The DFS **export** process allows aggregates to be exported by issuing the **dfsexport** command. The SIOELMOD load library entry **IOEDFSXP** is run in the **export** process.

unexport DFS **unexport** is a process that allows aggregates to be unexported by issuing the **dfsexport -detach** command. It is not a separate process but runs as part of the **export** process (see above).

boserver The **Basic OverSeer Server** (BOS) processes **bos** commands and monitors the server processes that it starts on the local machine.

The **boserver** can restart the processes it controls and provides an interface for administrative tasks. In OS/390 DFS, the **boserver** runs under the control of the Control Task, **dfscntl**. The names and descriptions of processes started by the **boserver** daemon are listed later in this section.

The name, **boserver**, is an alias in the SIOELMOD load library for the load module **IOEBOSRV**.

butcnn A **BackUp Tape Coordinator** (BUTC) daemon is a tape coordinator server on which backup and restore operations are physically conducted. Valid entries for *nn* are **01** through **08**. The names, **butc01** through **butc08**, are all aliases in the SIOELMOD load library for the load module **IOEBUTC**.

In OS/390 DFS, the **boserver** process, in turn, acts as the *parent process* to the following sub-processes listed below. The **bos create** command is used to control these processes. The processes that must be started by the **boserver** process are:

bkserv The **Backup Server** processes **bak** commands and maintains the Backup Database where information that is used to control the backup and restore of filesets is stored.

The name, **bkserv**, is an alias in the SIOELMOD load library for the load module **IOEBKSRV**.

flserver The **Fileset Location Server** maintains the Fileset Location Database (FLDB) which tracks the location of all DCE Local File System and non-Local File System filesets. The FLDB is a cell-wide database that maps filesets to servers on which they are located,

There must be at least one **flserver** running in a DCE cell.

The name, **flserver**, is an alias in the SIOELMOD load library for the load module **IOEFLSRV**.

ftserver The **Fileset Server** processes **fts** commands which allow DFS administrators to create, delete, duplicate, move, back up, or restore entire filesets.

An **ftserver** must be run on a host system to export any filesets to the DFS namespace.

The name, **ftserver**, is an alias in the SIOELMOD load library for the load module **IOEFTSRV**.

rpserver The **Replication Server** tracks the currency of replicas and updates the versions of data used at each replication site.

Only replicas of DCE Local File System filesets are supported; replicas of HFS and RFS data sets are not supported.

The name, **rpserver**, is an alias in the SIOELMOD load library for the load module **IOERPSRV**.

upclient The **Update Client** daemon checks specified files and directories on the local machine and ensures they match the corresponding files on the System Control machine.

The **upclient** contacts the **upserver** to verify that the most recent version of each DFS administrative list is being used.

The name, **upclient**, is an alias in the SIOELMOD load library for the load module **IOEUPCLN**.

upserver The **Update Server** distributes files such as common configuration files and administrative lists from System Control machines to other servers in a domain.

The name, **upserver**, is an alias in the SIOELMOD load library for the load module **IOEUPSRV**.

Note: In OS/390 DFS, the DFS System Control machine function is implemented as part of the **boserver** and **upserver** process. Also, in OS/390 DFS, the DFS file server support is used to export the DCE Local File System and non-Local File System (HFS and RFS) data to the global DFS namespace. The file server function runs as part of the **dfskern** process and optionally, the **rpserver** process.

DFS Administration Utilities

DFS provides the following administration utilities:

- The **bak** commands help you perform backup tasks for Local File System filesets.
- The **bos** commands help you contact the Basic OverSeer (BOS) Server used to monitor processes on servers in your cell. You can also use the **bos** commands to perform some Security tasks.
- The **cm** commands help you examine and customize Cache Manager attributes for the current instance of the Cache Manager.
- The **dfsexport** command makes DCE Local File System aggregates and non-Local File System partitions available to remote users through use of the File Exporter.
- The **fts** commands help you manage filesets.
- The **growaggr** command increases the size of DCE Local File System aggregates. This process is available **only** for DCE Local File System.
- The **newaggr** command can format a logical volume consisting of one or more OS/390 Virtual Storage Access Method (VSAM) linear data sets (LDS) for use as a DCE Local File System aggregate. This process is available **only** for DCE Local File Systems.
- The **salvage** process checks the DCE Local File System for internal consistency and corrects errors it finds. This process is available **only** for DCE Local File Systems.
- The **scout** program helps you monitor the File Exporters running on File Servers.
- The **udebug** command provides status information relevant to specified DFS database servers.

DFS Server Relationship to DCE

To configure and use the DFS server, DCE must be running on the same host system. The DFS server cannot successfully start if DCE is not running. If DCE stops after the DFS server is initialized, the DFS server support terminates.

Chapter 5. Migration Overview

This chapter discusses migration issues. Your plan for migrating to a new level of OS/390 Distributed File Service should include information from a variety of sources. These sources of information describe topics such as coexistence, service, hardware and software requirements, migration actions, and interface changes.

The following documentation, which is supplied with your product order, provides information about installing your OS/390 system. In addition to specific information about OS/390 Distributed File Service, this documentation contains information about all of the OS/390 elements.

- *OS/390 Up and Running*

This book describes the installation requirements for OS/390 at a system and element level. It includes hardware, software, and service requirements for both the driving and target systems. It also describes any coexistence considerations and actions.

- *OS/390 Program Directory*

This document, which is provided with your OS/390 product order, leads you through the specific installation steps for OS/390 Distributed File Service and the other OS/390 elements.

- *ServerPac Installing Your Order*

This is the order-customized, installation book for using the ServerPac Installation method. Be sure to review "Appendix A. Product Information," which describes data sets supplied, jobs or procedures that have been completed for you, and product status. IBM may have run jobs or made updates to PARMLIB or other system control data sets. These updates could affect your migration.

Within this chapter, you can find information about the specific updates and considerations that apply to this release of OS/390 Distributed File Service.

- "Steps for Migrating" on page 18

This section identifies tasks that must be done for any migration. These tasks are not associated with a specific new function, but must be completed to ensure successful migration to the new release.

- "Distributed File Service Migration Actions" on page 19

This section describes the migration actions that are typically required (or should be considered) whenever you migrate from a previous release of OS/390 DCE Distributed File Service to the current release. Note, if you are migrating from a previous release of OS/390 Distributed File Service SMB you should refer to the *OS/390 Distributed File Service SMB Administration Guide and Reference*.

- "Migration Roadmap" on page 22

This section identifies the migration paths that are supported with the current level of OS/390 Distributed File Service. It also describes the additional publications that can assist you with your migration to the current level.

- "Summary of Interface Changes" on page 36

This section provides a summary of the changes that are made to OS/390 Distributed File Service user and programming interfaces.

Steps for Migrating

The recommended steps for migrating to a new release of OS/390 Distributed File Service are:

1. Become familiar with the supporting migration and installation information for the release.

Determine what updates you need for products that are supplied by IBM, system libraries, and non-IBM products. Review the *OS/390 Up and Running* book and the *OS/390 Introduction and Release Guide* for information about OS/390 Distributed File Service and other OS/390 elements.

2. Develop a migration plan for your installation.

When planning to migrate to a new release of OS/390 Distributed File Service, you must consider high-level support requirements, such as machine and programming restrictions, migration paths, and program compatibility.

3. Obtain and install any required program temporary fixes (PTFs) or updated versions of the operating system.

Call the IBM Software Support Center to obtain the preventive service planning (PSP) upgrade for OS/390 Distributed File Service which provides the most current information about PTFs for OS/390 Distributed File Service. Check RETAIN again just before testing OS/390 Distributed File Service. For information about how to request the PSP upgrade, refer to the *OS/390 Program Directory*. Although the *OS/390 Program Directory* contains a list of the required PTFs, the most current information is available from the IBM Software Support Center.

4. Install the product using the *OS/390 Program Directory* or the *ServerPac Installing Your Order* documentation.

5. Contact programmers who are responsible for updating applications at your installation.

Verify that your installation's applications will continue to run, and, if necessary make changes to ensure compatibility with the new release.

- a. If you are migrating from a previous release of OS/390 DCE Distributed File Service, refer to "Distributed File Service Migration Actions" on page 19.

- b. The Distributed File Service server can also function as a file/print server for workstations that are not in a DCE cell. Files exported to DCE DFS clients can be accessed and shared with these non-DCE clients. Print requests from these clients can be routed to the Infoprint Manager on OS/390. Refer to the *OS/390 Distributed File Service SMB Administration Guide and Reference*.

6. Use the new release before initializing major new function.

7. If necessary, customize the new function for your installation.

8. Exercise the new functions.

Distributed File Service Migration Actions

The following sections describe common activities and considerations that are typically required (or should be considered) whenever you migrate from a previous release of OS/390 Distributed File Service to the current release.

DCE Cell and Host System Considerations

DFS must be configured and customized to run in a DCE cell.

DFS can be running on different host systems in the same DCE cell. Or, DFS can be running on different host systems in different DCE cells. Different releases of DCE and DFS can be running on these host systems in both cases.

This release of DFS can be installed on a host system where DFS is already installed and configured. If DFS is already installed and configured on a host system, existing DFS customizable and configuration files are not replaced during the installation of this DFS release.

If you have DFS configured on a different host system, **you should not** copy the entire HFS data set containing DFS customizable and configuration files to another host system. The DFS customizable and configuration files are contained in the same HFS data set in the directories `/etc/dfs/etc/IBM/` and `/etc/dfs/home/IBM/`. Refer to “DFS Customizable and Configuration Data Set Considerations” on page 20 for a description of the DFS customizable and configuration files that you can copy for use on another DFS host system.

DCE Cell Configuration Considerations

You should deconfigure DFS from a DCE cell before deconfiguring DCE. If you deconfigure DCE without first deconfiguring DFS, the task of reconfiguring (deconfigure/configure) DFS in the DCE cell is more difficult.

If you **have not** reconfigured (deconfigured/configured) DCE in the cell prior to installing this version of OS/390 DFS, you do not need to reconfigure DFS before installing this DFS release.

If you have reconfigured DCE prior to installing this version of OS/390 DFS, you should deconfigure DFS before proceeding.

Note: Refer to Chapter 10, “Deconfiguring DFS” on page 117 for information on how to deconfigure DFS. This information is superseded by any specific instructions on how to deconfigure DFS in the documentation for your current DFS release.

Test DCE Cell Considerations

If you install DFS into a test DCE cell, you can move the new release of DFS into the production at a later time by:

- Using the new DFS executable code in the production DCE cell;
- Creating any new customizable files in the DCE cell using **dfs_cpfiles** program; and
- Optionally updating the customizable files with new parameters available with this release.

DFS Customizable and Configuration Data Set Considerations

In general, the DFS customizable and configuration data sets should not be copied for use by a DFS client or server running in a different DCE cell or on a different system. Certain DFS customizable files are specific to the data sets exported by a DFS server. Since the same data set cannot be exported by a different DFS server the customizable files with data set specific information cannot be copied. Certain DFS configuration files contain DCE cell specific information that cannot be copied and used in a different DCE cell.

Prior to OS/390 Version 2 Release 6, DFS customizable files resided in a separate HFS data set mounted at `/usr/lpp/dfs/local`. Starting in OS/390 Version 2 Release 6, the DFS customizable files reside in the path `/etc/dfs`. During DFS installation, DFS symbolic links are deleted and recreated to link to files in `/etc/dfs`. If you have replaced any DFS symbolic links with user files, they will be deleted during DFS installation. You should save the file data before installing DFS. These files are listed in Appendix B, “Directories and Files” on page 147.

The `dfs_cpfiles` program creates (**not replace**) a required customizable file with default information if the file does not exist. For a list of customizable files see, Table 19 on page 150.

DFSKERN in a Separate Address Space

If DFS was installed and configured under a release previous to OS/390 Version 2 Release 6, the DFS server only runs in a single address space. With OS/390 Version 2 Release 6, the proclib member `IOEP0003` is supplied to run the DFSKERN process and sub-processes in a separate address space when the environment variable, `_IOE_DAEMONS_IN_AS=DFSKERN`, is specified in the file `/opt/dfslocal/home/dfscntl/envvar`. Refer to the *OS/390 Program Directory* (Distributed File Service Customization considerations section) for a description of the RACF entries required. You can also refer to “Customization of the DFS Server and the DFSCM envvar Files” on page 80.

RACF or Equivalent Security System Entries

RACF or equivalent security system entries are required to define the started tasks for the Distributed File Service and the users who can run them. Refer to the *OS/390 Program Directory* (Distributed File Service Customization considerations section) for a description of the RACF entries required.

Year 2000 Support for OS/390 Distributed File Service

OS/390 DFS complies with the Year 2000 requirements, no further migration actions are needed.

For further information on Year 2000 Support, see *The Year 2000 and 2-Digit Dates: Guide GC28-1251*.

For additional information on Year 2000, see: <http://www.ibm.com/IBM/year2000/>.

For information about migrating to OS/390, see the *OS/390 Up and Running* book, GC28-1726.

Migration to New DFS Release

When migrating to a new release of DFS, you must decide whether to install DFS into an existing DCE cell or into a new DCE cell. You must also insure that any DCE cell specific configuration or customizable file data is not inadvertently used in a different DCE cell. Refer to “DCE Cell and Host System Considerations” on page 19 and “DFS Customizable and Configuration Data Set Considerations” for more details.

| If you install DFS into an existing DCE cell and DCE must be deconfigured and configured to accomplish the installation of DCE, you should deconfigure DFS before deconfiguring DCE. Refer to “DCE Cell Configuration Considerations” on page 19 for more details.

| In general, if you are migrating from a previous release of OS/390 DFS (or re-installing this version) and have customized your system, it **is not necessary** to copy and save your customized data prior to installing this version. The customizable files listed in “Customizable Files in /opt/dfslocal” on page 149 are not replaced by the DFS installation or post installation processing.

| Some of the customizable files listed in “Customizable Files in /opt/dfslocal” on page 149 will not be present on your system when installing a new release of DFS (unless you are re-installing this version) because the files are new for this release. Customizable files that do not exist will be created as part of DFS post installation by the **dfs_cpfiles** program from sample files provided in the **/opt/dfsglobal/examples** directory.

| To install a new release of DFS:

- | 1. Install this version of OS/390 DFS using the System Modification Program/Extended (**SMP/E**) program. The SMP/E installation (or re-installation) of this version of OS/390 DFS does not replace any existing customizable file listed in “Customizable Files in /opt/dfslocal” on page 149.
- | 2. Use the DFS shell configuration script program **dfs_cpfiles**, to copy the IBM-supplied example files from the **/opt/dfsglobal/examples** directory to create (not replace) a customizable version of each file in the appropriate directory.

| To use the **dfs_cpfiles** program, the OS/390 shell and utilities must be installed. If this is not the case, then the example files listed in “Customizable Files in /opt/dfslocal” on page 149 that are new for this DFS release must be copied to the correct directory using other means.

| More information on the **dfs_cpfiles** program can be found in Chapter 6, “Post Installation Considerations” on page 41.

- | 3. Determine if any new parameters or variables for this release are applicable for your system and update your configuration files.

| For information on what is new for each release of DFS, refer to “Migration Roadmap” on page 22.

| For further information regarding the customizing of this release of DFS, refer to the sections “Completing the DFS Client (DFSCM) Configuration” on page 72 and “Completing the DFS File Server Configuration” on page 61.

Migration Roadmap

This section describes the migration paths that are supported by the current release of OS/390 Distributed File Service. It also provides information about how you can obtain the OS/390 Distributed File Service migration information from previous releases.

OS/390 Version 2 Release 10 Summary

Table 1 summarizes the updates that have been introduced to OS/390 Distributed File Service in OS/390 Version 2 Release 10 (OS/390 V2R10). If you are migrating from a previous release you should review the information in the detailed section for each item.

Table 1. Summary of OS/390 Distributed File Service Updates for OS/390 V2R10

For Information About:	Refer to page:
NT LM 0.12	25
RFS SMB (SMB support for MVS data sets)	25
Greater than 32 bit Episode files	25

OS/390 Version 2 Release 9 Summary

Table 2 summarizes the updates that have been introduced to OS/390 Distributed File Service in OS/390 V2R9. If you are migrating from a previous release you should review the information in the detailed section for each item.

Table 2. Summary of OS/390 Distributed File Service Updates for OS/390 V2R9

For Information About:	Refer to page:
Encrypted password login	26

OS/390 Version 2 Release 8 Summary

Table 3 summarizes the updates that have been introduced to OS/390 Distributed File Service in OS/390 V2R8. If you are migrating from a previous release you should review the information in the detailed section for each item.

Table 3. Summary of OS/390 Distributed File Service Updates for OS/390 V2R8

For Information About:	Refer to page:
SMB File/Print Server Support	27

OS/390 Version 2 Release 7 Summary

Table 4 on page 23 summarizes the updates that have been introduced to OS/390 Distributed File Service in OS/390 V2R7. If you are migrating from a previous release you should review the information in the detailed section for each item.

Table 4. Summary of OS/390 Distributed File Service Updates for OS/390 V2R7

For Information About:	Refer to page:
DFS Server Multi-home Support	28
Protected RPC	28
OSF/Transarc compatibility	28

OS/390 Version 2 Release 6 Summary

Table 5 summarizes the updates that have been introduced to OS/390 Distributed File Service in OS/390 V2R6. If you are migrating from a previous release you should review the information in the detailed section for each item.

Table 5. Summary of OS/390 Distributed File Service Updates for OS/390 V2R6

For Information About:	Refer to page:
DFSKERN in a Separate Address Space	30
DFS Backup	30
File basis translation	30

OS/390 Version 2 Release 5 Summary

Table 6 summarizes the updates that have been introduced to OS/390 Distributed File Service in OS/390 V2R5. If you are migrating from a previous release you should review the information in the detailed section for each item.

Table 6. Summary of OS/390 Distributed File Service Updates for OS/390 V2R5

For Information About:	Refer to page:
AIX compatibility	31
DFS RAS/NLS message support	31
OSF DCE 1.2.1 enhancements	31
RFS enhancements	31

OS/390 Version 1 Release 3 Summary

Table 7 summarizes the updates that have been introduced to OS/390 Distributed File Service in OS/390 V2R3. If you are migrating from a previous release you should review the information in the detailed section for each item.

Table 7. Summary of OS/390 Distributed File Service Updates for OS/390 V1R3

For Information About:	Refer to page:
Record Data	33

OS/390 Version 1 Release 2 Summary

Table 8 on page 24 summarizes the updates that have been introduced to OS/390 Distributed File Service in OS/390 V2R2. If you are migrating from a previous release you should review the information in the detailed section for each item.

<i>Table 8. Summary of OS/390 Distributed File Service Updates for OS/390 V1R2</i>	
For Information About:	Refer to page:
DFS client	34
FL server	34
Backup server	34

Version 2 Release 10 Overview

This section describes the new and changed OS/390 Distributed File Service functions that are introduced for OS/390 Version 2 Release 10 (V2R10).

NT LM 0.12 (includes NT LM 0.12 encrypted password)

Description	The OS/390 DFS/SMB server supports the LM 0.12 level of the SMB protocol used by Windows NT networking, thereby providing additional password encryption.
What This Change Affects	Distributed File Service SMB
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service SMB Administration Guide and Reference</i>

RFS SMB (SMB support for MVS data sets)

Description	The OS/390 DFS/SMB server supports workstation access to OS/390 data stored in SAM, PDS(E), and VSAM files to further expand the S/390 support for application development.
What This Change Affects	Distributed File Service SMB
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service SMB Administration Guide and Reference</i>

Greater than 32 bit Episode files

Description	This supports file sizes greater than 32 bits (4 gigabytes).
What This Change Affects	Distributed File Service SMB
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service SMB Administration Guide and Reference</i>

Version 2 Release 9 Overview

This section describes the new and changed OS/390 Distributed File Service functions that are introduced for OS/390 Version 2 Release 9 (V2R9).

Encrypted Password Login

Description	The OS/390 SMB server supports encrypted password login. When this function is used, the SMB server invokes OCSF services. PC users are allowed to store their SMB password in RACF for the encrypted password function.
What This Change Affects	Distributed File Service SMB Note: This support was also made available on OS/390 Release 7 and Release 8 (May 2000).
Dependencies	OCSF must be installed
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service SMB Administration Guide and Reference</i>

Authorized Programs

The list of authorized programs for the V2R9 Distributed File Service has been expanded and includes:

- IOEGRWAG
- IOENEWAG
- IOESALVG
- SMBPW

Refer to the *OS/390 Program Directory* for a description of the PARMLIB member updates required for the member IJKTSOxx.

Version 2 Release 8 Overview

This section describes the new and changed OS/390 Distributed File Service functions that are introduced for OS/390 Version 2 Release 8 (V2R8).

SMB File/Print Server Support

Description	SMB (Server Message Block) support is based on X/Open CAE Specification Protocols for X/Open PC Interworking: SMB, Version 2. The SMB File/Print Server processes requests from OS/2 Version 4, Windows 3.11 (Windows for Workgroups), Windows 95, Windows 98, or Windows NT 4.0 workstation users against a shared directory or shared printer on OS/390. The SMB File Serving support, only, is also available for OS/2 clients. The SMB requests are sent to and received from the OS/390 SMB server a TCP/IP connection. Shared directories (file shares) are defined to identify HFS data sets exported by the SMB File/Print Server. Shared printers are assigned to JES print queues and printer names defined to the OS/390 Infoprint Server. The JES queues are written to or queried by the SMB File/Print Server in response to SMB point requests. In turn, the Infoprint Server reads and processes print requests from the JES queue.
What This Change Affects	Distributed File Service SMB Note: The SMB File Serving support, only, was made available on OS/390 Release 7 (December 1999).
Dependencies	For SMB Print Serving you need the Infoprint Server installed.
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See "Steps for Migrating" on page 18.
For More Information	<i>OS/390 Distributed File Service SMB Administration Guide and Reference</i>

Authorized Programs

The list of authorized programs for the V2R8 Distributed File Service has been expanded and includes:

- IOEBAK
- IOEBOS
- IOECM
- IOEDCERR
- IOEDFSXP
- IOEFTS
- IOEGRWAG
- IOEMAPID
- IOENEWAG
- IOESALVG
- IOESCOU
- IOEUDBG.

Refer to the *OS/390 Program Directory* for a description of the PARMLIB member updates required for the member IJKTSOxx.

Version 2 Release 7 Overview

This section describes the new and changed OS/390 Distributed File Service functions that are introduced for OS/390 Version 2 Release 7 (V2R7).

DFS Server Multi-home Support

Description	The Distributed File Service currently assumes Distributed File Service servers are reachable by a single network interface. The OS/390 client will now recognize and exploit multiple network interfaces for a DFS server host. Multi-homed server capabilities allow administrators to specify up to four interfaces (either hostnames or IP addresses) in the FLDB for each File Server and FLDB machine.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

Protected RPC

Description	Distributed File Service allows you to set RPC authentication levels for Cache Manager to File Server communications. These levels can be set individually for each Cache Manager and File Server. In addition, you can also set advisory RPC authentication bounds on a per-fileset basis.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

OSF/Transarc compatibility

Description	OS/390 supports DCE Local File System aggregate sizes that are greater than 4GB. The actual maximum size is based on the aggregate blocksize. Aggregates can contain 2G blocks. For a 4K blocksize, the aggregate can be 8TB (4K x 2G). For a 64K blocksize, the aggregate can be 128TB (64K x 2G).
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.

I	Migration Tasks	See “Steps for Migrating” on page 18.
I	For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

Version 2 Release 6 Overview

This section describes the new and changed OS/390 Distributed File Service functions that are introduced for OS/390 Version 2 Release 6 (V2R6).

DFSKERN in a Separate Address Space

Description	This allows the DFSKERN server process to be run in a separate address space. This enables address space termination processing to free file and other resources held by DFSKERN.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

DFS Backup

Description	This eliminates tape data file open/close overhead by backing up multiple file sets to a single tape data set.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

File basis translation

Description	This allows the OS/390 DFS client to access file sets that include both binary and text files. File translation specifications will be supported on a file type (extension) basis.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

Version 2 Release 5 Overview

This section describes the new and changed OS/390 Distributed File Service functions that are introduced for OS/390 Version 2 Release 5 (V2R5).

AIX compatibility

Description	This upgrades OS/390 Distributed File Service R5 code to the same level as the latest available AIX code. The additional function includes limiting the use of DFS client cache by a user to a fair percentage of the available cache when the user is sequentially reading a large file.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

DFS RAS/NLS message support

Description	This includes enhancements related to RAS, NLS, and general improvements to DFS message generation. It also improves overall DCE Distributed File Service RAS to facilitate the development of new distributed applications on S/390 that use DFS.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

OSF DCE 1.2.1 enhancements

Description	This includes the port and/or implementation of OSF DCE 1.2.1 code function, including applicable service and new functions such as, replication enhancements, bulk status RPC of DFS, and set server preference.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

| RFS enhancements

Description	This improves the functionality and performance for record file system access over the support provided by Record Data Access. It includes fast PDS and PDS/E member lockup, set attributes (setattr) improvements, open/close reduction, create/mkdir improvements, and physical and logical I/O improvements.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

Version 2 Release 4 Overview

There were no functional changes in this release.

Version 1 Release 3 Overview

This section describes the new and changed OS/390 Distributed File Service functions that are introduced for OS/390 Version 1 Release 3 (V1R3).

Record Data

Description	Record Data Access (RDA) provides access to the following OS/390 record files: VSAM, SAM, and PDS. This allows OS/390 record data available to DFS client applications. This data is presented as byte stream data.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

Version 1 Release 2 Overview

This section describes the new and changed OS/390 Distributed File Service functions that are introduced for OS/390 Version 1 Release 2 (V1R2).

DFS client

Description	This allows users local OS/390 access to data stored on remote DFS servers. The client support includes the addition of the Cache Manager. In DFS, the Cache Manager processes local file requests and communicates with other DFS server machines on the network. In addition, the Cache Manager maintains connections between DFS file and fileset location servers, sends file system requests to DFS file servers as remote procedure calls (RPCs), and maintains a local cache of recently accessed information.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

File Location Server (flserver)

Description	The flserver process maintains the Fileset Location Database, which tracks the location of all DCE local and non-Local File system filesets.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.
For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>

Backup Server (bakserver)

Description	The bakserver process communicates with the Backup Database to perform dump and restore operations.
What This Change Affects	DCE Distributed File Service
Dependencies	OS/390
Coexistence Considerations	DFS clients and servers interact with different platforms and with different OS/390 releases. It is assumed that the DFS clients and servers are running at the same release level.
Migration Tasks	See “Steps for Migrating” on page 18.

I	For More Information	<i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>
---	----------------------	---

Summary of Interface Changes

This section summarizes the new and changed interface components of OS/390 Distributed File Service.

For Information About:	Refer to page:
Distributed File Service Files	36
Distributed File Service Commands	36
Distributed File Service APIs	37
SMB Files	37
SMB Commands	37
Environment Variables	37

Distributed File Service Files

Table 9 lists the new and updated Distributed File Service files. See the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information.

Table 9. Summary of New and Changed Distributed File Service Files

File Name	Release	Description
Attributes File	V1R3	Contains tables describing the attributes used to manipulate record files in the DFS Server.
cm attributes file (cmattr)	V2R6	Contains directives that map a file name extension (suffix) to an indication whether the OS/390 DFS Cache Manager should translate the file data from ASCII to EBCDIC and vice versa (encoding).
CMLog	V2R5	Contains messages generated by the Cache Manager.
devtab	V1R3	Stores identifying information for all DCE Local File System logical volumes and non-Local File System devices to be exported.
dfstab	V1R3	Specifies DCE Local File System aggregates and non-Local File System partitions that can be exported.

Distributed File Service Commands

Table 10 lists the new and updated Distributed File Service commands. See the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information.

Table 10 (Page 1 of 2). Summary of New and Changed Distributed File Service Commands

Command Name	Release	Description
cm getpreferences	V2R5	Displays the Cache Manager preferences for File Server machines.
cm getprotectlevels	V2R7	Returns the current DCE RPC authentication level settings for communications between the Cache Manager and File Servers.
cm setpreferences	V2R5	Sets the Cache Manager's preferences for File Server machines.
cm setprotectlevels	V2R7	Adjusts DCE remote procedure call (RPC) authentication levels for communications between the Cache Manager and File Servers.
dfsd	V2R5	Initializes the DFS Cache Manager and starts related daemons.

Table 10 (Page 2 of 2). Summary of New and Changed Distributed File Service Commands

Command Name	Release	Description
fts release	V2R5	Initiates Release Replication by placing a read-only version of a read-write DCE Local File System fileset at the local site.
fts setprotectlevels	V2R7	Set advisory DCE remote procedure call (RPC) authentication levels for a specified fileset.
fxd	V2R7	Initializes the File Exporter and starts associated kernel daemons.

Distributed File Service APIs

Table 11 lists the new and updated Distributed File Service application programming interfaces (APIs). See the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information.

Table 11. Summary of New and Changed Distributed File Service APIs

API Name	Release	Description
w_piocntl	V2R5	Retrieves the contents of a DFS mount point.

SMB Files

Table 12 lists the new and updated SMB files. See the *OS/390 Distributed File Service SMB Administration Guide and Reference* for more information.

Table 12. Summary of New and Changed SMB Files

File Name	Release	Description
smbidmap	V2R8	Maps an SMB user ID to an OS/390 user ID. The mapping is used to determine the SMB user's corresponding OS/390 user ID. This determines access permissions for shared HFS directories and files and owners for print requests sent to shared printers.
smbtab	V2R8	Specifies HFS shared directories and shared printers to be made available to PC clients.

SMB Commands

Table 13 lists the new and updated SMB commands. See the *OS/390 Distributed File Service SMB Administration Guide and Reference* for more information.

Table 13. Summary of New and Changed SMB Commands

Command Name	Release	Description
dfsshare	V2R8	An OMVS command that shares (or unshares) HFS directories or printers with SMB clients.

Environment Variables

Table 14 on page 38 lists new and changed environment variables.

Table 14 (Page 1 of 3). Summary of New and Changed Environment Variables

Environment Variable Name	Release	Description
_EUV_SVC_MSG_LEVEL	V2R5	A DCE environment variable that sets the minimum severity level of messages that are displayed.
_EUV_SVC_MSG_LOGGING	V2R5	A DCE environment variable used to control where messages are routed and displayed.
_IOE_CFG_CELL_ID	V1R3	Specifies the DCE principal name of the administrator who is performing the OS/390 DFS configuration.
_IOE_CFG_CELL_NAME	V1R3	Specifies the name of the DCE cell, without the leading /.../.
_IOE_CFG_INFORM_LEVEL	V1R3	Specifies the level of information that is displayed on the screen from the configuration of the host system.
_IOE_CFG_LOG_FILE	V1R3	Specifies the name of the configuration log file.
_IOE_BACKUP_TAPE_CAPACITY	V2R6	Specifies the size, in bytes, of the maximum size backup tape data set that is created during backup. It is desirable that this value equal the actual capacity of your tape volumes (or a little less).
_IOE_CM_ATTRIBUTES_FILE	V2R6	Specifies the name of the file that contains the definition of file name suffixes that controls whether file data should be translated by the DFS client.
_IOE_CM_DIRCACHE_SIZE	V2R6	Specifies the size, in bytes, of the directory cache for the DFS client.
_IOE_CM_PARMS	V2R5	
_IOE_DAEMONS_IN_AS	V2R6	The DFS environment variable controls the number DFSCM ioedfsd process threads set up to handle file requests.
_IOE_DIRECTORY_CACHE_SIZE	V2R8	The number of 512 byte blocks used to cache HFS and RFS directory entries.
_IOE_EPI_CACHE_SIZE	V2R6	The size, in bytes, of the DCE Local File System metadata buffer cache. This environment variable is used to increase the size of the buffer cache and to potentially improve performance.
_IOE_HFS_ATTRIBUTES_FILE	V2R7	Specifies the pathname of the hfsattr file that contains the definition of file name suffixes that controls whether file data should be translated by the DFS server.
_IOE_HFS_TRANSLATION	V2R7	A DFS server variable that controls the conversion of HFS data to the appropriate data format. Converts incoming data from ASCII ISO 8859-1 to the local OS/390 code page. Converts outgoing data from the local OS/390 code page to ASCII ISO 8859-1.
_IOE_LFS_SYNC_INTERVAL	V2R8	The number of seconds between synchronization operations for the DCE Local File System.
_IOE_MVS_DFSDFLT	V2R6	The name of the RACF-defined anonymous user that is associated with unauthenticated DCE users attempting access to exported HFS files. This ID must be a RACF-defined with an OS/390 UNIX segment.
_IOE_MVS_SERVER	V2R6	The name of the OS/390 server used for identity mapping and registering with OS/390 UNIX as a DFS server.
_IOE_PROTOCOL_RPC	V2R8	An environment variable that controls whether the DCE DFS protocol is supported (using DCE RPC).

Table 14 (Page 2 of 3). Summary of New and Changed Environment Variables

Environment Variable Name	Release	Description
_IOE_PROTOCOL_SMB	V2R8	An environment variable that controls whether the SMB protocol is supported (using TCP/IP).
_IOE_RFS_ALLOC_TIMEOUT	V2R10	Specifies the time period (in seconds) that an RFS data set will remain allocated after there has been no access to the data set through the DFS/SMB server. After this time period, the data set will be deallocated and will be available to other applications such as ISPF.
_IOE_RFS_ATTRIBUTES_FILE	V1R3	The name of the file that contains the table for describing the attributes used to manipulate record files in the DFS Server.
_IOE_RFS_STATUS_REFRESH_TIME	V1R3	The DFS Server refreshes its cache of exported record data set names and attributes at a regular interval.
_IOE_RFS_TRANSLATION	V1R3	A DFS Server variable that controls the conversion of RFS data to the appropriate data format.
_IOE_RFS_WORKER_THREADS	V1R3	A DFS Server variable that controls the number of threads to be started in dfskern to service open and close for RFS files.
_IOE_SERVER_NAME	V2R5	The network name of the system. This environment variable should be used if there are several network connections on the OS/390 system and the primary connection is not being used for the DFS system.
_IOE_SMB_BROWSE_INTERVAL	V2R8	Specifies the Browser announcement interval (in milliseconds).
_IOE_SMB_CALLBACK_POOL	V2R8	Specifies the number of secondary pool threads for processing SMB callback requests.
_IOE_SMB_CLEAR_PW	V2R9	Specifies the OS/390 SMB server policy for flowing the SMB password in the clear.
_IOE_SMB_COMPUTER_NAME	V2R8	Specifies the name to be used by SMB redirectors (that is, clients) to contact this server.
_IOE_SMB_DESCRIPTION	V2R8	Specifies the description of this server that appears on the PC.
_IOE_SMB_DIR_PERMS	V2R9	The permissions for a directory created by the SMB server at the request of an SMB client.
_IOE_SMB_DOMAIN_NAME	V2R8	Specifies the name to be used as the Domain name for this server.
_IOE_SMB_FILE_PERMS	V2R9	The permissions for a file created by the SMB server at the request of an SMB client.
_IOE_SMB_IDMAP	V2R8	Specifies the location of the smbidmap file. The smbidmap file contains the mapping of SMB user IDs to OS/390 user IDs.
_IOE_SMB_IDLE_TIMEOUT	V2R8	Specifies how long (in seconds) an SMB session can remain inactive before it is terminated.
_IOE_SMB_MAIN_POOL	V2R8	Specifies the number of primary pool threads for remain inactive before it is terminated.
_IOE_SMB_MAXXMT	V2R8	Specifies the maximum server buffer size that is returned on the SMB Server Negotiate response.

Table 14 (Page 3 of 3). Summary of New and Changed Environment Variables

Environment Variable Name	Release	Description
_IOE_SBM_NT_SMBS	V2R10	Specifies whether new SMBs in the NT protocol dialect are to be accepted from PC clients.
_IOE_SMB_OPLOCK_TIMEOUT	V2R8	Specifies the Opportunistic Lock Timeout period (in seconds).
_IOE_SMB_OPLOCKS	V2R9	Specifies whether the server allows clients to use Opportunistic Locks on files. This allows some SMB clients to cache data at the client.
_IOE_SMB_PRIMARY_WINS	V2R8	Specifies the IP address of the Windows Internet Service (WINS) server that this server announces itself to and forwards WINS requests to.
_IOE_SMB_PROTOCOL_LEVEL	V2R10	Specifies the highest level of SMB protocol dialect that will be negotiated with the PC client.
_IOE_SMB_RAW	V2R8	Specifies whether raw mode is supported in the SMB Server Negotiate response.
_IOE_SMB_SCOPE	V2R8	Specifies the Scope ID of the Windows Internet Naming Service (WINS) server.
_IOE_SMB_SECONDARY_WINS	V2R8	Specifies the IP address of the Windows Internet Naming Service (WINS) server that this server announces itself to and forwards WINS requests to if the Primary WINS server does not respond.
_IOE_SMB_TOKEN_FILE_MAX	V2R8	Specifies the maximum number of files that the SMB token cache should keep tokens for.
_IOE_SMB_WINS_PROXY	V2R8	Specifies whether this server can act as a Windows Internet Naming Service (WINS) server proxy.
_IOE_TKM_MAX_TOKENS	V2R6	Specifies the number of tokens that can be held in the DFS server memory.
_IOE_VM_CACHE_SIZE	V2R5	Specifies the size, in bytes, of the virtual memory used by the DFS client and DFSKERN for the DCE Local File System.
_IOE_VM_MAX_FILES	V2R5	Specifies the maximum number of files that can be contained in the virtual memory used by the DFS client and DFSKERN for the DCE Local File System.

Chapter 6. Post Installation Considerations

Before OS/390 DFS can be used, you must create the various DFS configuration files. These files are usually created before the DFS installation is verified by the **dfs_cpfiles** shell script program.

You also need to perform administrative actions such as, identifying all the DFS processes you wish to start; specifying DFS process options; defining DFS users and administrators and file data to export for access by the DFS server running on your host system. This chapter contains the following information, in the order that it should be performed, to assist and guide you in completing the installation and configuration of DFS.

- “DFS Installation and Configuration Overview”
- “Using dfs_cpfiles to Create Configuration Files (dfscpfil)”
- “DFS Configuration Considerations” on page 43
- “DFS Server Considerations” on page 43
- “DFS Client (DFSCM) Considerations” on page 45.

DFS Installation and Configuration Overview

To install and configure DFS for operation, you must perform the following administrative actions:

1. Install DFS using the applicable instructions in the *OS/390 Program Directory*.
2. If the DFS Client runs on the host system, and if it's not already done during the OS/390 installation, add the entry **FILESYSTYPE TYPE(DFSC)** in the system parmlib member **BPXPRMxx** as described in “System Parmlib Member BPXPRMxx Entry for the DFSCM” on page 72.
3. Create the default DFS configuration files using the **dfs_cpfiles** if they were not created during the DFS installation process. Refer to “Using dfs_cpfiles to Create Configuration Files (dfscpfil)” for more information.
4. Configure and customize either the DFS Client or DFS Servers or both on the host system following the applicable instructions in Chapter 7, “Configuring DFS” on page 47.
5. Define DFS administrators and users on the host system following the applicable instructions in Chapter 8, “Defining DFS Administrators and Users” on page 83.
6. If DFS Servers are run on the host system, define aggregates and filesets following the applicable instructions in Chapter 9, “Exporting Data in DFS” on page 101.
7. Start DFS following the applicable instructions in Chapter 11, “Starting and Stopping DFS Components” on page 131.

Using dfs_cpfiles to Create Configuration Files (dfscpfil)

| The **/opt/dfsglobal/scripts/dfs_cpfiles** program is an OS/390 shell script that creates customizable configuration files in **/opt/dfslocal** subdirectories. **dfs_cpfiles** copies IBM-supplied files from the **/opt/dfsglobal/examples** directory to **/opt/dfslocal** subdirectories. The **dfs_cpfiles** program will create files that do not exist. It will not replace an existing file to preserve any installation configuration data from a previous release.

| **/opt/dfslocal** is a symbolic link to **/etc/dfs**. So, all the files created by the **dfs_cpfiles** program are actually created in the **/etc** file system. Refer to Appendix B, “Directories and Files” on page 147 for more information on the symbolic links defined to identify the configuration files.

| Although some configuration files can be created in other directories and identified by **envar** specifications,
| it is recommended that all the configuration files reside in the **/opt/dfslocal** subdirectories in the **/etc** file
| system.

To invoke **dfs_cpfiles**:

1. Log in as **root** on the local machine. In OS/390 DFS, this means as a user with a **UID = 0**.
2. While in the OS/390 shell environment, invoke the OS/390 DFS Default Configuration Files Creation Program by entering the following:

\$ dfs_cpfiles

3. The following example output shows the output after using **dfs_cpfiles** to create all the customizable DFS configuration files:

```
| *****
| ** Create OS/390 Distributed File Service (DFS/SMB) **
| ** Configuration Files **
| *****
|
| Create envar Files....
|   File /opt/dfslocal/home/bakserver/envar created
|   File /opt/dfslocal/home/boserver/envar created
|   File /opt/dfslocal/home/butc01/envar created
|   File /opt/dfslocal/home/butc02/envar created
|   File /opt/dfslocal/home/butc03/envar created
|   File /opt/dfslocal/home/butc04/envar created
|   File /opt/dfslocal/home/butc05/envar created
|   File /opt/dfslocal/home/butc06/envar created
|   File /opt/dfslocal/home/butc07/envar created
|   File /opt/dfslocal/home/butc08/envar created
|   File /opt/dfslocal/home/daemonct/envar created
|   File /opt/dfslocal/home/dfscm/envar created
|   File /opt/dfslocal/home/dfscntl/envar created
|   File /opt/dfslocal/home/dfsexport/envar created
|   File /opt/dfslocal/home/dfskern/envar created
|   File /opt/dfslocal/home/flserver/envar created
|   File /opt/dfslocal/home/ftserver/envar created
|   File /opt/dfslocal/home/growaggr/envar created
|   File /opt/dfslocal/home/newaggr/envar created
|   File /opt/dfslocal/home/repserver/envar created
|   File /opt/dfslocal/home/salvage/envar created
|   File /opt/dfslocal/home/upclient/envar created
|   File /opt/dfslocal/home/upserver/envar created
|
| Create Miscellaneous Configuration Files....
|   File /opt/dfslocal/etc/ioepdcf created
|   File /opt/dcelocal/etc/Cacheinfo created
|   File /opt/dfslocal/var/dfs/devtab created
|   File /opt/dfslocal/var/dfs/dfstab created
|   File /opt/dfslocal/var/dfs/rfstab created
|   File /opt/dfslocal/var/dfs/smbtab created
|   File /opt/dfslocal/var/dfs/cmattr created
|   File /opt/dfslocal/var/dfs/hfsattr created
|   File /opt/dfslocal/home/dfskern/dfsimap created
|   File /opt/dfslocal/home/dfskern/smbidmap created
```

Notes:

1. The previous example shows the **dfs_cpfiles** messages when all files are created.
If a file already exists, an example of this message is:
File /opt/dfslocal/etc/ioepdcf already exists
If an error occurs creating the file, an example of this message is:
File /opt/dfslocal/etc/ioepdcf not created
2. If you are migrating to this release of OS/390 DFS from an earlier release of DFS and **dfs_cpfiles** created new customizable configuration files, you may need to add any previous user customization data to the newly created files.
3. If you are migrating to this release of OS/390 DFS from an earlier release of DFS and new customizable configuration files were not created by **dfs_cpfiles**, you may want to update pre-existing customizable files with any new customization options available with this release of DFS. Refer to Chapter 5, “Migration Overview” on page 17 for more information on what is new in this release.
4. The files created by the **dfs_cpfiles** program contain only sample entries or comments. The content of these files can be replaced with installation information.
5. The DFS user identity mapping file is identified by the envvar **_IOE_MVS_IDMAP**. It is recommended that the **/opt/dfslocal/home/dfskern/dfsidmap** file created by the **dfs_cpfiles** program is used by the installation for user identity mapping.
6. Refer to Chapter 7, “Configuring DFS” on page 47 for more information on how to customize these files.

After all the configuration files required by this DFS release have been created, you can proceed to perform the additional DFS configuration steps outlined in “DFS Installation and Configuration Overview” on page 41.

DFS Configuration Considerations

Before using DFS on a host system, you must configure the host system into a DCE cell. Refer to the *OS/390 DCE Configuring and Getting Started* book for information on how to configure DCE.

After DCE is configured on the host system, before using DFS, you must configure either or both the DFS client and server using the same DCE cell name (*dcecellname*) and DCE host name (*dcehostname*) used for DCE configuration. Refer to Chapter 7, “Configuring DFS” on page 47 for more information.

DFS Server Considerations

Optionally, you can configure DFS servers on the host system.

All DFS servers do not need to be configured at this time but it is recommended that you configure those that may be needed at some time in the future. If you choose to configure any DFS servers to export file data to the DFS namespace from the DCE host system where DFS is being configured, minimally you must:

- Configure the DFS File Server which includes the following DFS server process:
 - DFS Control (**dfscntl**)
 - DFS Kernel (**dfskern**)
 - DFS Export (**export** and **unexport**)
 - DFS Basic Overseer Server (**boserver**).

Note: You must also configure a DFS Fileset Location Database (FLDB) if one does not already exist in the DCE cell.

- If you are a new DFS customer, the DFS server is installed and configured to run the IOEPDCT and DFSKERN processes in the DFSKERN address space. This is separate from the DFS server address space where the other DFS server processes will run.

If you had DFS configured prior to OS/390 Version 2 Release 6, you can now configure the DFS server to run the IOEPDCT and DFSKERN process in the separate DFSKERN address space. This is done by setting the environment variable, `_IOE_DAEMONS_IN_AS=DFSKERN`, in the file `/opt/dfslocal/home/dfscntl/envvar`.

It is recommended that the **dfskern** processes be run in an address space separate from the other DFS server processes to take full advantage of the system termination processing. Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for additional information about running **dfskern** and **ioepdct** in a separate address space.

- Configure a DFS Fileset Location Database Server (**flserver**) if an **flserver** does not already exist in the DCE cell.

You can configure one or more DFS Fileset (Location) Database Servers.

The Fileset Database Servers maintain the Fileset Location Database, the mechanism by which DFS clients can find out which File Server machine actually holds the files being requested.

One Fileset Location Server is required in each DCE cell using DFS. Multiple Fileset Location Servers in the cell provide availability and load balancing of the data. For most cells, three Fileset Location Server machines are recommended because three machines are usually sufficient for the tasks. If you configure more than three, an odd number is preferable.

- Complete the configuration of the DFS Fileset Server (**ftserver**).

Optionally, you can complete configuration for the following DFS server processes that run under control of the DFS Basic Overseer Server (**boserver**):

- You can optionally complete configuration for the DFS Backup processes which includes the DFS Backup Server process **bkservice** and from one to eight Backup Tape Coordinator **butc** processes. Note that the OS/390 Backup Server process is named **bkservice** but the Backup Server is identified as the **bkservice** after it is initially created and started.

A **bkservice** maintains the Backup Database, which is used for managing individual and periodic backups of DFS files. Backup Servers are optional (required only if you want to take advantage of the backup capabilities of DFS). Multiple Backup Servers provide for availability of the data. For DCE cells with DFS running on three or more host machines, three Backup Servers are recommended. If you configure more than three Backup Servers in a DCE cell, an odd number is preferable.

You can also set up and start one or more DFS Backup Tape Coordinators (**butc01** through **butc08**). A DFS Backup Tape Coordinator (**butc**) server actually performs the DFS backup and restore operations on the host system. A **butc** server requires a **bkservice** to be running in the DCE cell. But, a **bkservice** does not need to be running on the same OS/390 DCE host system where a **butc** server is running.

- You can optionally complete configuration for the DFS Replication Server process. Note that the OS/390 Replication Server process is named **rpserver** but the Replication Server is identified as the **repserver** after it is initially created and started.
- You can optionally complete configuration for the DFS Update Client (**upclient**) process. An **upclient** is required on a DCE host system if there are more than one DCE host system in a DCE cell running DFS and there is a DFS Update Server (**upserver**) running on another system.

- You can optionally complete configuration for the DFS Update Server (**upserver**) process. An **upserver** is usually required on a DCE host system if there is more than one DCE host system in a DCE cell running DFS and there is no DFS Update Server (**upserver**) running in the DCE cell and the other systems in the DCE cell will run the **upclient**.

You can refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for more information on DFS servers.

For more information on configuring DFS servers, see “Using DFSCONF to Configure DFS Servers” on page 57.

DFS Client (DFSCM) Considerations

Optionally, you can configure the **DFSCM** for use by OMVS users on the host system.

The DFS client in OS/390 DFS, **DFSCM**, also referred to as the Cache Manager, maintains contact with the File Exporter and other server processes. The Cache Manager translates file requests into Remote Procedure Calls (RPCs) to File Exporter processes on File Server machines. The Cache Manager receives and caches requested data from a File Exporter before passing the data to the application program that requested it. The Cache Manager always has access to the most current copy of the data.

For more information about the DFS Client, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

For more information about DFS Client configuration, see Chapter 7, “Configuring DFS” on page 47.

Chapter 7. Configuring DFS

The purpose of this chapter is to assist you in creating and configuring the Distributed File Service. OS/390 DFS provides an ISPF dialog to assist you. The use of these panels is explained in this chapter. For information regarding planning considerations you should consider prior to configuring your system, see Chapter 2, “Global and Cell Considerations” on page 5.

For information on setting up the intercell environment, managing intercell naming, and administering a multicell environment, see the *OS/390 DCE Administration Guide*.

This chapter provides information on the following topics:

- “Overview of DFS Configuration”
- “DFS Configuration Steps” on page 48
- “Using the DFS Configuration Program (DFSCONF)” on page 53
- “Using DFSCONF to Configure DFS Servers” on page 57
- “Using DFSCONF to Configure the DFS Client” on page 59
- “Completing the DFS File Server Configuration” on page 61
- “Completing the boserver Configuration” on page 61
- “Completing the FLDB Server (flserver) Configuration” on page 61
- “Creating the DFS File Server FLDB Entry” on page 63
- “Completing Fileset Server (ftserver) Configuration” on page 63
- “Completing Replication (repserver) Server Configuration” on page 65
- “Completing Update Client (upclient) Configuration” on page 65
- “Completing Update Server (upserver) Configuration” on page 66
- “Completing the Backup Database Server (bakserver) Configuration” on page 68
- “Completing the DFS Client (DFSCM) Configuration” on page 72
- “Customizing the ioepdcf File” on page 78
- “Customization of the DFS Server and the DFSCM envvar Files” on page 80.

Notes:

1. The DCE cell name (*dcecellname*) and DCE host name (*dcehostname*) used for DCE configuration must be the same *dcecellname* and *dcehostname* used to configure DFS.
2. DFS configuration command examples sometimes use */.* to denote */.../dcecellname*.

Overview of DFS Configuration

Configuring DFS in a DCE cell takes several steps. You should follow the steps in the order presented, although some steps allow you to configure multiple servers before moving on to the next. You can also come back later to configure another server of a given role.

For more information on DFS, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

DFS Configuration Steps

The DFS configuration steps reference detailed explanations in other sections of this publication. Once you complete a step and its supporting instructions, you should refer to the next step in this list.

1. Before Proceeding with DFS Configuration

Note: The following important information is required before proceeding with the DFS configuration steps.

a. Install DCE on the Host System

Insure that DCE is installed on the host system where DFS is configured, following the instructions in the *OS/390 Program Directory*, GI10-4001

b. Install DFS on the Host System

Insure that DFS is installed on the host system where DFS is configured, following the instructions in the *OS/390 Program Directory*.

c. Insure DCE is Running on the Host System

Insure that the host system where DFS will be configured is already configured into a DCE cell and that DCE is running on the host system. Refer to the *OS/390 DCE Configuring and Getting Started* book for more information.

Attention: If you configure a DCE cell with an X.500-style name and plan to run DFS, you must also configure GDS on the host system. If you do not, you cannot access the local cell's space.

d. Complete DFS Post Installation Processing

Insure that the steps described in Chapter 6, "Post Installation Considerations" on page 41 have been completed successfully.

e. Determine DCE Login Information

You must log in as **cell_admin** to perform DFS configuration so you will need to know the correct **cell_admin** login password before proceeding. Contact the person who initially configured DCE on the host system or the DCE administrator for this information.

f. Determine DCE Cell Name and DCE Host Name

You must also know the correct **dcecellname** and **dcehostname** defined during DCE configuration. After DCE is configured on the host system, this information is defined in the **/opt/dcelocal/dce_cf.db** file.

g. Add DFS Administrative Principals to DCE Registry

This step only applies if DFS servers will be run on the host OS/390 system where DFS is being configured.

If this is not already done, it is recommended that you register at least one user or group principal to be used in all DFS administrative lists. You can register more administrative users or groups at a later time. If you define a group for DFS administrative authorization, you can add users to the group now or more at a later time. It is recommended that the user or group you now select be used in all the DFS administrative lists. Later, different users or groups can be selectively added to the various DFS administrative list files. To register a user or group in a DCE cell, refer to the instructions in the *OS/390 DCE Configuring and Getting Started* book or in the *OS/390 DCE Administration Guide*.

h. Stop DFS and the DFS Client

The OS/390 installation verifying instructions call for starting the DFS client and server address spaces. It is recommended that you stop DFS processing on the host system during configuration processing.

If DFS (the DFS server) is already running, issue the following OS/390 system command:

```
stop dfs
```

If the DFS Client is already running, issue the following OS/390 system command:

```
stop dfscm
```

i. **Validate that DCE is Configured for DFS on the Host System**

You should validate that DCE is properly configured on the OS/390 host system where DFS configuration is performed.

From OMVS on the host system, you can issue the following commands to receive the output shown, if the DCE cell is properly configured for DFS.

```
dcecp -c show rpcentry /./subsys/dce/dfs/bak  
dcecp -c show rpcentry /./fs
```

If either command results in the message indicating that the name service object is not found, then the DCE cell is not configured for DFS. You must contact the DCE cell administrator to determine the status of the DCE cell (see the *OS/390 DCE Configuring and Getting Started* book or the *OS/390 DCE Administration Guide*).

2. **Set OS/390 shell Flag**

If you plan to configure the DFS Client (**DFSCM**) on the OS/390 host system, it is recommended that the UNIX System Services shell's logical flag be set using the **set -o logical** command. See the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for more information about displaying the current working directory from the shell with the **DFSCM** configured. Also, see the *OS/390 UNIX System Services Command Reference*, SC28-1892, for more information about the **set** command.

3. **Set Up to Use the DFSCONF Command**

Refer to “Using the DFS Configuration Program (DFSCONF)” on page 53, for setting up information necessary to use the **DFSCONF** command from TSO.

4. **Optionally Modify the DFS envvar Files**

Default DFS process **envvar** files are created by **dfs_cpfiles** during DFS post installation (see Chapter 6, “Post Installation Considerations” on page 41). You can use these default **envvar** files to configure DFS and customize them at a later time. Instructions on how to modify the DFS **envvar** files can be found in “Customization of the DFS Server and the DFSCM envvar Files” on page 80.

5. **Use DFSCONF to Configure the DFS Client (DFSCM)**

Optionally, you can configure the **DFSCM** for use by OMVS users on the host system (see “DFS Client (DFSCM) Considerations” on page 45). If you do not want to configure the **DFSCM** on the host system at this time, proceed to step 6 on page 50.

In OS/390 DFS, the **DFSCM** does not require any of the DFS servers to be configured and running on the OS/390 system on which it is configured and running. But, if you want to configure the DFS client and DFS servers on the host system, it is recommended that you configure both at this time using **DFSCONF**.

For information on configuring a DFS client, see “Using DFSCONF to Configure the DFS Client” on page 59.

If you choose to also configure DFS servers using **DFSCONF** at this time, see

- “Using DFSCONF to Configure the DFS File Server” on page 58

- “Using DFSCONF to Configure the DFS Backup Server” on page 58
- “Using DFSCONF to Configure the DFS Fileset Location Database Server” on page 59.

Note: If no DFS servers will be configured on the host system, after configuring the DFS client using **DFSCONF**, you can now refer to step 10 on page 51, **Complete DFSCM Configuration**, and skip the DFS server configuration steps.

6. Use DFSCONF to Configure OS/390 DFS Servers

Optionally, you can configure DFS servers on the host system (see “DFS Server Considerations” on page 43).

It is recommended that you configure all DFS servers you initially want on the host system at this time using **DFSCONF** if not already done previously. For detailed information, see

- “Using DFSCONF to Configure the DFS File Server” on page 58
- “Using DFSCONF to Configure the DFS Backup Server” on page 58
- “Using DFSCONF to Configure the DFS Fileset Location Database Server” on page 59.

7. Modify the ioepdcf File and Complete boserver Configuration

This step only applies to DFS server configuration.

The default **/opt/dfslocal/etc/ioepdcf** file created during DFS post installation processing by **dfs_cpfiles** (see Chapter 6, “Post Installation Considerations” on page 41) is set up to automatically start only **dfskern** when the DFS server address space is next initialized. It is recommended that you update the **ioepdcf** file to automatically start the **boserver**. This is the recommended setting for subsequent DFS server configuration processing. If you want to change the default **ioepdcf** file settings, refer to “Customizing the ioepdcf File” on page 78 for more information.

8. Start the DFS Server

This step only applies to DFS server configuration.

Initialize the DFS server address space by issuing the following OS/390 system command:

```
start dfs
```

Notes:

- Only the required DFS server processes **dfskern** and **boserver** will be started if the recommended **/opt/dfslocal/etc/ioepdcf** settings are used (see step 7).
- Starting the **boserver** creates empty **/opt/dcelocal/var/dfs/admin.bos** and **/opt/dcelocal/var/dfs/BosConfig** files, if those files do not already exist.
- DFS should be running with DFS authorization disabled by the existence of the file **/opt/dcelocal/var/dfs/NoAuth**.

9. Create Initial DFS Administrative Authorization Lists

This step only applies to DFS server configuration.

You can now define the DFS administrators for the DFS servers you will run on the DCE host system by updating the DFS administrative list file for each server.

At this time, it is recommended that you only update all the DFS administrative list files on this DCE host system with a single user ID that has **cell_admin** authority. Note, the DCE **cell_admin** user ID inherits no special DFS authorization and has to be explicitly specified in a DFS administrative list file.

In case you are configuring this DCE host system into an existing DCE cell and another system is acting as a DFS System Control Machine (where the **upserver** is running), the user ID that is specified in the DFS administrative list file on this DCE host should also be one that is specified in the DFS administrative list file on the DFS System Control Machine. Later, you will be instructed when to

update the DFS administrative list files with additional users and user groups and to perform updates of the DFS administrative list files on the DCE host system where the **upserver** is running.

Refer to “Creating the Initial DFS Administrator(s)” on page 87, “Defining a DFS Administrator” on page 84 and “Adding Principals and Groups to Administrative Lists” on page 89 for further information.

10. Complete DFS Client Configuration

If the DFS Client (**DFSCM**) was configured on the host system using **DFSCONF**, you must complete configuring and start the **DFSCM** as described in “Completing the DFS Client (DFSCM) Configuration” on page 72.

11. Complete flserver Configuration

There must be a Fileset Location Database Server (**flserver**) running in the DCE cell. If there is no **flserver** running in the DCE cell or if you want an **flserver** to also run on this DCE host system, follow the instructions for “Completing the FLDB Server (flserver) Configuration” on page 61.

12. Create FLDB Entry for DFS File Server

After an **flserver** is running in the DCE cell (on this host or another host in the DCE cell), you must follow the instructions in “Creating the DFS File Server FLDB Entry” on page 63.

13. Complete ftserver Configuration

If you have configured the DFS File Server using **DFSCONF**, you must configure a DFS Fileset Server (**ftserver**) if you plan to export any file data from this DCE host system. See “Completing Fileset Server (ftserver) Configuration” on page 63 for more information.

14. Complete repserver Configuration

If you have configured the DFS File Server using **DFSCONF** and if you plan to run a DFS Replication Server (**repserver**) on this DCE host system, you must follow the instructions in “Completing Replication (repserver) Server Configuration” on page 65.

Note: If you plan to define a **root.dfs** fileset for the DCE cell on this DCE host system, you must configure a **repserver** on this DCE host.

15. Complete upclient Configuration

If you have configured the DFS File Server using **DFSCONF** and if you plan to run a DFS Update Client (**upclient**) on this DCE host system, you must follow the instructions in “Completing Update Client (upclient) Configuration” on page 65.

Note: If an **upserver** is already defined in the DCE cell, you should configure an **upclient** on this DCE host system. Otherwise, you should not configure an **upclient** and configure an **upserver** on this DCE host system as described in step 16.

16. Complete upserver Configuration

If you have configured the DFS File Server using **DFSCONF** and if you plan to run a DFS Update Server (**upserver**) on this DCE host system, you must follow the instructions in “Completing Update Server (upserver) Configuration” on page 66.

Note: If an **upserver** is not already defined in the DCE cell, you should configure an **upserver** on this DCE host system if you plan to have DFS running on more than one DCE host system in the DCE cell.

17. Complete bakserver Configuration

If you have configured the DFS Backup Server using **DFSCONF** and if you plan to run a DFS Backup Server (**bakserver**) on this DCE host system, you must follow the instructions in “Completing the Backup Database Server (bakserver) Configuration” on page 68.

Note: If you do not plan to use the DFS backup facility or if a **bakserver** is already defined in the DCE cell, you do not need to define one on this DCE host system.

18. Complete butc Configuration

If you have configured the DFS Backup Server using **DFSCONF** and if you plan to run at least one DFS Backup Tape Coordinator (**butc**) daemon on this DCE host system, you must follow the instructions in “Completing the DFS Backup Tape Coordinator Configuration” on page 69.

Note: If you want to use the DFS backup facility for DCE Local File System data stored on OS/390, you should configure at least one **butc** server on the host system.

19. Customize the ioepdcf File

It is recommended that you can now customize the **ioepdcf** file to indicate that all servers you have configured should be started automatically when the DFS address space is started. (see “Customizing the ioepdcf File” on page 78).

20. Restart DFS and Start DFS Client

It is recommended that you stop and restart DFS processing at this time. For detailed information, see Chapter 11, “Starting and Stopping DFS Components” on page 131.

- a. To stop DFS (all DFS servers running in the DFS address space), issue the following OS/390 system command:

stop dfs

- b. If the DFS Client is already running, issue the following OS/390 system command:

stop dfscm

- c. Issue the following OS/390 system command to start the DFS address space:

start dfs

- d. Restart the DFS Client following the instructions in “Stopping and Restarting the DFSCM” on page 133.

21. Create Additional DFS Administrators

This step only applies to DFS server configuration.

You can now define additional DFS administrators for the DFS servers you will run on the DCE host system by updating the DFS administrative list file for each server.

If this DCE host system was added to an existing DCE cell, and there is an **upclient** running on this system, you should update the DFS administrative list files on the DCE host machine where the **upserver** is running. The **upclient** and **upserver** processing will propagate the DFS administrative list file updates back to administrative list files located on this DCE host machine.

Refer to “Defining a DFS Administrator” on page 84 and “Adding Principals and Groups to Administrative Lists” on page 89, for further information.

22. Define a root.dfs Fileset for the DCE Cell

This step applies to configuring DFS servers on OS/390.

Each DCE cell requires a **root.dfs** fileset for DFS. If a **root.dfs** fileset does not already exist for the DCE cell into which the OS/390 host system was configured, one must be created.

The **ftserver** and **rpserver** must be configured and running on the OS/390 host system before proceeding to the instructions in Chapter 9, “Exporting Data in DFS” on page 101.

23. Create and Export a DCE Local File System Aggregate

If you want to export file data for general use that is stored in a DCE Local File System aggregate on the OS/390 host, refer to Chapter 9, “Exporting Data in DFS” on page 101.

24. Setting Up OS/390 User IDs for HFS and RFS Access

If you will run the DFS File Server on an OS/390 system and plan to export HFS or RFS data sets, see “Mapping OS/390 User IDs for HFS and RFS Access” on page 92.

25. Create and Export General Use HFS Aggregate

If you want to export file data for general use that is stored in an HFS aggregate on the OS/390 host, refer to Chapter 9, “Exporting Data in DFS” on page 101.

26. Create and Export General Use RFS Aggregate

If you want to export file data for general use that is stored in an RFS aggregate on the OS/390 host, refer to Chapter 9, “Exporting Data in DFS” on page 101.

27. Customize the DFS envar Files

Default **envar** files are created by **dfs_cpfiles** during post installation processing if they did not exist (see Chapter 6, “Post Installation Considerations” on page 41). At this time, you may want to follow the instructions in “Customization of the DFS Server and the DFSCM envar Files” on page 80.

Using the DFS Configuration Program (DFSCONF)

Use **DFSCONF** from TSO to configure the OS/390 DFS servers and the DFS client. The **DFSCONF** configuration program has interactive panels that prompt and guide you through the details of configuring the OS/390 DFS servers and the DFS client.

The OS/390 DFS Server Configuration program automates the DCE configuration (and deconfiguration) of the DFS servers and creation of symbolic links required by the DFS Client.

DFSCONF uses the DCE administrative facilities to enter configuration information in the Security Registry and the CDS namespace. These facilities are the Registry Editor, ACL Editor, RPC Control Program, CDS Control Program, and the DCE Control Program. You must be properly authenticated and authorized by the DCE Security Service.

In the **DFSCONF** panels, you will either be prompted to select from a list of options or to enter a value. To select from a list of options, enter the number corresponding to the desired function in the selection field. For panels that require values, enter the appropriate response in the input field.

Details on using these panels are described in this chapter.

Note: If you have previously configured DFS on this DCE host system, running **DFSCONF** does not alter any DFS administrative file that may exist in the **/opt/dcelocal/var/dfs** directory.

DCE Messaging Subsystem Environment Variables for DFSCONF

The **DFSCONF** configuration program uses the DCE messaging facility. The DCE messaging facility is controlled by the environment variables **_EUV_SVC_MSG_LOGGING** and **_EUV_SVC_MSG_LEVEL**.

It is recommended that when you run **DFSCONF**, **_EUV_SVC_MSG_LOGGING** should be set to **CONSOLE_LOGGING** and **_EUV_SVC_MSG_LEVEL** should be set to **VERBOSE** (the default value).

Refer to “Setting **DFSCONF** Environment Variables” on page 55 for more information. For more information on specifying the DCE messaging facility environment variables refer to the *OS/390 DCE Administration Guide*.

Configuration Log File for DFSCONF

The **DFSCONF** program appends messages to the **dfsconf.log** file that detail the steps that it performs in configuring or deconfiguring DFS on the host system. If the **dfsconf.log** file does not exist, it is created by **DFSCONF**.

By default, the **dfsconf.log** file resides in the home directory of the user running **DFSCONF**. The environment variable **_IOE_CFG_LOG_FILE** can be used to control the location of the **dfsconf.log** file. If the default is not used, be sure that the user running **DFSCONF** has write access to the specified **dfsconf.log** file.

The environment variable, **_IOE_CFG_INFORM_LEVEL** should be set to **2** during DFS configuration. This setting causes each DFS configuration or deconfiguration to generate approximately 7K bytes of message text. Insure that the **dfsconf.log** file resides in the file system that has sufficient free space to contain the messages.

Refer to “DFSCONF Environment Variables” for more information. Refer to Appendix A, “Example DFS Server Configuration Log File (dfsconf.log)” on page 141 for an example of the content of a **dfsconf.log** file.

DFSCONF Environment Variables

The **DFSCONF** configuration program uses environment variables to set the default values for host information. The environment variables are declared in the **envar** file in the home directory of the administrator running **DFSCONF**. If set, the values of these environment variables are displayed as default values in the DFS configuration panels. If the environment variables are not set, the appropriate information must be entered on the **DFSCONF** configuration panels.

Following is a brief description of the environment variables that apply when running the **DFSCONF** program:

Variable	Description
_EUV_SVC_MSG_LOGGING	Defines where DCE messages are written. The default value CONSOLE_LOGGING is used if this is not specified.
_EUV_SVC_MSG_LEVEL	Controls the types of DCE messages generated. The default value VERBOSE is used if this is not specified.
_IOE_CFG_INFORM_LEVEL	Controls the types of DFS messages generated. The default value 1 is used if this is not specified. The value 2 is recommended when running DFSCONF .
_IOE_CFG_LOG_FILE	Defines the name of the DFS configuration log file where DFS messages are written. The default value dfsconf.log is used if this variable value is not specified. By default, the file is located in the home directory of the administrator running DFSCONF .
_IOE_CFG_CELL_ID	Defines the DCE principal name of the administrator who is running DFSCONF . The default value cell_admin is used if this is not specified.
NLSPATH	Defines the search path used by DCE for the DCE and DFS message catalogs. The default value, /usr/lib/nls/msg/En_US.IBM-1047/%N , is used if this is not specified.

Refer to Appendix C, “Environment Variables in DFS” on page 153 for more information on these environment variables.

Important Note About the DCE Cell Name and DCE Host Name

It is critical that the DCE cell name and the DCE host name, specified on the **DFSCONF** configuration panels, are correct. They must be the same names used to configure DCE. DFSCONF displays the DCE cell name and the DCE host name obtained from the DCE configuration file `/opt/dcelocal/dce_cf.db`.

Setting DFSCONF Environment Variables: The **DFSCONF** environment variables are declared in the **envvar** file in the home directory of the administrator performing the OS/390 DFS configuration. Environment variables are set using the following syntax:

VARIABLE_NAME=value

Following is an example of the entries in the **envvar** file for the **DFSCONF** program:

```
_EUV_SVC_MSG_LOGGING=CONSOLE_LOGGING
_EUV_SVC_MSG_LEVEL=VERBOSE
_IOE_CFG_INFORM_LEVEL=2
_IOE_CFG_LOG_FILE=dfsconf.log
_IOE_CFG_CELL_ID=cell_admin
NLSPATH=/usr/lib/nls/msg/En_US.IBM-1047/%N
```

Set Up the DFSCONF Administrator TSO User ID

You must set up a TSO user ID that has authority to run the OS/390 DFS configuration program. Following are the specifics of this user ID, the **DFSCONF** administrator:

- The **DFSCONF** administrator must have a superuser ID. In OS/390, the superuser or root has a **UID = 0**. A user can be accorded superuser privileges by specifying zero in the UID parameter of the **ADDUSER** or **ALTUSER** commands.
- Add the **SIOEPNLE** data set to the **ISPPLIB** concatenation.
- Add the **SIOEMSGE** data set to the **ISPMLIB** concatenation.
- Add the **SIOEXEC** library to the **SYSEXEC** or **SYSPROC** concatenation.

DCE Login Panel

To configure OS/390 DFS on the OS/390 host, you must be appropriately authenticated and authorized by the DCE Security Service. If you attempt to perform any of the functions provided by **DFSCONF** that require DCE authentication, **DFSCONF** automatically displays the DCE Login panel. You must then enter the correct Cell Admin ID and password. This ensures that you are logged in to DCE as the administrator who has all the necessary permissions to configure OS/390 DFS.

The DCE Cell Admin ID and DCE Cell Admin Password are usually the same as those used during the initial configuration of the cell. However, these may have been changed. Figure 3 on page 56 shows the DCE Login panel.

```

IOEBLGN----- DCE LOGIN -----
COMMAND ==>
Login for DCE Cell Name: dcecellname
  DCE Cell Name      ==> dcecellname
  DCE Cell Admin ID   ==>
  DCE Cell Admin Password ==>
Enter END COMMAND to return to previous menu.

F1=HELP    F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE

```

Figure 3. DCE Login Panel

If you have previously set the **_IOE_CFG_CELL_ID** environment variable (note, this needs to be specified in the home directory), its value is displayed in the **DCE Cell Admin ID** field of the panel. You can either accept it or overwrite it with a new value.

DFS Configuration Main Menu

After you run **DFSCONF** from TSO, the DFSCONF Main Menu displays. The DFSCONF Main Menu is shown in Figure 4.

```

IOEBMAIN----- DFS CONFIGURATION/DECONFIGURATION MAIN MENU -----
SELECT OPTION ==>

      1. Configure DFS Servers
      2. Deconfigure DFS Servers
      3. Configure DFS Client
      4. Deconfigure DFS Client

Enter END COMMAND to terminate.

*****
* copyright:  Licensed Materials - Property of IBM          *
*              5696-942 (C) Copyright IBM Corporation 1997, 1998.*
*              All rights reserved.                          *
*              US Government Users Restricted Rights - Use,   *
*              duplication or disclosure restricted by GSA ADP *
*              Schedule Contract with IBM Corp.              *
*              See Copyright Instructions, G120-2083.         *
*****

F1=HELP    F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP     F10=LEFT    F11=RIGHT   F12=RETRIEVE

```

Figure 4. DFSCONF Main Menu

The menu items are:

Menu Item	Description
Configure DFS Servers	If you select option 1 you get a list of all the DFS Servers that may be configured. See “Using DFSCONF to Configure DFS Servers.”
Deconfigure DFS Servers	If you select option 2 you get a list of all the DFS Servers that may be deconfigured. See “Using DFSCONF to Deconfigure DFS Servers” on page 124.
Configure DFS Client	If you select option 3 you perform the DFS Client configuration. See “Using DFSCONF to Configure the DFS Client” on page 59.
Deconfigure DFS Client	If you select option 4 you perform the DFS Client deconfiguration. See “Using DFSCONF to Deconfigure the DFS Client” on page 126.

Using DFSCONF to Configure DFS Servers

Selecting Configure DFS Servers from the DFSCONF Main Menu displays the panel shown in Figure 5.

```
IOEBCCFG----- CONFIGURE DFS SERVERS -----
COMMAND ===>
DCE Cell Name ===> dcecellname
DCE Host Name ===> dcehostname

Options:
DFS File Server          ===> N
DFS Backup Server        ===> N
DFS Fileset Database Server ===> N

Enter END COMMAND to return to previous menu.

F1=HELP    F2=SPLIT    F3=END    F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN     F9=SWAP   F10=LEFT    F11=RIGHT   F12=RETRIEVE
```

Figure 5. Configuring DFS Servers Panel

The menu items are:

Menu Item	Description
DCE Cell Name	This is the DCE cell name used to previously configure DCE on this system and should not be changed on this panel. It is displayed for information purposes only. The DCE cell name is obtained from the DCE configuration file /opt/dcelocal/dce_cf.db .
DCE Host Name	This is the DCE host name of the system on which DFS is being configured and should not be changed on this panel. It is displayed for information purposes only. The DCE host

name is obtained from the DCE configuration file **/opt/dcelocal/dce_cf.db**. The DCE host name is case sensitive.

DFS File Server

Enter **Y** to configure the DFS server on this host system. See “Using DFSCONF to Configure the DFS File Server.”

DFS Backup Server

Enter **Y** to configure the DFS Backup Server and the DFS Tape Coordinator on this host system. See “Using DFSCONF to Configure the DFS Backup Server.”

DFS Fileset Database Server

Enter **Y** to configure the DFS Fileset Database server on this host system. See “Using DFSCONF to Configure the DFS Fileset Location Database Server” on page 59.

Note: When you specify the **DCE Cell Name**, it is case sensitive and it is not fully qualified (for example, you can enter **dcecell1**, but not **/../dcecell1**).

Using DFSCONF to Configure the DFS File Server

The DFS File Server Configuration Program automates the configuration of the DFS File Server. You **must** first configure the DFS File Server before configuring the DFS Backup Server or DFS Fileset Database Server. After configuring the DFS File Server, you can configure or reconfigure the remaining servers in any combination. Therefore, you can specify **Y** on each of the options to have all the servers configured at once.

To initiate the configuration of this server, change the DFS File Server option from **N** to **Y** and press **<Enter>**.

The program begins to issue the DCE administration commands that configures your DFS Server. If not interrupted, and if successfully completed, the program displays the DFS File Server Configuration Ended....SUCCESSFULLY message.

If interrupted, or if the program did not successfully complete configuration, the DFS File Server Configuration Ended....UNSUCCESSFULLY message is displayed. To determine where the error occurred that caused the configuration to fail, examine the **dfsconf.log** file, which by default is in your home directory, or at the location specified by the environment variable **_IOE_CFG_LOG_FILE**. Refer to Appendix A, “Example DFS Server Configuration Log File (dfsconf.log)” on page 141 for an example of a **dfsconf.log** file. If the **dfsconf.log** file already exists any execution of the DFSCONF program appends any messages to the end of the log file.

After configuring the DFS File Server using **DFSCONF**, additional DFS File Server configuration issues described in “Completing the DFS File Server Configuration” on page 61 must be addressed. But, first, proceed to complete the next step in the “DFS Configuration Steps” on page 48.

Using DFSCONF to Configure the DFS Backup Server

If you want to run the DFS Backup Database Server (**bakserver**) or any DFS Backup Tape Coordinators (**butc01** through **butc08**) on an OS/390 DCE host system, you should configure the DFS Backup Server using **DFSCONF**.

The OS/390 DFS Server Configuration Program (**DFSCONF**) can be used to configure the Backup server (**bakserver** and **butc**) processes. The procedure is identical with that used in configuring other OS/390 DFS server processes. It is recommended that the DFS File Server be the first DFS server component configured on your system as described in “Using DFSCONF to Configure the DFS File Server.”

To initiate the configuration of the Backup Server, change the DFS Backup Server option from **N** to **Y** and press **<Enter>**.

The program begins to issue the DCE administration commands that configures your DFS Backup Server. If not interrupted, and if successfully completed, the program displays the DFS Server Configuration Ended....SUCCESSFULLY message.

If interrupted, or if the program did not successfully complete configuration, the DFS Backup Server Configuration Ended....UNSUCCESSFULLY message is displayed. To determine where the error occurred that caused the configuration to fail, examine the **dfsconf.log** file, which by default is in your home directory, or at the location specified by the environment variable **_IOE_CFG_LOG_FILE**. Refer to Appendix A, “Example DFS Server Configuration Log File (dfsconf.log)” on page 141 for an example output of a **dfsconf.log** file. If the **dfsconf.log** file already exists any execution of the **DFSCONF** program appends any messages to the end of the log file.

After configuring the Backup Server using **DFSCONF**, additional Backup Server configuration issues described in “Completing the Backup Database Server (bakserver) Configuration” on page 68 must be addressed. But, first, proceed to complete the next step in the “DFS Configuration Steps” on page 48.

Using DFSCONF to Configure the DFS Fileset Location Database Server

The OS/390 DFS Server Configuration Program (**DFSCONF**) can be used to configure the Fileset Location Database server (**flserver**) process. The procedure is identical with that used in configuring other OS/390 DFS server processes. It is recommended that the DFS File Server be the first DFS server component configured on your system as described in “Using DFSCONF to Configure the DFS File Server” on page 58.

To initiate the configuration for this server, change the DFS Fileset Database Server option from **N** to **Y** and press **<Enter>**.

The program begins to issue the DCE administration commands that configures your DFS Fileset Database server. If not interrupted, and if successfully completed, the program displays the DFS Server Configuration Ended....SUCCESSFULLY message.

If interrupted, or if the program did not successfully complete configuration, the DFS Fileset Database Server Configuration Ended....UNSUCCESSFULLY message is displayed. To determine where the error occurred that caused the configuration to fail, examine the **dfsconf.log** file, which by default is in your home directory, or at the location specified by the environment variable **_IOE_CFG_LOG_FILE**. Refer to Appendix A, “Example DFS Server Configuration Log File (dfsconf.log)” on page 141 for an example output of a **dfsconf.log** file. If the **dfsconf.log** file already exists any execution of the **DFSCONF** program appends any messages to the end of the log file.

After configuring the DFS Fileset Database server using **DFSCONF**, additional DFS Fileset Database server configuration issues described in “Completing the FLDB Server (flserver) Configuration” on page 61 must be addressed. But, first, proceed to complete the next step in the “DFS Configuration Steps” on page 48.

Using DFSCONF to Configure the DFS Client

Selecting Configure DFS Client from the **DFSCONF** Main Menu displays the panel shown in Figure 6 on page 60.

```

IOEBCL----- CONFIGURE DFS CLIENT -----
COMMAND ==>
DCE Cell Name ==> dcecellname
DCE Host Name ==> dcehostname

Options:
  DFS Client  ==> N
Enter END COMMAND to return to previous menu.

F1=HELP    F2=SPLIT    F3=END    F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN    F9=SWAP   F10=LEFT    F11=RIGHT   F12=RETRIEVE

```

Figure 6. Configuring the DFS Client Panel

The menu items are:

Menu Item	Description
DCE Cell Name	This is the DCE cell name used to previously configure DCE on this system and should not be changed on this panel. It is displayed for information purposes only. The DCE cell name is obtained from the DCE configuration file /opt/dcelocal/dce_cf.db .
DCE Host Name	This is the DCE host name of the system on which DFS is being configured and cannot be changed on this panel. It is displayed for information purposes only. The DCE host name is obtained from the DCE configuration file /opt/dcelocal/dce_cf.db . The DCE host name is case sensitive.
DFS Client	To initiate the configuration of the DFS Client, change the DFS Client option from N to Y and press <Enter> .

The OS/390 DFS Server Configuration Program (**DFSCONF**) can be used to configure the DFS Client (**DFSCM**).

The program begins to issue the DCE administration commands that configures your **DFSCM**. If not interrupted, and if successfully completed, the program displays the DFS Client Configuration Ended....**SUCCESSFULLY** message.

If interrupted, or if the program did not successfully complete configuration, the DFS Client Configuration Ended....**UNSUCCESSFULLY** message is displayed. To determine where the error occurred that caused the configuration to fail, examine the **dfsconf.log** file, which by default is in your home directory, or at the location specified by the environment variable **_IOE_CFG_LOG_FILE**. Refer to Appendix A, "Example DFS Server Configuration Log File (dfsconf.log)" on page 141 for an example output of a **dfsconf.log** file. If the **dfsconf.log** file already exists any execution of the **DFSCONF** program appends any messages to the end of the log file.

After configuring the DFS Client using **DFSCONF**, additional **DFSCM** configuration issues described in “Completing the DFS Client (DFSCM) Configuration” on page 72 must be addressed. But, first, proceed to complete the next step in the “DFS Configuration Steps” on page 48.

Completing the DFS File Server Configuration

To complete the configuration of the DFS File Server, you must identify the new DCE host system as a DFS file server system to other systems in the DCE cell. You must also complete the configuration of the server components that will comprise the DFS File Server on the DCE host system. This includes updating the associated DFS administrative files and starting the servers.

The DFS configuration steps will instruct you when to perform the additional configurations steps. For your reference, here are the sections where these additional server configurations steps are described:

1. “Completing the FLDB Server (flserver) Configuration”
2. “Creating the DFS File Server FLDB Entry” on page 63
3. “Completing Fileset Server (ftserver) Configuration” on page 63
4. “Completing Replication (repserver) Server Configuration” on page 65
5. “Completing Update Client (upclient) Configuration” on page 65
6. “Completing Update Server (upserver) Configuration” on page 66
7. “Completing the Backup Database Server (bakserver) Configuration” on page 68
8. “Customizing the ioepdcf File” on page 78
9. “Customization of the DFS Server and the DFSCM envvar Files” on page 80.

Completing the boserver Configuration

When configuring and running DFS, it is recommended that the **boserver** be automatically started. To automatically start, the **boserver**, update the **boserver** setting in the **ioepdcf** file. Refer to “Customizing the ioepdcf File” on page 78 for more information.

Completing the FLDB Server (flserver) Configuration

After using the OS/390 DFS Server Configuration Program, the following additional steps required to configure the Fileset Database Location Database (FLDB) Server (**flserver**) on a DCE host system.

Note: If you are configuring this DCE host system into an existing DCE cell, DFS administrative lists are typically updated on the cell's System Control machine and propagated by the Update Server to other DCE hosts running Update Clients. Refer to Chapter 8, “Defining DFS Administrators and Users” on page 83 for more information on DFS administrative lists and System Control Machines.

The following instructions assume that the system with DCE host name *dcehostname* is the system where the **flserver** and **upserver** will be run.

1. Use the **bos addadmin** command to add the abbreviated DFS server principal (**hosts/dcehostname/dfs-server**) of the new DFS server machine on the system with DCE host name *dcehostname* to the **/opt/dcelocal/var/dfs/admin.fl** administrative list file on DCE host system where the **flserver** will be being run.

You can use the **dcecp group add** command to add the abbreviated DFS server principal to a security group and include the group in the **admin.fl** administrative list file. Note that DFSCONF processing created the security group **subsys/dce/dfs-fs-servers** and added the member **hosts/dcehostname/dfs-server** for this DCE host system to the group. It is recommended that you add this group to the **admin.fl** file using the command:

```
$ bos addadmin -server /./hosts/dcehostname \
               -adminlist admin.fl -group subsys/dce/dfs-fs-servers -createlist
```

Alternatively, you can add the abbreviated DFS server principal for the DCE host system to the **admin.fl** file using the command:

```
$ bos addadmin -server /./hosts/dcehostname \
               -adminlist admin.fl \
               -principal hosts/dcehostname/dfs-server -createlist
```

2. Ensure that the **admin.fl** administrative list is also updated with administrator principals and groups in the **/opt/dcelocal/var/dfs** directory on DCE host system where the **flserver** will be being run.

To update the **admin.fl** administrative list on the system with DCE host name *dcehostname*, issue the following:

```
$ bos addadmin -server /./hosts/dcehostname \
               -adminlist admin.fl \
               -principal cell_admin -group dfsteam -createlist
```

To verify the **admin.fl** administrative list has been updated on the new database server machine with DCE host name **DCEDFS**, issue the following:

```
$ bos lsadmin /./hosts/DCEDFS -adminlist admin.fl
```

```
Admin Users are: user: cell_admin,
user: hosts/DCEDFS/dfs-server,
group: dfsteam,
group: subsys/dce/dfs-fs-servers
```

3. Stop and restart the **flserver** process on each system in the DCE cell. Restarting the existing database server processes causes the processes to read the updated RPC server group. This ensures that each Ubik coordinator agrees on the number and identities of the other database server machines of its type, which is vital to Ubik's use of a quorum of database server machines to maintain database consistency. For additional information on Ubik, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.
4. Start the **flserver** database server process on this DCE host machine using the **bos create** command. The following is an example of how to start the **flserver** on a system with the DCE host name of **DCEDFS**:

```
$ bos create /./hosts/DCEDFS flserver simple "flserver \
          envvar('_EUV_HOME=/opt/dfslocal/home/flserver')/ >dd:flserver 2>&1"
```

The **bos create** command will also instruct the **boserver** to automatically start the **flserver** when the **boserver** is subsequently restarted. For more information on using the **bos create** command, refer to "Starting the boserver Processes from OMVS" on page 136.

5. Verify that there is not already a DFS server entry for the system with DCE host name *dcehostname* by issuing the following command from OMVS which displays all server entries:

```
$ fts lsserverentry -all
```

If the output identifies that a list of sites can not be acquired (normally because this is the first system in the DCE cell where DFS will be configured) or the the **principal=hosts/dcehostname** identifying this DCE host system is not displayed, then a server entry must be created in the Fileset Location Databases (FLDB).

To create a DFS server entry in the FLDB with the DCE cell name *dcecellname* and the DCE host name *dcehostname* issue the command:

```
$ fts crserverentry -server /.../dcecellname/hosts/dcehostname \
                   -principal hosts/dcehostname
```

6. Synchronize the Fileset Location Databases in the DCE cell.

If the FLDB entry was created, it is recommended that you synchronize the Fileset Location Databases in the DCE cell by issuing the following command from OMVS using the DCE host name (*dcehostname*) for each system in the DCE cell where an **flserver** is configured:

```
$ udebug -rpcgroup /./fs -server /./hosts/dcehostname
```

For additional detailed information, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

This completes **flserver** configuration. You should proceed to the next configuration step.

Creating the DFS File Server FLDB Entry

It is recommended that you always create a DFS File Server entry in the Fileset Location Database (FLDB). You must create this entry for a DCE host system if you will ever export files using DFS from the DCE host system.

To create the FLDB entry, an **flserver** must be running in the DCE cell. Note that if you previously completed an **flserver** configuration on this DCE host system, you were instructed to create the entry for the DCE host system at that time.

To create the FLDB entry, use the following procedure:

1. Verify that there is not already a DFS server entry with the DCE host name of *dcehostname* by issuing the following command from OMVS to display all server entries:

```
$ fts lserverentry -all
```

If the output identifies that a list of sites cannot be acquired (normally because this is the first system in the DCE cell where DFS will be configured) or the **principal** = **hosts/dcehostname** identifying this DCE host system is not displayed, then a server entry must be created in the FLDB.

To create a DFS server entry in the FLDB with the DCE cell name of *dcecellname* and the DCE host name of *dcehostname* issue the following command:

```
$ fts crserverentry -server /.../dcecellname/hosts/dcehostname \  
-principal hosts/dcehostname
```

2. If the FLDB entry was created, it is recommended that you synchronize the Fileset Location Databases in the DCE cell by issuing the following command from OMVS using the DCE host name of *dcehostname* for each system in the DCE cell where an **flserver** is configured:

```
$ udebug -rpcgroup /./fs -server /./hosts/dcehostname
```

For additional detailed information, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Completing Fileset Server (ftserver) Configuration

After using the OS/390 DFS Server Configuration Program to configure the DFS File Server, the following additional steps required to configure the DFS Fileset Server on a DCE host.

If you are configuring this DCE host system into an existing DCE cell, DFS administrative lists are typically updated on the cell's System Control machine and propagated by the Update Server to other DCE hosts running Update Clients. Refer to Chapter 8, "Defining DFS Administrators and Users" on page 83 for more information on DFS administrative lists and System Control Machines.

Note: The following instructions assume that there is only one system in the DCE cell where DFS is running or that the **upserver** will be run on the same system with the **ftserver**.

1. Use the **bos addadmin** command to add the abbreviated DFS server principal (**hosts/dcehostname/dfs-server**) of the new DFS server machine on the system with DCE host name *dcehostname* to the **/opt/dcelocal/var/dfs/admin.ft** administrative list file.

You can use the **dcecp -c group add** command to add the abbreviated DFS server principal to a security group and include the group in the **admin.ft** administrative list file. Note that DFSCONF processing created the security group **subsys/dce/dfs-fs-servers** and added the member **hosts/dcehostname/dfs-server** for this DCE host system to the group. You can add this group to the **admin.ft** file using the command:

```
$ bos addadmin -server ./:/hosts/dcehostname -adminlist admin.ft \
               -group subsys/dce/dfs-fs-servers -createlist
```

Alternatively, you can add the abbreviated DFS server principal for the DCE host system to the **admin.ft** file using the command:

```
$ bos addadmin -server ./:/hosts/dcehostname -adminlist admin.ft \
               -principal hosts/dcehostname/dfs-server -createlist
```

2. Ensure that the **/opt/dcelocal/var/dfs/admin.ft** administrative list file is also updated with administrator principals and groups on the DCE host system where the **ftserver** will be run. To update the **admin.ft** administrative list file with the user **cell_admin** and the group **dfsteam** on the system with DCE host name *dcehostname*, issue the following:

```
$ bos addadmin -server ./:/hosts/dcehostname -adminlist admin.ft \
               -principal cell_admin -group dfsteam -createlist
```

To verify the **admin.ft** administrative list has been updated on the system with DCE host name *dcehostname*, issue the following:

```
$ bos lsadmin ./:/hosts/dcehostname -adminlist admin.ft
```

Admin Users are: user: cell_admin,
user: hosts/dcehostname/dfs-server,
group: dfsteam,
group: subsys/dce/dfs-fs-servers

3. Start the **ftserver** on this DCE host machine using the **bos create** command. The following is an example of how to start the **ftserver** on a system with the DCE host name of **DCEDFS**:

```
$ bos create ./:/hosts/DCEDFS ftserver simple "ftserver \
          envvar('_EUV_HOME=/opt/dfslocal/home/ftserver')/ >dd:ftserver 2>&1"
```

For more information on using the **bos create** command, refer to "Starting the boserver Processes from OMVS" on page 136.

4. Verify that there is not already a DFS server entry for the system with DCE host name *dcehostname* by issuing the following command from OMVS which displays all server entries:

```
$ fts lserverentry -all
```

If the output identifies that a list of sites can not be acquired (normally because this is the first system in the DCE cell where DFS will be configured) or the **principal=hosts/dcehostname** identifying this DCE host system is not displayed, then a server entry must be created in the Fileset Location Databases (FLDB).

To create a DFS server entry in the FLDB with the DCE cell name *dcecellname* and the DCE host name *dcehostname* issue the command:

```
$ fts crserverentry -server ../../dcecellname/hosts/dcehostname \
                   -principal hosts/dcehostname
```

For additional detailed information, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

This completes **ftserver** configuration. You should proceed to the next configuration step.

Completing Replication (repserver) Server Configuration

After using the OS/390 DFS Server Configuration Program, additional steps required to configure and start the Replication Server (**repserver**) on a DCE host system to replicate filesets or house a replica of a fileset. If you plan to define the **root.dfs** fileset for the DCE cell on the OS/390 host system where DFS is being configured, you must configure the **repserver**.

To complete the initial configuration and start a **repserver** on a DCE host system:

1. Ensure that a FLDB entry for the DCE host system running as a DFS File server machine is created as described in “Creating the DFS File Server FLDB Entry” on page 63.
2. Ensure that the **ftserver** is running on the DCE host system as described in “Completing Fileset Server (ftserver) Configuration” on page 63.
3. Start the **repserver** on this DCE host machine using the **bos create** command. The following is an example of how to start the **repserver** on a system with the DCE host name of **DCEDFS**:

```
$ bos create ././hosts/DCEDFS repserver simple "rpserver \  
    envvar('_EUV_HOME=/opt/dfslocal/home/repserver')/ >dd:rpserver 2>&1"
```

Note: Enter the previous command specifying **repserver** and **rpserver** exactly as shown.

The **bos create** command will also instruct the **boserver** to automatically start the **repserver** when the **boserver** is subsequently restarted. For more information on using the **bos create** command, refer to “Starting the boserver Processes from OMVS” on page 136.

You must refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for additional information on how to set up filesets for replication in a DCE cell.

This completes **repserver** configuration. You should proceed to the next configuration step.

Completing Update Client (upclient) Configuration

After using the OS/390 DFS Server Configuration Program, additional steps are required to configure and start the Update Client (**upclient**) on a DCE host system. This section explains how to complete the **upclient** configuration.

An **upclient** is run to allow the **upclient** to periodically request an **upserver** to identify the latest level of specific administrative files that exist on the **upserver** DCE host machine. If the latest level of a file does not exist on the **upclient** DCE host system, the **upclient** requests the latest level of the file from the **upserver** and over writes the file that is on the **upclient** DCE host system.

Note: You can run either an **upserver** or **upclient** on a DCE host system. You cannot run an **upserver** and an **upclient** on the same system. Also, the **upserver** or **upclient** must run in the same DCE cell.

Completing the configuration of the **upclient** consists of:

1. Updating the **admin.up** administrative list file on the **upserver** system.

To identify the DCE host system where the **upclient** will be run to the **upserver**, you must update the **/opt/dcelocal/var/dfs/admin.up** administrative list file on the DCE host system where the **upserver** is running.

For example, if the **upserver** is running on the system with the DCE host name *dcehostnameups* and the **upclient** will be run on the system with the DCE host name *dcehostname*, you can add the abbreviated DFS server principal for the **upclient** system (**hosts/dcehostname/dfs-server**) to the **admin.up** file on the **upserver** system using the following command:

```
$ bos addadmin -server ./:/hosts/dcehostnameups \  
-adminlist admin.up -principal hosts/dcehostname/dfs-server
```

2. Ensuring all administrative list files are updated on the **upserver** system before the **upclient** is started.

Once the **upclient** is started, DFS administrative list files on the system where the **upclient** is running can be overlayed by the interaction of the **upclient** with the **upserver**. You must ensure that all administrative list file updates made during the configuration of the DFS on this DCE host system are reflected in the administrative list files on the system where the **upserver** is running if the **upserver** is exporting the administrative list file. See Chapter 8, “Defining DFS Administrators and Users” on page 83 for more information.

3. Verifying **admin.up** file updates.

To verify that the **admin.up** administrative list file has been updated on a system with DCE host name *dcehostnameups*, issue the following:

```
$ bos lsadmin ./:/hosts/dcehostnameups -adminlist admin.up
```

```
Admin Users are: user: hosts/DCEDFS/dfs-server,  
group: dfsteam,  
group: subsys/dce/dfs-fs-servers
```

4. Starting the **upclient**

The following command is an example of how to start the **upclient** on the system with DCE host name *dcehostname* when the **upserver** is running on the system with DCE host name *dcehostnameups*:

```
$ bos create ./:/hosts/dcehostname upclient simple "upclient \  
  envvar('_EUV_HOME=/opt/dfslocal/home/upclient')/ >dd:upclient 2>&1 \  
  ./:/hosts/dcehostnameups \  
  (/opt/dcelocal/var/dfs/ admin.bos admin.ft admin.up) -f UpCLog"
```

The previous **bos create** command example requests that the **upserver** allow the **upclient** to acquire **admin.bos**, **admin.ft** and **admin.up** administrative list file information if the **upserver** has been started to export these files. The **bos create** command also causes the **boserver** to automatically start the **upclient** when the **boserver** is subsequently restarted.

Refer to “Starting the boserver Processes from OMVS” on page 136, for more information.

This completes **upclient** configuration. You should proceed to the next configuration step.

Completing Update Server (upserver) Configuration

Note: You can run either an **upserver** or an **upclient** on the host system where DFS is being configured. You cannot run an **upserver** and an **upclient** on the same host system.

After using the OS/390 DFS Server Configuration Program, additional steps are required to configure and start the Update Server (**upserver**) on a DCE host system. An **upserver** is run to allow an **upclient** to periodically request an **upserver** to identify the latest level of specified administrative files that exist on the **upserver** DCE host machine. If the latest level of a file does not exist on the **upclient** DCE host system,

the **upclient** requests the latest level of the file from the **upserver** and over writes the file that is on the **upclient** DCE host system.

Completing the configuration of the Update Server (**upserver**) consists of:

1. Updating the **admin.up** File on the **upserver** system.

Before starting the **upserver**, insure that the **admin.up** file is created (see “Defining a DFS Administrator” on page 84 and “Adding Principals and Groups to Administrative Lists” on page 89 and “System Control Machines and Domains” on page 85).

To identify the DCE host system where the **upclient** will be run to the **upserver**, you must update the **/opt/dcelocal/var/dfs/admin.up** administrative list file on the DCE host system where the **upserver** is running.

For example, if the **upserver** is running on the system with the DCE host name *dcehostnameups* and an **upclient** will be run on the system with the DCE host name **DCEDFS**, you can add the abbreviated DFS server principal for the **upclient** system (**hosts/DCEDFS/dfs-server**) to the **admin.up** file on the **upserver** system using the following command:

```
$ bos addadmin -server ./:/hosts/dcehostnameups \  
              -adminlist admin.up -principal hosts/DCEDFS/dfs-server
```

The **admin.up** file should contain an entry identifying the abbreviated DFS server principal for each file server machine in the domain where an **upclient** is run (see “Completing Update Client (upclient) Configuration” on page 65).

2. Verifying **admin.up** file updates.

To verify that the **admin.up** administrative list file has been updated on a system with DCE host name *dcehostnameups*, issue the following:

```
$ bos lsadmin ./:/hosts/dcehostnameups -adminlist admin.up
```

```
Admin Users are: user: hosts/DCEDFS/dfs-server,  
group: dfsteam,  
group: subsys/dce/dfs-fs-servers
```

3. Starting the **upserver**

The following is an example of how to create and start the **upserver** on the system with the DCE host name of *dcehostnameups*:

```
$ bos create ./:/hosts/dcehostnameups upserver simple "upserver \  
  envvar('_EUV_HOME=/opt/dfslocal/home/upserver')/ >dd:upserver 2>&1 \  
  (/opt/dcelocal/var/dfs/ admin.bos admin.up admin.ft)"
```

The previous **bos create** command example requests the **upserver** to export the **admin.bos** and **admin.ft** administrative files to any **upclient** with an entry in the **/opt/dcelocal/var/dfs/admin.up** file on the DCE host system where the **upserver** is running. The **bos create** command also causes the **boserver** to automatically start the **upserver** when the **boserver** is subsequently restarted.

Refer to “Starting the boserver Processes from OMVS” on page 136 for more information.

This completes **upserver** configuration. You should proceed to the next configuration step.

Completing the Backup Database Server (bakserver) Configuration

This section describes how to complete the configuration of the DFS Backup Server (**bakserver**) and the Backup Tape Coordinator (**butc**) servers. The **bakserver** is not required on a DCE host system when DFS is running. However, if you plan to run a Backup Tape Coordinator (**butc**) server on the system you should configure **bakserver**.

If you are configuring this DCE host system into an existing DCE cell, DFS administrative lists are typically updated on the cell's System Control machine and propagated by the Update Server to other DCE hosts running Update Clients. Refer to Chapter 8, "Defining DFS Administrators and Users" on page 83, for more information on DFS administrative lists and System Control Machines.

Note: The following instructions assume that there is only one system in the DCE cell where DFS is running or that the **upserver** will be run on the same system with the **bakserver**. Also, DFS configuration command examples sometimes use `./:` to denote `.../dcecellname`.

1. Use the **bos addadmin** command to add the abbreviated DFS server principal (**hosts/dcehostname/dfs-server**) of the new DFS server machine on the system with the DCE host name of **dcehostname** to the **/opt/dcelocal/var/dfs/admin.bak** administrative list file.

You can use the **dcecp -c group add** command to add the abbreviated DFS server principal to a security group and include the group in the **admin.bak** administrative list file. Note that DFSCONF processing created the security group **subsys/dce/dfs-fs-servers** and added the member **hosts/dcehostname/dfs-server** for this DCE host system to the group. You can add this group to the **admin.bak** file using the following command:

```
$ bos addadmin -server ./:/hosts/dcehostname \  
              -adminlist admin.bak -group subsys/dce/dfs-fs-servers -createlist
```

Alternatively, you can add the abbreviated DFS server principal for the DCE host system to the **admin.bak** file using the following command:

```
$ bos addadmin -server ./:/hosts/dcehostname \  
              -adminlist admin.bak \  
              -principal hosts/dcehostname/dfs-server -createlist
```

2. Ensure that the **/opt/dcelocal/var/dfs/admin.bak** administrative list is also updated with administrator principals and groups on the DCE host system where the **bakserver** will be running. To update the **admin.bak** administrative list on the system with the DCE host name of **DCEDFS**, issue the following command:

```
$ bos addadmin -server ./:/hosts/dcehostname \  
              -adminlist admin.bak \  
              -principal cell_admin -group dfsteam -createlist
```

To verify that the **admin.bak** administrative list has been updated on the system with the DCE host name of **DCEDFS**, issue the following command:

```
$ bos lsadmin ./:/hosts/DCEDFS -adminlist admin.bak
```

```
Admin Users are: user: cell_admin,  
user: hosts/DCEDFS/dfs-server,  
group: dfsteam,  
group: subsys/dce/dfs-fs-servers
```

3. Stop and restart the **bakserver** process on each system in the DCE cell where it is running. Restarting the **bakserver** processes causes the updated RPC server group to be read. This ensures that each Ubik coordinator agrees on the number and identities of the other database server machines of its type, which is vital to Ubik's use of a quorum of database server machines to maintain database

consistency. You must refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for additional Ubik information.

4. Start the **bakserver** database server process on this DCE host machine using the **bos create** command. The following is an example of how to start the **bakserver** on a system with the DCE host name of **DCEDFS**:

```
$ bos create ./:/hosts/DCEDFS bakserver simple "bakserver \  
    envar('_EUV_HOME=/opt/dfslocal/home/bakserver')/ >dd:bakserver 2>&1"
```

Notes:

- a. Enter the previous command specifying **bakserver** and **bakserver** exactly as shown.
- b. The OS/390 DFS implementation assumes that the UUID for the **bakserver** is exported when DCE configures the Cell Directory Server (CDS). If the **bakserver** fails to start because the UUID for the **bakserver** is not exported, refer to “Re-Establish the bakserver Object UUID” on page 71 for a description of the recovery actions that you can take.
- c. The **bos create** command will also instruct the **boserver** to automatically start the **bakserver** when the **boserver** is subsequently re-started.

For more information on using the **bos create** command, refer to “Starting the boserver Processes from OMVS” on page 136.

5. Synchronize the Backup Databases in the DCE cell.

To insure that all the DFS backup databases in the DCE cell are synchronized, issue the following command from OMVS using the DCE host name (*dcehostname*) for each system in the DCE cell where a **bakserver** is running. :

```
$ udebug -rpcgroup ./:/subsys/dce/dfs/bak -server ./:/hosts/dcehostname
```

For additional detailed information, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

This completes **bakserver** configuration. You should proceed to “Completing the DFS Backup Tape Coordinator Configuration” or to the next configuration step.

Completing the DFS Backup Tape Coordinator Configuration

This section provides information on completing the DFS Backup Tape Coordinator (**butcnn**) configuration. You must complete the DFS Backup Database Server (**bakserver**) configuration before you complete the Backup Tape Coordinator (**butc**) configuration. Additionally, setting the Backup Tape Coordinator environment variables and creating a VSAM tape backup management file are discussed in this section and should be performed in the order they appear.

If you plan to use the OS/390 DFS Backup Tape Coordinator with the OS/390 DFS Backup Database Server, complete the following steps:

1. Verify that the Backup Database Server (**bakserver**) is running. Issue the **bos status** command to the DCE host system where the **bakserver** is running, by entering the following:

```
$ bos status ./:/hosts/dcehostname bakserver
```

Instance bakserver, currently running normally.

If a **bakserver** is not running in the DCE cell, a **bakserver** should be started before proceeding to complete **butc** configuration.

2. The OS/390 DFS implementation assumes that the object UUID for the **bakserver** is exported during DCE configuration when the Cell Directory Server (CDS) is configured.

Display the exported object UUID for the **bakserver** by issuing the following command from OMVS:

```
dcecp -c show rpcentry /./subsys/dce/dfs/bak
```

```
{003c4de0-9485-6182-07ef-b1e000000000}
```

If the command displays an object UUID (for example, 003c4de0-9485-6182-07ef-b1e000000000), then the object UUID is exported and you can proceed to the next configuration step.

If an object UUID is not shown and a message indicating that the name service object is not found is displayed, the UUID for the **bakserver** is not exported at this time. If the UUID is not exported, refer to “Re-Establish the bakserver Object UUID” on page 71, for a description of the recovery actions that you can take to allow **butc** configuration to continue.

3. Verify that a **bak addhost** command has been issued for the ID number associated with the **butcnn** server you want to start. Issue the following command to determine what **butcnn** servers are defined:

```
$ bak lshosts
```

Tape hosts:

```
Host /.../dcecellname/hosts/dcehostname, port offset 0
```

The **bak lshosts** command output shown above indicates that the **butc01** server is already defined at offset 0. No tape hosts are displayed if there are no **butc** servers defined on this DCE host system.

If you want to start a **butcnn** and it was not displayed by the **bak lshost** command, issue the following **bak addhost** command.

Note: The numeric offset value shown at the end of the **bak addhost** command is:

- 0 to indicate **butc01**
- 1 to indicate **butc02**
- 2 to indicate **butc03**
- 3 to indicate **butc04**
- 4 to indicate **butc05**
- 5 to indicate **butc06**
- 6 to indicate **butc07**
- 7 to indicate **butc08**

For example, if you want to start **butc01** on an OS/390 DCE host with the DCE host name of *dcehostname*, issue the following command:

```
$ bak addhost /./hosts/dcehostname 0
```

```
Adding host '/.../dcecellname/hosts/dcehostname' offset 0  
to tape list ...
```

```
The host name and port information were added to the list of tape  
hosts successfully
```

4. Issue the OS/390 operator command **modify start** command from the operator console:

```
modify dfs,start butc01
```

```
IOEN00106I DFS daemon BUTC01 is ready for requests.
```

Note: OS/390 DFS uses OS/390 services to control and dynamically allocate tape drives. Because of this, the **TapeConfig** feature of DFS is not necessary and not available in OS/390 DFS.

5. Set the Backup Tape Coordinator environment variables.

The **/opt/dfslocal/home/butcnn/envvar** files can be edited to set the environment variables for the **butcnn** processes, where valid entries for *nn* are 01 through 08. There are four environment variables that may be updated. All have pre-defined default values assigned during installation. These values may be changed if desired.

The environment variables and default values are:

_IOE_BUTC_DISKUNIT

The unit name to be used when dynamically allocating a data definition for disk data sets. This is the name passed to dynamic allocation when allocating the disk device. The default value is **SYSDA**.

_IOE_BUTC_DUMP_HLQ

The high-level qualifier for the dump (backup) data set names. The default value is **DFSBKUP**.

_IOE_BUTC_KSDS

The name of the virtual storage access method (VSAM) DFS tape backup management file. The default value is **DFSBUTC.KSDS.LABEL**. Refer to step 6 for more information.

_IOE_BUTC_TAPEUNIT

The unit name to be used when dynamically allocating a data definition for tape drives. This is the unit name passed to dynamic allocation when allocating the tape device. The default value is **3490**.

At this point, an alias must be created in the OS/390 master catalog for the backup data set high-level qualifier. This is specified in the **_IOE_BUTC_DUMP_HLQ** environment variable for the appropriate **butc** process (01 through 08). The default value is **DFSBKUP**. If you choose another name as an alias, the name may be up to 17 characters in length.

For further information on how to set environment variables, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*. All variables relevant to DFS administration are listed in Appendix C, "Environment Variables in DFS" on page 153.

6. Create a VSAM tape backup management file.

A VSAM file should be created to maintain a record of each dump processed on the OS/390 system. The name of this file must be specified in the environment variable, **_IOE_BUTC_KSDS**. The file should contain the tape name, the dump number, and the tape label. The tape label will contain the tape's expiration date. A new record is added each time a new label is created and written to tape. When the expiration date is reached, the tape becomes expired and is eligible for re-use. The record for the expired tape is deleted from the VSAM data set, the dump information is deleted from the backup database, and the data sets contained in the dump are uncataloged.

To create the VSAM dataset to contain the tape management backup file identified by the environment variable **_IOE_BUTC_KSDS**, you can copy, edit and submit the IBM supplied sample job statements found in SIOESAMP(BUTCVSAM). To print this VSAM dataset, you can copy, edit and submit the sample job statements found in SIOESAMP(BUTCVSMP).

Note: If a VSAM data set is not created, an error message will be generated by the **butc** process. The dump process will, however, continue to process correctly and no expired backup records will be deleted from the system.

Re-Establish the bakserver Object UUID

The OS/390 DFS implementation assumes that the universal unique identifier (UUID) for the **bakserver** is exported when DCE configures the Cell Directory Server (CDS).

You can verify that the UUID for the **bakserver** is exported using the DCE RPC control program (**rpccp**):

```
$ rpccp show entry /./subsys/dce/dfs/bak
objects:
  3396dbb0-1bc1-11d0-a4f6-02608ce88731
>>> no matching binding information found
group members:
  /.../dces390.endicott.ibm.com/hosts/allanon.endicott.ibm.com/self
```

If an object UUID (for example, 3396dbb0-1bc1-11d0-a4f6-02608ce88731) is not displayed in the **rpccp show entry** command output and the message **EUVR12352I Objects not found in server entry** is returned, the UUID for the **bakserver** is not exported at this time. To recover from this condition, you can:

1. Stop other bakservers running in the DCE cell on different DCE host systems.
2. Generate a new UUID using the **uuidgen** command:

```
$ uuidgen
```

The *uuid-value* is displayed by the **uuidgen** command.

3. Issue the **rpccexport** command using the *uuid-value* returned by the **uuidgen** command:

```
$ rpccp export -o uuid-value /./subsys/dce/dfs/bak
```

4. Re-issue the **rpccp show entry /./subsys/dce/dfs/bak** command (see above) to verify that the UUID object is exported.
5. Restart the **bakservers** in the DCE cell. For information on how to start the **bakserver** on an OS/390 DCE host system, refer to “Starting the boserver Processes from OMVS” on page 136.

Completing the DFS Client (DFSCM) Configuration

This section describes additional configuration topics for the DFS client (**DFSCM**) in an OS/390 environment. They include:

- “System Parmlib Member BPXPRMxx Entry for the DFSCM”
- “DFSCM Cache Configurations” on page 73
- “DFSCM User .profile File Considerations” on page 77
- “DFSCM Environment Variable Considerations” on page 77.

System Parmlib Member BPXPRMxx Entry for the DFSCM

This section describes the system parmli member **BPXPRMxx** entry required to run the DFS Client (**DFSCM**).

Note: The variable, *xx*, is an installation dependent suffix used to uniquely identify system parmli members.

The **DFSCM** runs as a physical file system and requires a **FILESYSTYPE TYPE(DFSC)** entry in the system **BPXPRMxx** parmli member. This entry is usually made during the DFS installation process as described in the *OS/390 Program Directory* instructions used to install OS/390 and the Distributed File Service (DFS) element.

In case the **BPXPRMxx** parmli member for the **DFSCM** was not made when OS/390 was installed, you must now update the **FILESYSTYPE TYPE(DFSC)** entry in the parameter library (parmli) member **BPXPRMxx**. Refer to the *OS/390 UNIX System Services File System Interface Reference*, SC28-1909, for detailed information on defining a physical file system on OS/390 using the system parmli member **BPXPRMxx FILESYSTYPE** statement.

To cause OS/390 to start the **DFSCM**, make the following addition to the parameter member **BPXPRMxx** in the data set that is allocated to the IEFPARM DD statement in the OMVS Started Task PROC:

```
FILESYSTYPE TYPE(DFSC)
    ENTRYPOINT(IOECMINI)
    PARM('ENVAR("_EUV_HOME=/opt/dfslocal/home/dfscm") /
>DD:IOEDFSD 2>&1')
    ASNAME(DFSCM)
```

The following provides additional information regarding the above required entries in the parmlib member, **BPXPRMxx**:

TYPE(DFSC)

This entry must be specified as shown or the DFS client will fail in an indeterminate manner. **DFSC** is the well-known name that is used to identify the DFS client (**DFSCM**) as a physical file system.

ENTRYPOINT(IOECMINI)

This entry must be specified as shown or OS/390 will not successfully initialize the DFS client.

_EUV_HOME

This environment variable must be specified in the **FILESYSTYPE** statement. A directory other than **/opt/dfslocal/home/dfscm** may be specified but the use of this directory for the DFS client is recommended.

Note: A trailing slash '/' after the **_EUV_HOME** specification is required. **_EUV_HOME** is a DCE runtime option (not a **DFSCM** parameter). A slash identifies the end of the runtime options and the start of the parameters. If the slash '/' is omitted, it indicates that there are no runtime options and **_EUV_HOME** is not defined because it is treated as a **DFSCM** parameter.

>DD:IOEDFSD 2>&1

This entry must be specified as shown. It assigns standard out (**STDOUT**) and standard error (**STDERROR**) for the **DFSCM** DCE login processing to the ddname IOEDFSD. The ddname IOEDFSD is required to be included as started procedure JCL for the DFS client.

ASNAME(DFSCM)

Specifies the name of a procedure in SYS1.PROCLIB that is to be used to start the address space that is initialized by the DFS Client that is run as UNIX System Services physical file system (PFS). The name is 1 to 8 characters and the system converts the name to uppercase. The name you specify is also used for the name of the DFS Client address space. It is recommended that the name DFSCM be specified because the procedure named DFSCM (alias for the IOEP0002) is supplied in the hlq.SIOEPROC library and was used to define the DFS Client as a started task to RACF in the *OS/390 Program Directory* instructions. If an alternate name is specified, a new procedure library member must be created with that name and the procedure name defined to RACF.

DFSCM Cache Configurations

This section describes the different configurations available for the DFS client (**DFSCM**) in an OS/390 system environment which include:

- "Minimum Configuration of DFSCM" on page 74
- "DFSCM Memory Caching" on page 74
- "DFSCM Disk Caching" on page 75.

Minimum Configuration of DFSCM: The minimum configuration for the **DFSCM** (or DFS) on an OS/390 system is to run with memory caching without any DFS servers. **DFSCM** does not require any DFS servers be running on the OS/390 system where **DFSCM** is running.

DFSCM Memory Caching:

To configure an OS/390 system to run the **DFSCM** with memory caching, use the following procedure:

1. Create and update the local Hierarchical File System **/opt/dcelocal/etc/CacheInfo** file appropriately.

The **CacheInfo** file defines the DCE Local File System aggregate to use for a disk cache and the size of a disk or memory cache. The Cache Manager checks this file at initialization to determine this information. (The installation instructions provide details for creating the **CacheInfo** file.)

The **CacheInfo** file contains the following three fields separated by colons.

field 1 A directory on the local disk where the Cache Manager mounts the DFS global namespace.

Note: If **/...** is not specified, symbolic links to the global namespace fail.

field 2 The minor device number for a local DCE Local File System aggregate that serves as the DFS cache for a disk cache. The Cache Manager creates its cache files in this aggregate. There is no default for this entry. The entry must be specified.

Note: Although this aggregate is not used with a memory cache, a placeholder entry must appear in this field even if memory caching is used for the DFS client.

field 3 A definition of the cache size in kilobyte blocks.

Following is an example of a **CacheInfo** file. The file lists the DCE namespace mounted at the global namespace designation (**/...**), the minor device number of **999** for the local DCE Local File System aggregate used for the cache files, and a defined cache size of 512 kilobyte blocks (the aggregate must have this many blocks available on its disk(s)):

```
/...:999:512
```

Notes:

- a. A default **CacheInfo** file is usually created in **/opt/dcelocal/etc** during installation by the **dfs_cpfiles** program. Refer to the *OS/390 Program Directory* for this release or to the Chapter 6, "Post Installation Considerations" on page 41 for more information on the **dfs_cpfiles** program.

- b. The **DFSCM** default is disk caching. But, the **DFSCM** initialization parameter, **-memcache**, is specified as an **_IOE_CM_PARMS** environment variable value in the **/opt/dfslocal/home/dfscm/envar** file created by the **dfs_cpfiles** program during the OS/390 installation process. Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information on specifying **DFSCM** startup parameters using the environment variable **_IOE_CM_PARMS**.

2. Create and update the **/opt/dfslocal/home/dfscm/envar** file using the **DFSCM** initialization parameter, **_IOE_CM_PARMS**, to specify **-memcache**.

This step is necessary as the DFS client (**DFSCM**), when initialized by OMVS, reads the **envar** file specified as the **_EUV_HOME** variable value in the parameter library **BPXPRMxx FILESYSTYPE TYPE(DFSC)** entry. A **_EUV_HOME** variable value must be specified in the **BPXPRMxx** entry for the **DFSCM**. The use of the file **/opt/dfslocal/home/dfscm/envar** is recommended for **DFSCM** processing.

The DCE and DFS Environment Variables that are key to the DFS client are listed in "Customizing the DFSCM envar File" on page 81. For information on other DCE and DFS environment variables, see Appendix C, "Environment Variables in DFS" on page 153.

Note: An OS/390 shell configuration script program, **dfs_cpfiles**, is provided in the **/opt/dfsglobal/scripts** directory to copy an example **dfscm.envar** file from the **/opt/dfsglobal/examples** directory to **/opt/dfslocal/home/dfscm**. For further information, see “Using dfs_cpfiles to Create Configuration Files (dfscpfil)” on page 41.

Following is an example of a **DFSCM envar** file. The first line in the file disables the DFS client usage of the DCE single sign-on function. This controls whether the DFS client will attempt a DCE single sign-on for a user during a file request. The second line in the file controls where **DFSCM** messages are to be sent. In this example, they will be sent to the operator's console. The third line in the file passes initialization parameters to **DFSCM** or options that will take effect when the DFS client is started. In this example, the mounted file system name for the DFS client is set to **IOE_DFS_CLIENT_DATA** (this name is displayed by the OS/390 UNIX command, **df**). Translation is set to **ON** for ASCII to EBCDIC by the **-translation text** entry. Memory caching is specified for the DFS client. The last line in the example sets the time zone to Eastern Standard Time.

```
_EUV_AUTOLOG=NO
_EUV_SVC_MSG_LOGGING=CONSOLE_LOGGING
_IOE_CM_PARMS=-mountfilesystem IOE_DFS_CLIENT_DATA -translation text -memcache
TZ=EST5EDT
```

Note: The **DFSCM envar** file example shown above reflects the contents of the file after the example **DFSCM envar** file has been copied from the **/opt/dfsglobal/examples** directory.

3. Stop and restart the **DFSCM**. For details on stopping **DFSCM**, see “Starting and Stopping the DFS Client (DFSCM)” on page 131.

DFSCM Disk Caching: The **DFSCM** disk cache resides on a local DCE Local File System aggregate that must be allocated and defined on the local OS/390 system. **DFSCM** disk caching does not require the running of any DFS servers on the OS/390 system on which **DFSCM** is running.

To configure an OS/390 system to run **DFSCM** with disk caching:

1. Update the parmlib member, **BPXPRMxx**, to add the **FILESYSTYPE TYPE(DFSC)** entry. Refer to “System Parmlib Member BPXPRMxx Entry for the DFSCM” on page 72 and to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for further information.
2. Update the local Hierarchical File System file, **/opt/dcelocal/etc/CacheInfo**, appropriately. Refer to “DFSCM Memory Caching” on page 74 and to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for further information.
3. Specify disk caching by adding the **-dcache** entry to the **DFSCM** initialization parameter, **_IOE_CM_PARMS** in the **/opt/dfslocal/home/dfscm/envar** file. (Note: **-dcache** is the default for **DFSCM** processing if **-memcache** is not specified.) Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for further information.
4. Allocate and define a local DCE Local File System aggregate.

The SIOESAMP library member **NEWAGGR** contains sample JCL and control statements that can be modified to allocate VSAM Linear Datasets and initialize a Logical Volume as a DCE Local File System aggregate.

- a. Modify the **/opt/dfslocal/var/dfs/devtab** file. The **devtab** file describes all the Linear Datasets that make up a Logical Volume. It consists of the names of the Linear Datasets that make up the Logical Volume.

```
* Devtab - Example Entry for a logical volume
define_lfs 999
DFS.DCELFS.AGGR999.LDS00001
DFS.DCELFS.AGGR999.LDS00002
```

The previous example defined the device name **/dev/lfs999** that is specified as a PGM=IOENEWAG parameter in the JCL example in item 4c on page 76 that initializes a logical volume as a DCE Local File System aggregate.

- b. Use JCL similar to the following to allocate one or more VSAM Linear Datasets that will make up a Logical Volume. The Linear Datasets may reside on the same or different OS/390 disk volumes. The size of the Linear Datasets determines the size of the DCE Local File System. Refer to the SIOESAMP library member **NEWAGGR** for example JCL.

```
//DFSDEFIN JOB , 'DFS Define LDSs',
//          CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*-----
//*
//* Allocate a VSAM Linear Datasets for use as a Logical Volume
//*
//*-----
//DEFINE    EXEC    PGM=IDCAMS
//SYSPRINT DD      SYSOUT=*
//SYSUDUMP DD      SYSOUT=*
//AMSDUMP  DD      SYSOUT=*
//DASD0    DD      DISP=OLD,UNIT=3390,VOL=SER=PH2020
//DASD1    DD      DISP=OLD,UNIT=3390,VOL=SER=PH2021
//SYSIN    DD      *
//          DEFINE CLUSTER (NAME(DFS.DCELFS.AGGR999.LDS00001) VOLUMES(PH2020) -
//              LINEAR CYL( 50 0) SHAREOPTIONS(2) )
//          DEFINE CLUSTER (NAME(DFS.DCELFS.AGGR999.LDS00002) VOLUMES(PH2021) -
//              LINEAR CYL( 50 0) SHAREOPTIONS(2) )
//
```

- c. Run JCL similar to the following to initialize the Logical Volume as a DCE Local File System aggregate. Refer to the xxx.SIOESAMP(NEWAGGR) library member (where xxx is installation dependent) for example JCL. To access the **devtab** file, the PGM=IOENEWAG must be run under **UID = 0 (root)**.

```
//DFSNEWAG JOB , 'DFS NewAggr',
//          CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)
//*-----
//*
//* Format the Logical Volume as a DCE Local File System aggregate.
//*
//* This version of newaggr also "loads" the VSAM linear data
//* set(s) if they have not been previously loaded.
//*
//* NOTES:
//*      - Note that the "/dev/lvm" *must* be lower case!
//*      - Note that the extra '/' is required because LE
//*        runtime parameters are separated from program
//*        parameters by a '/'.
//*
//* The syntax of the newaggr command is as follows:
//*
//* newaggr -aggregate name -blocksize bytes -fragsize bytes
//*          [-initialempty blocks] [-aggrsize blocks]
//*          [-logsize blocks] [-overwrite] [-verbose] [-noaction]
//*
//*-----
//NEWAGGR EXEC PGM=IOENEWAG,
//          PARM=('//dev/lfs999 8192 1024 -verbose')
//*
```

```
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//*
```

Notes:

- 1) IBM recommends using a **-blocksize 8192** for optimum disk cache performance and storage utilization.
 - 2) No updates to the **/opt/dfslocal/var/dfs/dfstab** file are required since the local DCE File System aggregate used for **DFSCM** disk caching is not exported. Also, **DFSCM** processing completes any initialization of the aggregate to be used for disk caching.
5. Stop and restart the **DFSCM**. For details on stopping the **DFSCM**, see “Starting and Stopping the DFS Client (DFSCM)” on page 131.

DFSCM User .profile File Considerations

To reduce overall processing overhead associated with the **DFSCM** support for the OS/390 DCE single sign-on function, it is recommended that users issue the following command after initializing OS/390 UNIX:

```
$ cd /...
```

This can be automated by updating the users **/home/mvs_userid/.profile** file. Edit the file, adding the following statements:

```
cd /...
cd users_home_directory
```

The first entry will change the directory to the DFS global namespace's pathname prefix, **/...**. The second entry changes the directory to the user's home directory.

Enabling DFSCM to Use DCE Single Sign-on

OS/390 DFS supports OS/390 DCE's single sign-on function. To enable DCE single sign-on processing for a user accessing the DFS global namespace, **/...**, a Resource Access Control Facility (RACF) segment must first be created. If there are currently no DCE credentials (or they have expired) when the DFS client is invoked, then the DFS client does one of the following:

- Attempts a DFS single sign-on, or
- Sends the file request as an unauthenticated request.

Note: Although RACF is mentioned, any OS/390 external security manager (ESM) that has equivalent support can be used instead of RACF.

For detailed information regarding the DFS client usage of the DCE single sign-on function, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

For user **.profile** file considerations related to DCE single sign-on function, see “DFSCM User .profile File Considerations.”

DFSCM Environment Variable Considerations

To complete the **DFSCM** configuration, you can optionally customize the **DFSCM** envvar file as described in Appendix B, “Directories and Files” on page 147.

Customizing the ioepdcf File

On OS/390 DFS, the DFS Server **ioepdcf** file contains information for each server process you can run in DFS. This file can be customized to meet your needs. The **ioepdcf** file must be located in the following directory: **/opt/dfslocal/etc**. An example **ioepdcf** file can be found in **/opt/dfsglobal/examples**.

If you are installing OS/390 DFS for the first time, you can copy the example file to the **/opt/dfslocal/etc** directory and customize the file. The shell script, **/opt/dfsglobal/scripts/dfs_cpfiles**, automates this procedure (see “Using dfs_cpfiles to Create Configuration Files (dfscpfil)” on page 41 for further information).

If you have previously installed an earlier version of OS/390 DFS, you will normally be able to maintain your customization by using your existing **ioepdcf** file stored in **/opt/dfslocal/etc**.

The following describes how to customize the **ioepdcf** file for your system.

1. Determine what OS/390 DFS processes you want to run.

All processes running under the control of the Control Task program, **DFSCNTL**, are valid process names that can be specified in the **ioepdcf** file. These processes are:

- **dfskern**
- **export**
- **unexport**
- **boserver**
- **butcnn**. Valid entries for *nn* are 01 through 08.

2. Specify a configuration type for each process you have selected.

Each process identified in the **ioepdcf** file, must specify a configuration type. There are four valid types available:

Configuration Type	Explanation
--------------------	-------------

CONFIGURED=Y	The specified process will start during DFS initialization. Should the process abend or end for any reason, it will automatically be restarted by the DFS Control Process. Note: This configuration type is not supported with the butc processes.
CONFIGURED=N	The process will not be started by the DFS Control Process. The process cannot be started manually.
CONFIGURED=I	The specified process will be started during DFS initialization or when the OS/390 Operator command modify dfs,start all is issued. The process may also be started manually. The specified process is not restarted when it is terminated. Note: This configuration type is not supported with the butc processes.
CONFIGURED=M	The process will not be started by the DFS Control Process but can be manually started by an OS/390 system modify command to DFS. The specified process will not restart if it ends for any reason. Always use CONFIGURED=M for butcnn processes.

3. Identify the name of the load module (**LMD**) which corresponds to the name of the member in the SIOELMOD data set in the OS/390 partitioned data set (PDS) load library. The SIOELMOD data set was created during installation.

Each process entry in the **ioepdcf** file must contain the name of the **LMD** which refers to the name of the member in the SIOELMOD data set in the OS/390 partitioned data set (PDS) load library. For further information, refer to the *OS/390 Program Directory*.

The following are the PDS member names for the DFS processes started by the DFS Control Task:

- boserver** The **boserver** daemon load module name. The name, **boserver**, is an alias for the load library entry, **IOEBOSRV**.
- butcnn** The **butcnn** daemon load module name. There are eight valid load module names for the butc processes: **butc01**, **butc02**, **butc03**, **butc04**, **butc05**, **butc06**, **butc07**, and **butc08**. All are valid aliases for the single load module used for all **butcnn** processes- **IOEBUTC**. Always use CONFIGURED=M for **butcnn** processes. The **butcnn** processes require **bakserver** to be running to complete configuration.
- dfskern** The **dfskern** daemon load module name. The name, **dfskern**, is an alias for the load library entry, **IOEDFSKN**.
- export** The **export** process is started by the DFS Control Task and executes the load library entry, **IOEDFSXP**. The **export** process has no alias.

4. Include any special parameters to be passed to the **LMD** when a process is started.

Each process entry in the **ioepdcf** file includes special parameters that will be passed to the load module when a daemon is started (including Language Environment/370 (LE/370) runtime options). This is also called the argument list. Any runtime overrides such as storage specifications and redirection of output may also be added. The first argument is the home directory for each process. The home directory points the process to the directory holding the environment variable (**envar**) file for the process. Program parameters for the DFS process are preceded with a slash, **/**, in the argument list.

5. Include any special parameters for controlling restart and timeout intervals for your system.

Additional special parameters that control restart and timeout intervals may also be entered in the **ioepdcf** file. The **ioepdcf** file can also be used to override the default parameter options for the **file exporter (fxd)** daemon. Changes are specified in the **dfskern** process entry. For further information on the **ioepdcf** file, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

The following example shows an **ioepdcf** file entry for the **dfskern** process. The configuration type is **Y**, specifying that the process start during DFS initialization. The load module, **LMD**, is identified as **DFSKERN**. In the argument list, **ARG**, **ENVAR** indicates the environment variables to be used. In the example, the home directory is identified as **_EUV_HOME=/opt/dfslocal/home/dfskern**. An LE/370 runtime option is specified: **RPSTG(OFF)**. Parameters for the **fxd** program follow the **/**. The **>** symbol is a redirection character which indicates that the output will be redirected to the data definition name (ddname) that follows, **DFSKERN**. The **2>&1** entry redirects **STDERR** to the same destination as **STDOUT**, **DFSKERN**. **RESTART** and **TIMEOUT** values are both set at 300 seconds. The **RESTART** parameter specifies how long the process must be down before DFS control will attempt to restart it. The **TIMEOUT** parameter specifies how long DFS control will wait for the process before it times out.

```
DFSKERN CONFIGURED=Y LMD=DFSKERN \
  ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/dfskern'),RPSTG(OFF)/ \
  -mainprocs 7 -admingroup subsys/dce/dfs-admin \
  >DD:DFSKERN 2>&1" RESTART=300 TIMEOUT=300
```

Note: The **-admingroup** parameter is required for the file exporter daemon (**fxd**).

Customization of the DFS Server and the DFSCM envar Files

This chapter provides information on the following topics:

- “Customizing the DFS Server envar Files” on page 80
- “Customizing the DFSCM envar File” on page 81.

Customizing the DFS Server envar Files

Before OS/390 DFS can be initialized, each DFS process that you choose to start must have a corresponding **envar** file created. On OS/390 DFS, the **envar** file for each server process includes information that affects the behavior of the process. The variables for each **envar** may be customized. Setting or changing these variables is optional. Default values are provided if you choose not to set them.

To function properly, the files **must** be stored in the appropriate directories. The following table shows the correct directory for each processes' **envar** file, and the name of the example **envar** supplied in the **/opt/dfsglobal/examples** directory.

Table 15. Directory Locations for envar Files on OS/390 DFS		
OS/390 DFS Process	Directory Where Located in /opt/dfslocal	IBM-Supplied example envar in /opt/dfsglobal/examples
bakserver	home/bakserver/envar	bakserver.envar
boserver	home/boserver/envar	boserver.envar
butc01	home/butc01/envar	butc01.envar
butc02	home/butc02/envar	butc02.envar
butc03	home/butc03/envar	butc03.envar
butc04	home/butc04/envar	butc04.envar
butc05	home/butc05/envar	butc05.envar
butc06	home/butc06/envar	butc06.envar
butc07	home/butc07/envar	butc07.envar
butc08	home/butc08/envar	butc08.envar
daemonct	home/daemonct/envar	daemonct.envar
dfscm	home/dfscm/envar	dfscm.envar
dfscntl	home/dfscntl/envar	dfscntl.envar
dfsexport	home/dfsexport/envar	dfsexport.envar
dfskern	home/dfskern/envar	dfskern.envar
flserver	home/flserver/envar	flserver.envar
ftserver	home/ftserver/envar	ftserver.envar
growaggr	home/growaggr/envar	growaggr.envar
newaggr	home/newaggr/envar	newaggr.envar
repserver	home/repserver/envar	repserver.envar
salvage	home/salvage/envar	salvage.envar
upclient	home/upclient/envar	upclient.envar
upserver	home/upserver/envar	upserver.envar

If you are installing OS/390 DFS for the first time, the **envvar** files for each process you choose to run must be copied from **/opt/dfsglobal/examples** to the appropriate directory identified in the previous table. An OS/390 shell script provided with OS/390 DFS, **dfs_cpfiles**, automates the process of copying the IBM-supplied **ioepdcf** and **envvar** files from the **/opt/dfsglobal/examples** directories to the appropriate locations if they do not already exist from a previous installation of OS/390 DFS. **dfs_cpfiles** is located in the directory **/opt/dfsglobal/scripts**. For further information on this utility, see “Using **dfs_cpfiles** to Create Configuration Files (**dfscpfil**)” on page 41.

If you have previously installed an earlier version of OS/390 DFS, you will normally be able to use your existing **envvar** files without additional action. You can also take advantage of additional processes when you install the release, by copying the example **envvars** from **/opt/dfsglobal/examples** to the respective directories, **/opt/dfslocal/home/bakserver/envvar** or **/opt/dfslocal/home/flserver/envvar**. The OS/390 DFS Default Configuration Files Creation Program **/opt/dfsglobal/scripts/dfs_cpfiles** can also be used to automate this process (for further information on this utility, see “Using **dfs_cpfiles** to Create Configuration Files (**dfscpfil**)” on page 41).

The following example shows how to customize the settings for the **dfskern** process. This variable sets the number of threads to be started in **dfskern** to service token requests from the glue layer. The number of threads may be set to any number, *n*. The default is **5**.

_IOE_TKMGLUE_SERVER_THREADS=*n*

Note: Appendix C, “Environment Variables in DFS” on page 153 provides a description of all environment variables relevant to DFS administration. The appendix provides default values, examples, and descriptions of the environment variables. See the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for further information on setting environment variables.

Customizing the DFSCM envvar File

When initialized by OMVS, the DFS client (**DFSCM**) reads the environment variable, **envvar**, file specified as the **_EUV_HOME** variable value in the system parmlib member **BPXPRMxx FILESYSTYPE TYPE(DFSC)** entry. An **_EUV_HOME** variable value must be specified in the **BPXPRMxx** entry for the **DFSCM**. The use of the file, **/opt/dfslocal/home/dfscm/envvar**, is recommended for the **DFSCM**. See the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for more information.

The DCE and DFS environment variables key to the DFS client include:

- **_EUV_AUTOLOG**
- **_EUV_SVC_MSG_LOGGING**
- **_IOE_CM_PARMS**.

See Appendix C, “Environment Variables in DFS” on page 153, for a more comprehensive listing of the environment variables used in OS/390 DFS.

Chapter 8. Defining DFS Administrators and Users

A **DFS administrator** refers to an OS/390 user ID used to issue DFS administrator requests to a DFS server running on the same DCE host system or another DCE host system in the DCE cell. A **DFS user** refers to an OS/390 user ID used to issue file requests that access the DFS namespace. The file requests are routed through the DFS Client to the appropriate DFS server.

All DFS administrators and users must be authenticated and authorized by the DCE Security Service to be able to access the DFS servers. The DCE administrator has to create a DCE account for each DFS administrator and user.

Also, a DFS administrative list file must be created and updated for each DFS server process run on a DCE host system.

If DFS administrators and users also need access to HFS or RFS filesets exported by DFS servers, a relationship (a **mapping**) must be established between their DCE user IDs and their OS/390 user IDs on the host system where DFS servers are running and exporting the HFS or RFS filesets.

This chapter discusses the following topics related to defining DCE administrator and user accounts for DFS. This information should be performed in the order that appears.

- “Creating DCE User Accounts for DFS Administrators and Users”
- “Defining a DFS Administrator” on page 84
- “System Control Machines and Domains” on page 85
- “Description of DFS Administrative List Files” on page 86
- “Creating the Initial DFS Administrator(s)” on page 87
- “Disable/Enable DFS Authorization (NoAuth File)” on page 88
- “Creating the admin.bos File” on page 89
- “Adding Principals and Groups to Administrative Lists” on page 89
- “Mapping OS/390 User IDs for HFS and RFS Access” on page 92 includes the following sections:
 - “Registering Users With the OS/390 Security Subsystem” on page 92
 - “Mapping DCE User IDs to OS/390 User IDs” on page 92
- “Using the Identity Mapping File Method” on page 93
- “Using the RACF Identity Mapping Function” on page 97
- “DFS Anonymous User Considerations” on page 98.

Creating DCE User Accounts for DFS Administrators and Users

The DCE administrator (also known as the cell administrator) must add the DCE user account for the DFS administrator and users to the Security registry using the DCE administrative interface, **dcecp**.

The procedures for creating a DCE account described here assume that the principal, group, and organization have not been created prior to the creation of the account. Skip the relevant steps if the principal, group, and organization have previously been created for the user.

To create a DCE account, follow these steps:

1. Invoke the DCE administrative interface

⇒ **dcecp**

2. Use the **principal** subcommand to add the principal for the user. For example, if the user is named **smith**:

```
dcecp> principal create smith
```

This assigns a default UNIX number, full name, and object creation quota. If you want to enter values other than the defaults, see the *OS/390 DCE Administration Guide* for details.

3. Add the group (or groups) to which the user belongs. This step must only be performed if the group (or groups) have not been previously added to the registry. For example, if the name of the group is **usergrp**:

```
dcecp> group create usergrp -inprojlist yes
```

This assigns a default UNIX number, full name, and includes the group, **usergrp**, in the project list. If you want to enter values other than the defaults, see the *OS/390 DCE Administration Guide* for details.

4. Add the principal to the group. For example, to add the principal, **smith**, to the group, **usergrp**:

```
dcecp> group add usergrp -member smith
```

5. Add the organization to which the user belongs. This step must only be performed if the organization has not previously been created in the registry. For example, to add the organization, **ibm**:

```
dcecp> organization create ibm
```

The above assigns a default UNIX number and full name. If you want to enter values other than the defaults, see the *OS/390 DCE Administration Guide* for details.

6. Add the principal to the organization. For example, to add the principal, **smith**, to the organization, **ibm**:

```
dcecp> organization add ibm -member smith
```

7. Add the account for the user. For example, to add the account for a user whose principal is **smith**:

```
dcecp> account create smith -group usergrp -organization ibm \  
-password xxx -mypwd dce_principal_password
```

Note: In the above example, a user's password, **xxx**, associated with the newly created account and the administrator's password, *dce_principal_password*, must be supplied.

After creating the DCE user account for a DFS administrator, the applicable DFS administrative lists must be updated as described in “Defining a DFS Administrator.”

Additionally, if the DFS administrators and/or users need access to HFS and RFS filesets exported by DFS servers, their DCE user IDs must be mapped to their OS/390 user IDs on the host system for the DFS servers. This procedure is described in “Mapping OS/390 User IDs for HFS and RFS Access” on page 92.

Defining a DFS Administrator

Before an administrator is added to DFS, the principal has to be authenticated and authorized by the DCE Security Service to be able to perform administrative tasks. A DCE account has to be created by the DCE administrator for the DFS administrator. Creating an account for the DFS administrator follows the same procedure as creating an account for a DFS user. For detailed information regarding this procedure, see “Creating DCE User Accounts for DFS Administrators and Users” on page 83.

Note: The DCE **cell_admin** user ID inherits no special DFS authorization. The user ID **cell_admin** has to be explicitly specified in a DFS administrative list file.

In DFS, administrative lists are used to define the principals and groups that can perform actions affecting specific server processes on a server machine. There is one DFS administrative list for each DFS server process running on a machine. For example, a server's **admin.bos** file defines who has administrative rights to the BOS Server (**boserver**) and determines the users and servers that can manipulate and maintain server processes on that server.

Because administrative lists exist on a per-process and per-machine basis, different groups of principals can have different sets of administrative privileges within a domain. It is often useful to have the same group or user on several lists.

The following sections contain topics related to defining a DFS administrator:

- “System Control Machines and Domains”
- “Description of DFS Administrative List Files” on page 86
- “Creating the admin.bos File” on page 89
- “Creating the Initial DFS Administrator(s)” on page 87
- “Adding Principals and Groups to Administrative Lists” on page 89.

System Control Machines and Domains

A group of DCE host machines in a DCE cell where DFS server processes run is called a **domain**. There is a System Control Machine for each domain. The **upserver** process runs on the System Control Machine. The administrative list file, for each DFS server process run on any DCE host machine in the domain, is kept up to date and synchronized on each DCE host machine by the interaction of the Update Client (**upclient**) or the Update Server (**upserver**) processes running on the DCE host machines in the domain. Refer to *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information on System Control machines and domains.

Administrative list files should be updated on the DCE cell's System Control machine where the Update Server (**upserver**) is running. The **upserver** then propagates administrative list file changes to the secondary sites where an Update Client (**upclient**) is running. If an administrative file is updated on a DCE host machine where an **upclient** is running and the updates are not also made on the system where the **upserver** is running, the updates are overlaid on the **upclient** system if the **upserver** is acting as a system control machine for that file.

DFS Administrator Considerations for a DCE Cell

It is recommended that you add the same DFS administrator principals or groups to all the administrative list files in the DCE cell. This allows these DFS administrators to issue requests to any DCE host system in the DCE cell where DFS is running.

DFS Server Principal Considerations for a DCE Cell

When adding a new system with DCE host name *dcehostname* to a DCE cell, the abbreviated DFS server principal **hosts/dcehostname** for the new DCE host system should be added to the **/opt/dcelocal/var/dfs/admin.fl**, **/opt/dcelocal/var/dfs/admin.ft**, and **/opt/dcelocal/var/dfs/admin.bak** files on the other systems in the DCE cell where DFS is configured. This should be done even if the related DFS server is not being run on a DCE host system at this time. Updating the DFS administrative file on the system where the **upserver** is running enables the update to be propagated to other systems in the DCE cell running an **upclient**. Refer to the following sections for information on how to update these administrative files with the abbreviated DFS server principal:

- “Completing the FLDB Server (flserver) Configuration” on page 61
- “Completing Fileset Server (ftserver) Configuration” on page 63

- “Completing the Backup Database Server (bakserver) Configuration” on page 68.

Description of DFS Administrative List Files

In DFS, administrative lists are used to define the principals and groups that can perform actions affecting specific server processes on a server machine. There is one DFS administrative list for each DFS server process running on a machine. For example, a server's **admin.bos** file defines who has administrative rights to the BOS Server (**boserver**) and determines the servers and/or users that can manipulate and maintain server processes on a host system.

Administrative lists exist on a per-process and per-machine basis so different groups of principals can have different sets of administrative privileges within a domain. But, it is often useful to have the same group or user on several DFS administrative lists.

DFS **bos** commands are used to maintain the administrative list files. Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information on the DFS **bos** commands.

The default directory for the administrative lists is the configuration directory (**/opt/dcelocal/var/dfs**). If the lists are stored in the default directory, you need to provide only the specific file name (for example: **admin.bak**, **admin.bos**, **admin.fl**, **admin.ft**, or **admin.up**) in the DFS **bos** commands.

The DFS administrative list files are :

- The **admin.bak** file is associated with the Backup Server (**bkserver**). It designates the users and groups allowed to issue commands in the **bak** command suite. These commands are used to configure the Backup System and to dump and restore data. The Backup Database, like the FLDB, is typically replicated to several different machines in the cell. Therefore, you need to ensure that the **admin.bak** lists on all machines that house the Backup Database are identical.
- The **admin.bos** file is associated with the BOS Server (**boserver**). It designates the users and groups permitted to create, start, and stop DFS server processes and other processes to be controlled by the BOS Server on a machine. The BOS Server runs as **root**, so processes that it starts run with **root** privileges. Because they can direct the BOS Server to start any process, and because they can add and remove members from the other administrative lists on the machine, users in the **admin.bos** list are usually a subset of the users in the other lists for a machine or domain.
- The **admin.fl** file is associated with the Fileset Location Database Server (**flserver**). It designates the users and groups permitted to create server entries and fileset entries in the Fileset Location Database (FLDB). Because the FLDB is usually replicated to several different machines in the cell, you need to ensure that the **admin.fl** lists on all machines housing the FLDB are identical. Otherwise, an administrator may be able to execute a command from one machine but not from another. To simplify management, it is best that the server principal names of all server machines in the domain be represented in the **admin.fl** list on the System Control machine so that the list is distributed to all File Server machines in the domain. (The server principals can be included directly, or a group to which they belong can be included.)
- The **admin.ft** file is associated with the Fileset Server (**ftserver**). It designates the users and groups permitted to administer filesets on a machine. Because some fileset operations (such as moving filesets) affect multiple machines, the server principal names of the machines involved in the operations must also be in this administrative list. To simplify management, it is best that the server principal names of all server machines in the domain be represented in the **admin.ft** list on the System Control machine so that the list is distributed to all File Server machines in the domain. (The server principals can be included directly, or a group to which they belong can be included.)
- The **admin.up** file is associated with the Update Server (**upserver**). It contains the server principals for all server machines in the domain, allowing the **upclient** processes on those machines to obtain

files such administrative lists from the **upserver** process. The list should be stored on machines such as the System Control machine which runs the **upserver** process.

Administrative lists for server processes are created in one of two ways:

1. When a server is started and a list does not already exist. By default, a server process automatically creates an administrative list and places its list in the configuration directory (**/opt/dcelocal/var/dfs**). An administrative list generated by a process is always empty.

The BOS server **admin.bos** administrative list file is created before any other administrative list file using this method. Refer to “Creating the admin.bos File” on page 89 for a description of how to establish the **admin.bos** administrative list.

2. Administrative lists other than the **admin.bos** list are created by issuing the **bos addadmin** command with the **-createlist** option. Refer to “Adding Principals and Groups to Administrative Lists” on page 89 for a description of how to create an administrative list file using the **bos addadmin** command.

Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information on **bos** commands and the DFS administrative list files.

Creating the Initial DFS Administrator(s)

During the initial installation of DFS on a host system, an empty **admin.bos** file is created when **boserver** is first started if the file does not exist. Consequently, there may be no **admin.bos** file entries that define a DFS administrators.

The **boserver** authorization checking must be disabled to allow the initial DFS administrators to be defined. When the **boserver** is running, creating the **/opt/dcelocal/var/dfs/NoAuth** file temporarily disables **boserver** authorization.

The initial DFS administrators can be defined by:

1. Logging in as **root**.

Creating the file directly requires logging into the OS/390 system with a **root** userid with **UID = 0**.

2. Creating the **NoAuth** file to disable **boserver** authorization checking by issuing the command:

```
touch /opt/dcelocal/var/dfs/NoAuth
```

The **NoAuth** file must be created in the directory **/opt/dcelocal/var/dfs**. The file is always deleted when the **boserver** is re-started. Additionally, the **NoAuth** file is created by the **bos setauth ./:/hosts/dcehostname off** command and deleted by the **bos setauth ./:/hosts/dcehostname on** command.

3. The **bos status ./:/hosts/dcehostname** can be issued to determine if **boserver** authorization is disabled. Note that the **bos status** command reports if authorization is disabled but the command output does not explicitly report if authorization is enabled.

An example of the **bos status** command issued for the **boserver** running on a system with DCE host name *dcehostname* is :

```
$ bos status ./:/hosts/dcehostname
```

4. Define the initial DFS administrators while DFS authorization is disabled.

During the initial setup the initial setup of DFS on an OS/390 DCE host system, you should initially add a user and a group with DCE **cell_admin** authority to the **/opt/dfslocal/var/dfs/admin.bos** file which was created when the **boserver** was first started. Note that if DFS was previously configured

on the DCE host system, DFS administrators may still be defined in the various DFS administrator list files including the **bos admin** file.

Before any principal or group can be added to a DFS administrative list file, the DCE principal and group must be defined as described in “Creating DCE User Accounts for DFS Administrators and Users” on page 83.

The DFS configuration examples assumes that the userid **cell_admin** was previously defined during DCE configuration and that the group **dfsteam** was defined by the command:

```
$ dcecp -c group create dfsteam -inprojlist yes
```

The following command is an example of how to add the userid **cell_admin** and the group **dfsteam** to the **admin.bos** file on a system with DCE host name *dcehostname*:

```
$ bos addadmin -server /./hosts/dcehostname \  
              -adminlist admin.bos -principal cell_admin -group dfsteam \  
              -createlist
```

Later you are instructed when to add additional administrators and update the DFS administrative list files on the DCE host system where the **upserver** is running.

5. Enable boserver DFS authorization checking on a system with DCE host name *dcehostname* using the command:

```
$ bos setauth /./hosts/dcehostname on
```

After enabling DFS authorization checking using the **bos setauth** command, **boserver** DFS authorization checking can be enabled and disabled while the **DFS** is running using the **bos setauth** on command or creating the **NoAuth** file. Note that DFS authorization checking is re-enabled each time the **boserver** is restarted.

6. Insure that **boserver** authorization checking is enabled on a system with DCE host name *dcehostname* using the command:

```
$ bos status /./hosts/dcehostname
```

The **NoAuth** file should subsequently be created and deleted using the **bos setauth** command. For detailed information on the **bos** commands, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

After adding DFS administrators to the **admin.bos** file and enabling **boserver** DFS authorization checking, a DFS administrator can make additional updates to the **admin.bos** administrative list file using the same method employed for the other administrative list files. Refer to “Adding Principals and Groups to Administrative Lists” on page 89 for details on creating and updating the DFS administrative list files after initially establishing the **admin.bos** list.

Disable/Enable DFS Authorization (NoAuth File)

During the initial installation of DFS on a host system, an empty **/opt/dfslocal/var/dfs/admin.bos** file is created when the **boserver** is first started if the file does not exist. Additionally, each time the **boserver** is started, DFS authorization is enabled.

Before DFS administrators are defined, you can not issue a **bos setauth off** command to disable DFS authorization nor can you update the **admin.bos** file to define DFS administrators using the **bos addadmin** command. So, after the **boserver** is first started, you can disable DFS authorization by creating the **/opt/dcelocal/var/dfs/NoAuth** as described in “Creating the Initial DFS Administrator(s)” on page 87

After initial DFS administrators are defined, the **bos setauth** can be used to disable and enable DFS authorization.

The **/opt/dcelocal/var/dfs/NoAuth** file is created as described in “Creating the Initial DFS Administrator(s)” on page 87 or when a **bos setauth off** command is successful. The **/opt/dcelocal/var/dfs/NoAuth** file is deleted each time the **boserver** is started or when a **bos setauth on** command is successful.

Creating the admin.bos File

Starting the BOS Server **boserver** for the first time on a DCE host system creates an empty **/opt/dfslocal/var/dfs/admin.bos** file if the file does not exist. But, the **admin.bos** file can not be updated to define the DFS administrators unless DFS authorization is disabled. During the initial configuration of DFS, you should follow the instructions described in “Creating the Initial DFS Administrator(s)” on page 87.

Adding Principals and Groups to Administrative Lists

After the **admin.bos** file is created and updated to establish initial DFS administrators as described in “Creating the admin.bos File,” you can create the other administrative list files and add administrators or groups to them by:

1. Logging into DCE as a DFS administrator.

The DFS administrator must be included in the **admin.bos** list on the machine on which the administrative list to be affected is located.

2. Verify that you have the necessary privilege to issue the command. You must be included in the **admin.bos** list on the machine on which the administrative list to be affected is located. If necessary, issue the **bos lsadmin** command to check the **admin.bos** list.

For example, the following command lists the members of the **admin.bos** file on the DCE host name of **fs1** in the DCE cell name of **abc.com**. The administrative list contains two users, a server machine, and two groups, all of which are from the local cell.

```
$ bos lsadmin -server ../../abc.com/hosts/fs1 -adminlist admin.bos
```

```
Admin Users are: user: jones, user: smith,  
user: hosts/fs1/self, group: dfs-admin, group: fs1-admin
```

For detailed information, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

3. Issue the **bos addadmin** command to create an administrative list file if it does not exist and add principals, groups, or both to the administrative list:

```
$ bos addadmin -server machine -adminlist filename [-principal name...] [-group name...] [-createlist]
```

- The **-server machine** option identifies the DCE host machine where the administrative list file should be updated. Note that there is an **upclient** running on this DCE host machine, and the **upserver** is running on a different DCE host machine, you should update the administrative list file on the DCE host machine where the **upserver** is running. Refer to “System Control Machines and Domains” on page 85 for more information.
- The **-adminlist filename** option specifies the name of the administrative list to which principals and groups are to be added. The default directory for the administrative lists is the configuration directory (**/opt/dcelocal/var/dfs**). If the lists are stored in the default directory, you need to provide only the specific file name (**admin.fl**, **admin.ft**, **admin.up**, **admin.bos**, or **admin.bak**). If

the lists are stored elsewhere, you must enter the path name that was used when the specific process was started.

- The **-principal** *name* option specifies the principal name of each user or server machine to be added to the list. A user from the local cell can be specified by a full or abbreviated principal name (for example, */../cellname/username* or just *username*); a user from a foreign cell can be specified only by a full principal name. A server machine from the local cell can be specified by a full or abbreviated principal name (for example, */../cellname/hosts/hostname/self* or just *hosts/hostname/self*); a server machine from a foreign cell can be specified only by a full principal name.
- The **-group** *name* option specifies the name of each group to be added to the list. A group from the local cell can be specified by a full or abbreviated group name (for example, */../cellname/group_name* or just *group_name*); a group from a foreign cell can be specified only by a full group name.
- The **-createlist** option specifies that the administrative list indicated with **-adminlist** is to be created if it does not already exist. Any principals or groups specified with the command are added to the new file; if no principals or groups are specified, the command creates an empty file. This option has no effect if the specified file already exists.

Example of Updating the admin.bos File

Before you are able to issue any **bos** commands, the **/opt/dfslocal/var/dfs/admin.bos** file must be created and updated to include the user principals and groups that have the authority to issue these commands.

If you are initially configuring DFS, the **admin.bos** file is created earlier during the initial configuration process. But, if it does not exist, see “Creating the admin.bos File” on page 89.

The following command is an example of adding the user **cell_admin** and the group **dfsteam** to the **admin.bos** file on a system with the DCE host name of *dcehostname*:

```
$ bos addadmin -server /../hosts/dcehostname -adminlist admin.bos \  
-principal cell_admin -group dfsteam -createlist
```

Example of Updating the admin.fl File

Before you are able to issue any commands that update the Fileset Location Database (FLDB), you must create the **/opt/dfslocal/var/dfs/admin.fl** file and update it to include the user principals, groups and other DFS server machine principals that have the authority to issue these commands or communicate with the **flserver** on the host system where DFS is being configured.

The following command is an example of adding the user **cell_admin**, the abbreviated server principal for the system with a DCE host name of *dcehostname* and the group **dfsteam** to the **admin.fl** file on the system with a DCE host name of *dcehostname*:

```
$ bos addadmin -server /../hosts/dcehostname -adminlist admin.fl \  
-principal cell_admin hosts/dcehostname/dfs-server \  
-group dfsteam -createlist
```

Note that the abbreviated server principals for all the systems in the DCE cell where DFS is running should be added to the **admin.fl** file on each system in the DCE cell where an **flserver** is running.

Example of Updating the admin.ft File

Before you are able to issue any **fts** commands, you must create the **/opt/dfslocal/var/dfs/admin.ft** file and update it to include the user principals and groups that have the authority to issue these commands.

The following command is an example of adding the user **cell_admin**, the abbreviated server principal for the system with a DCE host name of *dcehostname* and the group **dfsteam** to the **admin.ft** file on the system with a DCE host name of *dcehostname*:

```
$ bos addadmin -server /./hosts/dcehostname -adminlist admin.ft \  
               -principal cell_admin hosts/dcehostname/dfs-server \  
               -group dfsteam -createlist
```

Note that the abbreviated server principals for all the systems in the DCE cell where DFS is running should be added to the **admin.ft** file on each system in the DCE cell where an **ftserver** is running.

Example of Updating the admin.bak File

Before you are able to issue any **bak** commands, you must create the **/opt/dfslocal/var/dfs/admin.bak** file and update it to include the user principals, groups, and DFS server machine principals that have the authority to issue these commands or communicate with the **bakserver** on the host system where DFS is being configured.

The following command is an example of adding the user **cell_admin**, the abbreviated server principal for the system with a DCE host name of *dcehostname* and the group **dfsteam** to the **admin.bak** file on the system with a DCE host name of *dcehostname*:

```
$ bos addadmin -server /./hosts/dcehostname -adminlist admin.bak \  
               -principal cell_admin hosts/dcehostname/dfs-server \  
               -group dfsteam -createlist
```

Note that the abbreviated server principals for all the systems in the DCE cell where DFS is running should be added to the **admin.bak** file on each system in the DCE cell where a **bakserver** or **butc** server is running.

Example of Updating the admin.up File

Before you are able to run an **upserver** on a host system, you must create the **/opt/dfslocal/var/dfs/admin.up** file and update it to include the DFS server principal for each **upclient** server or machine that can obtain information from the **upserver**.

The **upclient** and **upserver** relationship defines a domain. The system where the **upserver** runs is the System Control machine for the domain.

The following command is an example of adding the the DFS server principal for an **upclient** on a system with the DCE host name of *dcehostname2* to the **admin.up** file on the system with the DCE host name of *dcehostname*, where the **upserver** is being configured:

```
$ bos addadmin -server /./hosts/dcehostname -adminlist admin.up \  
               -principal hosts/dcehostname2/dfs-server
```

Mapping OS/390 User IDs for HFS and RFS Access

For authorized HFS and RFS access on the host system for the DFS servers, DFS administrators and users must be registered with the OS/390 security subsystem on the host system for the DFS servers. There must also be a defined relationship (a **mapping**) between their DCE user IDs and their OS/390 user IDs.

Included in this and the following sections are topics related to setting up DFS administrator and user access to HFS and RFS filesets which include:

- “Registering Users With the OS/390 Security Subsystem”
- “Mapping DCE User IDs to OS/390 User IDs”
- “Using the Identity Mapping File Method” on page 93
- “Using the RACF Identity Mapping Function” on page 97
- “DFS Anonymous User Considerations” on page 98.

Note: Although RACF is discussed in the following sections, any OS/390 external security manager (ESM) that has equivalent support can be used instead of RACF.

Registering Users With the OS/390 Security Subsystem

Clients of DFS may access HFS or RFS filesets exported by OS/390 DFS servers. Access to these resources is controlled by the local OS/390 security subsystem (for example, RACF). Thus, all users of the DFS must be registered with the specific security subsystem on the OS/390 host where DFS servers are running and exporting HFS or RFS filesets. Refer to the documentation provided by the Security software on your host for information on registering users on the local security subsystem.

Mapping DCE User IDs to OS/390 User IDs

The DFS server enables DFS clients to access non-DCE Local File System filesets.

The local security subsystem determines if the client is authorized to use the resource. Because the local security subsystem is not integrated with the DCE Security service, the local security subsystem cannot recognize DCE user IDs.

The administrator has to establish a relationship (that is, a **mapping**) between a user's DCE user ID and OS/390 user ID. This section describes methods that OS/390 DFS may use to map DCE user IDs to OS/390 user IDs. The methods are:

1. Mapping the DCE user IDs to the local OS/390 user IDs using the **mapid** utility.

Note: The mapping procedure in OS/390 DFS is identical with that used for OS/390 DCE Application Support.

Refer to “Using the Identity Mapping File Method” on page 93 for details.

2. Mapping the DCE user IDs to the local OS/390 user IDs using the identity mapping function of the external security manager (ESM) in use on the host system.

Note: Although RACF is discussed in this chapter, any OS/390 external security manager (ESM) that has equivalent support can be used instead of RACF.

Refer to “Using the RACF Identity Mapping Function” on page 97 for details.

3. Optionally establishing a default user ID for unauthenticated access to HFS and RFS filesets.

Refer to “DFS Anonymous User Considerations” on page 98 for details.

Note: In some cases, the DFS client request is denied if the DFS server cannot determine a mapping to an OS/390 user ID. For example, if the user is unauthenticated or not mapped or mapped but not in RACF and if `_IOE_MVS_DFSDFLT` is not specified or the `_IOE_MVS_DFSDFLT` user ID is not in RACF the mapping function denies the request.

Refer to Appendix C, “Environment Variables in DFS” on page 153 for a description of `_IOE_MVS_DFSDFLT`.

Using the Identity Mapping File Method

To use the Mapping Identity File method to map the DCE user ID to the OS/390 user ID, follow the procedures that are described in:

1. “Creating the Identity Mapping Input File (dfsidmap)”
2. “Creating the Identity Mapping File” on page 95
3. “Setting the `_IOE_MVS_IDMAP` Environment Variable” on page 96
4. “Ensuring the `_IOE_MVS_IDMAP_SAF` Environment Variable Is Set NO” on page 96
5. “Modifying and Deleting Identity Mapping Entries” on page 97
6. “Allowing Foreign Users Access to the DFS Server” on page 97
7. “Authorizing Access to the Identity Mapping File” on page 97.

Creating the Identity Mapping Input File (dfsidmap)

The Identity Mapping input file (**dfsidmap**) is a text file that the administrator creates and maintains. This must be created as an HFS file.

Note: The DFS server administrator must also map his DCE user ID to his OS/390 user ID.

The Identity Mapping input file (**dfsidmap**) contains one or more identity mapping declarations and has the following general format:

```
DCE-user-ID1
OS/390-user-ID1
DFS-server-name
DCE-user-ID2
OS/390-user-ID2
...
```

This format illustrates the two types of entries that can exist on the Identity Mapping input file (**dfsidmap**). The first entry has three elements: DCE user ID, OS/390 user ID, and DFS server name. The second entry only has two elements: DCE user ID and OS/390 user ID. These are explained later in this chapter. A blank line is required between entries. The following list explains each element in an identity mapping entry:

<i>DCE-user-ID</i>	Is the client's DFS identity. This may either be a simple DCE principal name (for clients within the cell) or a fully qualified global name (for clients within and outside the cell). The DCE user ID can be up to 256 characters in length.
<i>OS/390-user-ID</i>	Is the OS/390 user ID of the client. All potential DFS server clients must have user IDs on the OS/390 host where the DFS server is running.
<i>DFS-server-name</i>	Is the name of the CDS object that represents a specific DFS server. This is optional. If the mapping entry includes the name of a DFS server, the mapping is valid only for that DFS server. If the entry does not include the name of a DFS server, the mapping is valid for all servers that are running on the host system that use the Identity Mapping input file (dfsidmap) (for example, Application Support for

Information Management System servers). The *DFS-server-name* for a DFS server is specified in the **_IOE_MVS_SERVER** environment variable in the **dfskern** process.

The first entry type (with three elements) is known as the **specific mapping** to a DFS server. The second entry type (with two elements) is known as the **default mapping** for a DFS server user. You can have one or both of these entry types in your Identity Mapping input file (**dfsidmap**).

Although not required, it is recommended that all servers that require identity mapping on a host system share the same Identity Mapping input file (**dfsidmap**) (for example, any OS/390 DCE Application Support Servers should share the same Identity Mapping input file with DFS servers). Figure 7 shows a sample Identity Mapping input file (**dfsidmap**).

```
smith
CMSMITH
/./subsys/dce/dfs

jones
TSJONES
```

Figure 7. Example Identity Mapping Input File (dfsidmap)

In the first entry, the DCE user ID **smith** is mapped to the OS/390 user ID **CMSMITH**. This mapping is effective only when accessing the DFS server whose CDS name is **/./subsys/dce/dfs**. In the second entry, the DCE user ID **jones** is mapped to the OS/390 user ID **TSJONES**. This mapping is effective when accessing **any** server on the host system.

If a user has both specific mapping and default mapping entries with the same principal in the input file, the specific mapping entry overrides the default mapping entry. Consider an Identity Mapping input file (**dfsidmap**) that has the following entries:

```
smith
CMSMITH

smith
TSJOHN
/./subsys/dce/dfs
```

The second entry (specific) overrides the first entry.

Each user can only have one default mapping for every DCE principal in the file. For example, the following mapping entries are **not** allowed:

```
smith
CMSMITH

smith
TSJOHN
```

Also, each user can only have one specific mapping entry to a particular server for every DCE principal in this file. For example, the following entries are **not** allowed:

```
smith  
CMSMITH  
/./subsys/dce/dfs
```

```
smith  
TSJOHN  
/./subsys/dce/dfs
```

Creating the Identity Mapping File

The **Identity Mapping File** is a binary file that is created by running the **mapid** program on the Identity Mapping input file (**dfsimap**). The Identity Mapping file is created as an HFS file.

The **mapid** program is run with two parameters as follows:

mapid *input-file output-file*

where:

- | *input-file* Is the HFS pathname of the Identity Mapping input file (**dfsimap**).
- | *output-file* Is the HFS pathname of the Identity Mapping output file.

The **mapid** program can be run from TSO, the OS/390 shell, or in batch. For example, you can enter the following from TSO:

```
| mapid "/opt/dfslocal/home/dfskern/dfsimap" "/opt/dfslocal/home/dfskern/mapid.output"
```

Important Note to Users

This section includes information on using the **mapid** program. The **mapid** program is run as a batch job, from the TSO/E command line, or from the OS/390 shell. The xxx.SIOESAMP(MAPID) member (where xxx is installation dependent) has sample JCL for running this program in batch.

For information on running **mapid** from TSO/E, or from the shell, refer to *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Figure 8 on page 96 shows an example JCL that is used to run this program.

```

/*JOB CARD...
/*
/******
//MAPID      EXEC PGM=IOEMAPID,REGION=0M,TIME=1440,
//          PARM=(' /DD:INFILE DD:OUTFILE')
/******
/* * Parameters
/******
I //INFILE   DD PATH='/opt/dfslocal/home/dfskern/dfsimap',
//          PATHMODE=(SIRWXU,SIRGRP,SIXGRP),
//          PATHOPTS=(ORDONLY),PATHDISP=(KEEP)
I //OUTFILE  DD PATH='/opt/dfslocal/home/dfskern/mapid.output',
//          PATHMODE=(SIRWXU,SIRGRP,SIXGRP),
//          PATHOPTS=(ORDWR,OCREAT),PATHDISP=(KEEP)
//*****
//*****
//SYSOUT    DD SYSOUT=*
//SYSERR    DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//CEEDUMP   DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*

```

Figure 8. MAPID Program JCL

Note: The Identity Mapping File must be regenerated every time the Security server host is reconfigured in the cell.

Setting the `_IOE_MVS_IDMAP` Environment Variable

The `_IOE_MVS_IDMAP` environment variable must be set to the name of the Identity Mapping Output file used by the DFS server. The declaration of this environment variable can be made in the **envar** file of the **dfskern** process located in `/opt/dfslocal/home/dfskern/envar`. (For other ways of declaring environment variables, see *OS/390 DCE Administration Guide*.)

For example, if the HFS pathname of the Identity Mapping Output file is

```

I /opt/dfslocal/home/dfskern/mapid.output, this variable is set by the following entry in the envar file:
I _IOE_MVS_IDMAP=/opt/dfslocal/home/dfskern/mapid.output

```

You can choose to have one Identity Mapping File for each server, or a single Identity Mapping File for all servers that require identity mapping (for example, any OS/390 DCE Application Support Servers should share the same Identity Mapping file with DFS servers). In both cases, the declaration of the `_IOE_MVS_IDMAP` environment variable can be made in the **envar** file of the **dfskern** process.

Ensuring the `_IOE_MVS_IDMAP_SAF` Environment Variable Is Set NO

The `_IOE_MVS_IDMAP_SAF` environment variable must be set to **NO** or not specified in the **dfskern** process to enable Identity Mapping.

Note: The environment variables for the **dfskern** process are only read on start-up of the **dfskern** process (not as a result of the **modify dfs,send dfskern,reload,idmap** operator command).

Modifying and Deleting Identity Mapping Entries

- | Edit the Identity Mapping input file (**dfsimap**) to modify or delete mapping entries. Then run the **mapid** program on the Identity Mapping input file (**dfsimap**) to generate the new Identity Mapping file. For these changes to take effect, you have to either restart the DFS server or reload the Identity Mapping Output file by using the **modify dfs,send dfskern,reload,idmap** command. See the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information on the **modify** command.

Allowing Foreign Users Access to the DFS Server

- | If there are DFS server users from other cells (that is, **foreign cells**), the fully-qualified DFS name of each foreign principal must be mapped to an OS/390 user ID. To define this mapping, the foreign principals are entered in the Identity Mapping Input File (**dfsimap**) that is used by the **mapid** utility to generate the binary Identity Mapping Output File. The **mapid** utility must translate the foreign principal name to its corresponding UUID. Hence, it must read the mapping from the foreign cell. To be able to perform this, the administrator running **mapid** must have authority to read the **./sec/principal** directory in the foreign cell.

This is accomplished by having the DFS administrator add a **foreign_user** entry for the administrator (who runs **mapid**) in the ACL of the **./sec/principal** directory of the foreign cell. This entry must give this administrator read permission to this directory. For example, if the DFS server belongs to the cell **./cell1.tid.ibm.ca (/.../c=ca/o=ibm/ou=tid/cn=cell1)**, and the principal name of the administrator is **cell_admin**, the DCE control program administration interface (**dcecp** session) performed on the foreign cell is as follows:

```
$ dcecp
dcecp> acl modify ./sec/principal -add {foreign_user /.../c=ca/o=ibm/ou=tid/cn=cell1/cell_admin r}
dcecp> exit
```

- | **Authorizing Access to the Identity Mapping File:** The Identity Mapping file serves as the link between the DCE Security service and the local security subsystem. Access to the Identity Mapping Input file (**dfsimap**) and the resulting Identity Mapping Output binary file must be restricted to authorized administrators only.

The DFS server must also be given read access to the Identity Mapping file that it uses. The administrator must be given read and write access to the Identity Mapping files.

Using the RACF Identity Mapping Function

To use the RACF Identity Mapping Function method to map the DCE user ID to the OS/390 user ID, follow the procedures that are described in:

1. "Creating the RACF DCEUUIDS Profile"
2. "Creating the RACF DCEUUIDS APPLDATA OS/390 ID Entries" on page 98
3. "Ensuring _IOE_MVS_IDMAP_SAF Environment Variable is Set YES" on page 98
4. "Modifying and Deleting RACF DCEUUIDS APPLDATA OS/390 ID Entries" on page 98
5. "Modifying and Deleting RACF DCEUUIDS APPLDATA OS/390 ID Entries" on page 98.

Creating the RACF DCEUUIDS Profile

The RACF DCEUUIDS profile is a profile in the RACF database that allows you to store mappings of DCE user IDs to OS/390 user IDs. It is created with the following command:

```
RDEFINE FACILITY IRR.RDCERUID UACC(NONE)
PERMIT IRR.RDCERUID CLASS(FACILITY) ID(DFS1) ACCESS(READ)
SETROPTS CLASSACT(DCEUUIDS)
```

Creating the RACF DCEUUIDS APPLDATA OS/390 ID Entries

RACF DCEUUIDS APPLDATA MVS ID entries specify the OS/390 user ID that the DCE cell and principal should be mapped to. RACF DCEUUIDS APPLDATA OS/390 ID entries are created with the following command:

```
RDEFINE DCEUUIDS cell_uuid.principal_uuid APPLDATA('os/390_id')
```

Ensuring _IOE_MVS_IDMAP_SAF Environment Variable is Set YES

The **_IOE_MVS_IDMAP_SAF** environment variable must be set to **YES** to ensure that the RACF identity mapping facility is used. It must be set in the **dfskern** process **envvar** file.

Note: The environment variables for the **dfskern** process are only read on start-up of the **dfskern** process (not as a result of the **modify dfs,send dfskern,reload,idmap** operator command).

Modifying and Deleting RACF DCEUUIDS APPLDATA OS/390 ID Entries

RACF DCEUUIDS APPLDATA OS/390 ID entries may be modified by using the following command:

```
RALTER DCEUUIDS cell_uuid.principal_uuid APPLDATA('os/390_id')
```

RACF DCEUUIDS APPLDATA OS/390 ID entries may be deleted by using the following command:

```
RALTER DCEUUIDS cell_uuid.principal_uuid
```

For these changes to take effect, you have to restart the DFS server, or reload the identity map (causing any cached mappings to be deleted by the **modify dfs,send dfskern,reload,idmap** command. See the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information on the **modify** commands).

DFS Anonymous User Considerations

An anonymous user ID is required for unauthenticated access to non-DCE Local File System filesets (for example, HFS or RFS filesets) exported by DFS servers running on the host system. An anonymous user ID is necessary if the DCE-principal-to-OS/390 ID mapping fails or if a user has not completed the **dce_login** process and is, therefore, unauthenticated. If an anonymous user ID is not defined, unauthenticated access to HFS or RFS filesets fails.

The DFS anonymous user ID can be created at this time to allow unauthenticated access to HFS or RFS filesets. If the anonymous user ID is not created at this time, it can be added later.

The following example shows how you can create the DFS anonymous user ID by the Resource Access Control Facility (RACF) using TSO/E:

```
ADDUSER dfsdf1t DFLTGRP(df1tgroup) OWNER(owner) NAME('DFS DEFAULT ID') \
  OMVS(UID(df1tuid) HOME(/u))
```

Once created, you must specify the ID in the environmental variable file for **dfskern**. To do so, enter the following, where **DFSDFLT** is the default ID for the DFS anonymous user ID you have created:

`_IOE_MVS_DFSDFLT=DFSDFLT`

Note that you must specify an OS/390 Unique Identifier (UID) segment and its group **must** have an OS/390 Group Identifier (GID) segment.

Chapter 9. Exporting Data in DFS

Exporting data in DFS requires you to:

1. Create DFS aggregates for the filesets or data sets to be exported by updating the **/opt/dcelocal/var/dfs/devtab** file.
For DCE Local File System aggregates, you must also allocate VSAM linear data sets and format them so that they can be used for DFS filesets.
2. Export the aggregates by updating the **/opt/dcelocal/var/dfs/dfstab** file.
3. Create the first fileset **root.dfs** for the DCE cell if it does not already exist.
4. Create mount points in the DFS namespace for other filesets to be exported by DFS from this host system.

This chapter describes the procedures for exporting data in OS/390 DFS. Topics included in this chapter are:

- “Creating Aggregates in DFS”
- “Exporting File System Data Using DFS” on page 103
- “Setting Up the root.dfs Fileset” on page 106
- “Creating Mount Points for Filesets” on page 109
- “Steps to Create and Export a DCE Local File System Aggregate” on page 110
- “Steps to Create and Export an HFS Aggregate” on page 113
- “Steps to Create and Export an RFS Aggregate” on page 115.

Creating Aggregates in DFS

To make data available from the DFS file server you have just configured (see “Using DFSCONF to Configure the DFS File Server” on page 58), you must export data and create filesets from a File Server using either OS/390 Hierarchical File System (HFS), Record File System (RFS), or DCE Local File System filesets.

The following sections provide information on how to properly create aggregates for HFS, RFS, and DCE Local File System filesets and how to export the aggregates.

Creating Non-Local File System Aggregates in DFS

On OS/390 DFS, a Hierarchical File System (HFS) aggregate is the equivalent of a single HFS partition. The HFS aggregate also is seen as a single complete fileset by the DFS File Server. Because of this, exported HFS aggregates (non-Local File System aggregates) can contain only one fileset per partition and the logical volume of any HFS aggregate equals a single HFS partition. Before you can export an HFS data set, it must be mounted locally to the OS/390 address space. For details regarding allocating and mounting an HFS file system, see *OS/390 UNIX System Services Planning*, SC28-1890.

Once mounted, an entry must be created in the **/opt/dcelocal/var/dfs/devtab** file. This entry maps a minor device number to the HFS file system you wish to export. The minor device number is an identifier for the device you plan to mount. The number can be any integer greater than zero. Each HFS (non-Local File System) aggregate must have a unique minor device number. The minor device number need not be unique across both DCE Local File System and non-Local File System aggregates.

The following is an example of a completed **devtab** file entry. Lines beginning with an asterisk (*) are comment lines. The first entry in the file defines the type of file system, **ufs**. This defines the file system

as HFS. The minor device number, **2**, is assigned. This number is a unique identifier and becomes part of the name of the defined logical volume. The logical volume name must be specified in the **/opt/dcelocal/var/dfs/dfstab** file when exporting the HFS partition to DFS (see “Exporting Non-Local File System Data” on page 104). The last entry in the file, **omvs.user.abc**, is the name of the HFS file system you are preparing to export.

```
* HFS Devices
define_ufs 2
omvs.user.abc
```

An RFS (Record File System) aggregate is an OS/390 data set name that is either the name of a partitioned data set or is the prefix of a set of data sets and is a non-Local File System aggregate. The whole aggregate is also a single complete fileset to the DFS server.

You must add an entry in **/opt/dcelocal/var/dfs/devtab** which maps a minor device number to the RFS fileset you wish to export. For each RFS fileset add the following entries:

```
* RFS devices
define_ufs 3 rfs
USERA
```

In the previous two examples, lines beginning with an asterisk are comments. The next line defines the type of file system (**ufs**), the minor device number and whether the **ufs** file system is an **hfs** (hierarchical file system) subtype or an **rfs** (record file system) subtype. The default subtype is **hfs**. In this case, we are defining a **ufs** with a subtype of **rfs**.

Note: From now on, we will generally refer to this as an **rfs** fileset. You should keep in mind, however, that it is really a **ufs** fileset with a subtype of **rfs**. When we make comments about a **ufs** fileset, they pertain to both subtypes - **hfs** and **rfs**.

As with a DCE Local File System aggregate, the HFS fileset and the RFS fileset have a minor number describing the device, **2** in the HFS example and **3** in the RFS example. This number can be any integer greater than zero. Each **ufs** fileset must have a unique minor device number. (That is, a **ufs** fileset can have the same minor device number as an **lfs** fileset, but each **ufs** fileset, whether it is an **hfs** subtype or an **rfs** subtype, must have a unique minor device number.) Similarly, two DCE Local File System aggregates cannot have the same minor device number.

The third line in the example is the name of the HFS file system or RFS fileset (prefix) you wish to export. The third line also supports an optional parameter in addition to the data set name in the **devtab** for HFS or RFS filesets to control character data translation.

You may also specify the translation control parameter after the HFS or RFS file system name, on the same line. Valid values for the translation control parameter are:

binary	Do not translate data.
text	Converts incoming data from ASCII ISO 8859-1 to the local OS/390 code page. Converts outgoing data from the local OS/390 code page to ASCII ISO 8859-1.

The following example demonstrates the use of the translation control parameter. In this example, the fileset identified as **omvs.user.abc** will not have data translated.

```
* HFS Devices
define_ufs 2
omvs.user.abc binary
```

If the translation control parameter is omitted the default is controlled by the **_IOE_HFS_TRANSLATION** environment variable setting (for HFS filesets) and the **_IOE_RFS_TRANSLATION** environment variable setting (for RFS filesets) in the **dfskern** process, see the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information.

Finally, the **attrfile** *attributes_file* parameter (where *attributes_file* is the name of the attributes file that controls the data set creation, processing, and site attributes for this RFS fileset) is supported on the same line as the data set name in the **devtab** for RFS filesets. The default attributes file for RFS filesets is specified in the **_IOE_RFS_ATTRIBUTES_FILE** environment variable specified in the **dfskern** process.

Note: The use of the translation control parameter overrides the **_IOE_HFS_TRANSLATION** environment variable and the **_IOE_RFS_TRANSLATION** environment variable settings for this specific fileset.

The logical volume defined is **/dev/ufs2**. This unique logical volume name should be used when specifying the device in the **dfstab** file (located in the **/opt/dcelocal/var/dfs** directory) for exporting to DFS.

Creating DCE Local File System Aggregates in DFS

In OS/390 DFS, DCE Local File System aggregates are made up of one or more formatted linear data sets (LDS). The data sets for each aggregate are specified in the **/opt/dcelocal/var/dfs/devtab** file and make up a logical volume. The logical volume consists of one or more LDSs that have been formatted to create a single aggregate that will be exported by the **dfskern** process. To define a volume, edit the **/opt/dcelocal/var/dfs/devtab** file.

The following example shows a complete entry in the **devtab** file for a DCE Local File System logical volume. Lines preceded by an asterisk (*) are comments. The type of file system, **lfs**, and the minor device number 1 are defined. The minor device number can be any integer greater than zero. Each logical volume should have a unique minor device number. The volume's minor device number will become part of the aggregate's identifying device name. The minor device number need not be unique across both DCE Local File System and non-Local File System aggregates. Following the definition of the logical volume are the names of three linear data sets that make up the DCE Local File System aggregate:

```
*Devtab- Example Entry for a DCE LFS Logical LFS Volume
define_lfs 1
DFS.DCELFS.AGGR001.LDS00001
DFS.DCEFFS.AGGR001.LDS00002
DFS.DCELFS.AGGR001.LDS00003
```

The logical volume defined will be known as **/dev/lfs1**, with 1 being the device's minor number. This logical volume name should be used when you specify the device in DCE Local File System utilities such as **newaggr** and **growaggr** (for detailed information regarding these utilities, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*). The logical volume name must also be specified in the **dfstab** file when exporting the volume to **DFS** (see "Exporting DCE Local File System Data" on page 105).

OS/390 Job Control Language (JCL) found in the SIOESAMP library member **NEWAGGR** can be used to both allocate LDSs and to create a DCE Local File System aggregate. The SIOESAMP JCL can be edited to reflect the correct linear data set names and logical volume. To access the **devtab** file, the program must run as **UID = 0 (root)**. After using IDC Access Method Services to allocate each Linear Data Set, the **NEWAGGR** JCL then formats the logical volume into a DCE Local File System aggregate.

Exporting File System Data Using DFS

This section discusses exporting non-Local File System data and exporting DCE Local File System data using DFS.

Exporting Non-Local File System Data

The HFS and RFS aggregates to be exported by the **dfskern** process must now be added to the **dfstab** file. This file is located in **/opt/dcelocal/var/dfs**. The **dfstab** file is used by the File Exporter, **fxd**, to determine the aggregates available for exporting. The **dfstab** file describes each aggregate and contains the aggregate device name, the aggregate name, the file system type, the aggregate id, and, for HFS and RFS aggregates, the fileset id.

Edit the **dfstab** file to add the appropriate control line describing the aggregate. For HFS aggregates, this requires an additional step. Since an HFS aggregate is a complete DFS fileset and not created by issuing the **fts** command, an entry must be made in the Fileset Location Database for the fileset. The **dfstab** entry for HFS aggregates requires that a fileset id be assigned when the Fileset Location Database entry is created. Issue the following **fts** command, **fts crfldbentry**, to create the Fileset Location Database entry for the HFS aggregate (for further information regarding this command, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*). The fileset name, *fileset*; server, *server*; and aggregate id, *aggrid*, are assigned by the DFS administrator.

```
$ fts crfldbentry -ftname fileset -server server -aggrid aggrid
```

Once the fileset entry is created, the fileset IDs assigned to the fileset's read-write, read-only, and backup filesets will be displayed. The ID for the read-write fileset must be entered in the **dfstab** file for exporting. The device name for an HFS or RFS aggregate is **/dev/ufs*n***, where *n* is the minor device number previously specified in the **devtab** file. The name is any unique name assigned by the administrator. This name **must** be unique. The export program will not mount any aggregate with a duplicate device name.

Edit the **dfstab** file. HFS or RFS entries in the **dfstab** file should always specify an aggregate type of **ufs**. The aggregate ID is a unique ID assigned by the DFS administrator. The read-write fileset ID from the Fileset Location Database entry is entered as the fileset ID. The following fields appear for each entry in the file, in the order listed. Each field must be separated by a minimum of one space or tab; each entry must be on a separate line.

Device Name

The device name of the partition, for example, **/dev/ufs2**.

Aggregate Name

The name to be associated with the exported partition. An aggregate name can contain any characters, but it can be no longer than 31 characters, and it must be different from any other aggregate name in the file. Aggregate names cannot be abbreviated, so you should choose a short, explicit name. For example, the aggregate name of a non-Local File System partition could be **hfs1**.

File System Type

The identifier for the file system type of the partition. For non-Local File System file systems, this must be **ufs**. It must be in lowercase letters.

Aggregate ID

A positive integer to serve as the aggregate ID of the exported partition. The integer must match the aggregate ID specified with the **-aggrid** option of the **fts crfldbentry** command, and it must be different from any other aggregate ID in the **dfstab** file. (If the ID is changed, fileset operations on the partition's fileset will fail.)

Fileset ID

The unique fileset ID number returned by the **fts crfldbentry** command for the fileset on the partition (for example, **0,,1715**). Use the read-write ID number (not the read-only or backup ID number) returned by the command as the value for this field.

The fileset ID is initially represented as **0,,*x***, where *x* is incremented by one until it exceeds 2^{32} . Once 2^{32} is exceeded, the ID becomes **1,,1**. When creating the root fileset, specify the

OS/390 server name and aggregate name (as specified in the **dfstab** file, see “Exporting Non-Local File System Data” and “Exporting DCE Local File System Data” on page 105) on your OS/390 system where the fileset is to reside.

The following is an example of a **dfstab** entry for an HFS aggregate and an RFS aggregate:

device	name	type	id	fileset id
/dev/ufs2	hfs1	ufs	102	0,,1715
/dev/ufs3	rfs1	ufs	103	0,,1718

If DFS is not running, you can start DFS address spaces and processes (refer to “DFS Address Space, Processes, and PROCLIBs” on page 13). If DFS is already started, issue the following command from the operator's console to begin exporting HFS data:

modify dfs,start export

A message written by **dfsexport** specifies each aggregate that you have exported and that is attached to the file system. If there are problems in exporting an aggregate, **dfsexport** issues a message identifying the aggregate that failed to attach.

If you plan to export DCE Local File System data, continue to the next section, “Exporting DCE Local File System Data.”

For further information regarding exporting data, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Exporting DCE Local File System Data

The DCE Local File System aggregates to be exported by the **dfskern** process must also be added to the **dfstab** file. This file is located in **/opt/dcelocal/var/dfs**. The **dfstab** file is used by the File Exporter, **fxd**, to determine the aggregates available for exporting. The **dfstab** file describes each aggregate and contains the aggregate device name, the aggregate name, the file system type, and the aggregate id.

Edit the **dfstab** file to add the appropriate control line describing the aggregate. DCE Local File System entries in the **dfstab** file should always specify an aggregate type of **lfs**. The minor device number, *n*, as specified in the **devtab** file should also be included (see “Creating DCE Local File System Aggregates in DFS” on page 103). The administrator assigns the aggregate name and aggregate ID. The following fields appear for each entry in the file, in the order listed. Each field must be separated by a minimum of one space or tab; each entry must be on a separate line.

Device Name

The device name of the partition, for example, **/dev/lfs1**.

Aggregate Name

The name to be associated with the exported aggregate. An aggregate name can contain any characters, but it can be no longer than 31 characters, and it must be different from any other aggregate name in the file. Aggregate names cannot be abbreviated, so you should choose a short, explicit name (for example, **lfs1**).

File System Type

The identifier for the file system type of the partition. For DCE Local File Systems, this must be **lfs**. It must be in lowercase letters.

Aggregate ID

A positive integer to serve as the aggregate ID of the exported aggregate. The integer must be different from any other aggregate ID in the **dfstab** file. (If the ID is changed, fileset operations on the partition's fileset will fail.)

The following is an example of a **dfstab** entry for a DCE Local File System aggregate:

device	name	type	id
/dev/lfs1	lfs1	lfs	6

If DFS is not running, you can start DFS address spaces and processes (refer to “DFS Address Space, Processes, and PROCLIBs” on page 13). If DFS is already started, issue the following command from the operator's console to begin exporting DCE Local File System data:

modify dfs,start export

For further information regarding exporting data in OS/390 DFS, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Setting Up the root.dfs Fileset

To make data available from the DFS file server you have just configured (see “Using DFSCONF to Configure the DFS File Server” on page 58), you must export data and create filesets from a File Server using either HFS, RFS, or DCE Local File System filesets. The first fileset you create should be **root.dfs**. The **root.dfs** fileset can be created on an OS/390 or non-OS/390 DCE Distributed File Service system. In addition, a server entry must be created for the OS/390 DFS Server (this procedure is explained in “Creating the DFS File Server FLDB Entry” on page 63).

Note: DFS must be configured on a host system that has been configured into a DCE cell. If you have already created a **root.dfs** fileset for the DCE cell, skip this section.

The main read/write fileset, **root.dfs**, is required in every DCE cell's file system. It is the first fileset created during DFS configuration on a host system in the cell.

Important Note to Users

The **root.dfs** fileset is the implied fileset for the root of a cell's DFS filespace (*/.../cellname/fs*). It can be a fileset in a DCE Local File System aggregate or in a non-Local File System Aggregate. However, it must be a fileset in a DCE Local File System aggregate, if DFS functionality such as replication is to be available in the DCE cell. Also, on OS/390, the Record File System (RFS) is not supported for **root.dfs**.

Before proceeding to allocate **root.dfs** on an OS/390 DCE host system, verify that the DFS (the DFS File Server) including the **ftserver** which is a **boserver** sub-process, has been configured and started on the system. Refer to “DFS Configuration Steps” on page 48, for more information on configuring DFS. Refer to Chapter 11, “Starting and Stopping DFS Components” on page 131, for information on starting and stopping DFS components.

Steps to Define a root.dfs Fileset

This section describes the detailed steps that must be performed to allocate, define, and export a DCE Local File System aggregate on OS/390. It also describes how to define a fileset in the DCE Local File System aggregate to be used as the **root.dfs** fileset.

Note: Defining a **root.dfs** fileset is a special case of creating and exporting a DCE Local File System aggregate for general use.

Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for further information about defining filesets and the commands issued in the following instructions.

1. Logon to OS/390 and OMVS

Logon to OS/390 and OMVS.

Optionally, logon to OS/390 and OMVS as **root** user with **UID = 0**.

Note: Unless otherwise noted, commands described in this section should be issued from OMVS.

2. **dce_login as cell_admin**

Login to DCE as a user with DCE cell administrator authorization by issuing the following command:

```
dce_login cell_admin
or
dce_login userid
```

3. **Verify that there is a server entry for the OS/390 DFS Server**

The following command displays the server entry, in this example, **dcedfs**, from the Fileset Location Database (FLDB). The command is shown as entered from an OS/390 UNIX session.

```
$ fts lsserverentry -server dcedfs
```

```
Description for site 'dcedfs':
dcedfs.endicott.ibm.com (2:0.0.9.130.79.35)
FLDB quota: 0; uses: 35: principal= hosts/DCEDFS; owner=<nil>
```

If a server entry has not yet been created, see “Creating the DFS File Server FLDB Entry” on page 63.

4. **Create VSAM Data Sets for the DCE Local File System Aggregate for root.dfs**

In order to have **root.dfs** fileset reside on OS/390, you need to allocate VSAM linear data set(s) for the logical volume that will be associated with the device name for the DCE Local File System aggregate that will contain the **root.dfs** fileset.

To do this, you can make the necessary changes to the example control statements in the installation data set member SYS1.SIOESAMP(NEWAGGR) and submit the NEWAGGR job for execution.

Note: The following **root.dfs** related examples assume that the data set **OMVS.PRIV.DCELFS.AGGR.ROOT.DFS** was created. The data set comprises the logical volume that will be associated with the device name **/dev/lfs1** and the aggregate name **lfs1**.

5. **Create a devtab File for root.dfs**

To export an aggregate, you need to create and/or update the **/opt/dfslocal/var/dfs/devtab** file. Use OMVS **OEDIT** to update the **devtab** file.

The following is an example of the contents of a **devtab** file that defines the linear data set name that comprises the logical volume associated with the **root.dfs** fileset and the device name **/dev/lfs1**.

Note: The device name **/dev/lfs1/** will be assigned the DCE Local File System aggregate name **lfs1** in the **dfstab** entry (see below).

```
* Logical Volume - DCE LFS (root.dfs) - Device Name /dev/lfs1
define_lfs 1
OMVS.PRIV.DCELFS.AGGR.ROOT.DFS
*
```

Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for further information about the **devtab** file.

6. **Create a dfstab File for root.dfs**

An **/opt/dfslocal/var/dfs/dfstab** file needs to be created and/or updated in order to export the aggregate that will contain the **root.dfs** fileset. Use OMVS **OEDIT** to update the **dfstab** file.

The example **dfstab** file entry shown below for the **root.dfs** fileset:

- Associates the device name **/dev/lfs1/** with the DCE Local File System aggregate name **lfs1**,
- Defines the aggregate type **lfs**,
- Defines the aggregate ID **1**, and
- Identifies that the aggregate should be exported when the file is read by the DFS File Server.

The example of a **dfstab** file entry for the **root.dfs** fileset is:

```
/dev/lfs1          lfs1    lfs  1
```

Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for further information about the **dfstab** file.

7. Export the root.dfs Aggregate

To enable the aggregate and its filesets to be accessed by DFS clients and used for the **root.dfs** fileset, the aggregate needs to be exported.

To export aggregates that have entries in the **devtab** and **dfstab**, issue the following command from the DFS host OS/390 system operator's console:

```
modify dfs,start export
```

Refer to Chapter 11, "Starting and Stopping DFS Components" on page 131 or the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for further information about the **modify** command for DFS.

8. Create the root.dfs Fileset

You now need to create and define the **root.dfs** fileset by issuing the following **fts create** command. The command will create the read-write fileset and create the FLDB entries for the fileset. These entries include a unique ID number assigned by the **flserver** for the read-write, read-only (for replication of DCE Local File System filesets), and backup filesets.

The following shows the syntax for creating the **root.dfs** fileset. The **-server machine** is the name of the OS/390 server where **root.dfs** resides and **-aggregate name** is the name of the aggregate (for further information regarding this command, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*).

The following example assumes that the DCE host name for the OS/390 system is **GDLCST** and creates the fileset named **root.dfs** in the aggregate named **lfs1**. A unique aggregate name (**-aggregate** specification) must be assigned by the location and is also specified in the second field of the associated **dfstab** file entry.

```
fts create -ftname root.dfs -server /./hosts/GDLCST -aggregate lfs1
```

An example of the output from issuing this command would be:

```

      readWrite  ID 0,,1  valid
      readOnly  ID 0,,2  invalid
      backup     ID 0,,3  invalid
number of sites: 1
  server          flags      aggr   siteAge principal      owner
gdlsst.endicott.ib RW        lfs1   0:00:00 hosts/GDLCST <nil>
      Fileset 0,,40 created on aggregate lfs1 of GDLCST
```

Fileset 0,,1 created on aggregate lfs1 of GDLCST

Note: Because this is the **root.dfs** fileset, it will automatically be mounted as the directory **/./**. This may take several minutes for this to take effect on your system.

9. Create the Read/Write Mount Point for root.dfs

In order to replicate **root.dfs**, an explicit read/write mount point must be created before replicating **root.dfs**. Issue the following command to create a read/write mount point for the **root.dfs** fileset:

```
fts crmount .rw root.dfs -rw
```

10. Set Replication Type for root.dfs

In order to replicate **root.dfs**, the replication type must be established. Issue the following command to accomplish this:

```
fts setreplinfo -fileset root.dfs -release
```

An example of the output from issuing this command would be:

```
fts setreplinfo: Using default value for maxage of 2:00:00
fts setreplinfo: Using derived value for failage of 1d0:00:00
fts setreplinfo: Using default value for reclaimwait of 18:00:00
```

11. Add Replication Site for root.dfs

You must add the replication site for **root.dfs** to the Fileset Location Database (FLDB). The following example can be issued for an OS/390 host system with the DCE host name of **GDLCST**:

```
fts addsite -fileset root.dfs -server ./hosts/GDLCST -aggr lfs1
```

An example of the output from issuing this command would be:

```
Added replication site GDLCST lfs1 for fileset root.dfs
```

12. Initiate Release Replication of root.dfs

Initiate replication of **root.dfs** by issuing the following command:

```
fts release -fileset root.dfs
```

An example of the output from issuing this command would be:

```
Released fileset root.dfs successfully
```

Notes:

1. After completing the definition of the **root.dfs** fileset, if you are unable to **cd** to **./.../dcecellname/fs/.rw** from the DFS client (**DFSCM**) system, issue the following commands from the DFS client:

```
cm flush
cm flushfile
cm checkf
```

2. On OS/390, if the DFS Client (**DFSCM**) was configured and started, these commands can be issued from OMVS.

Creating Mount Points for Filesets

The **root.dfs** fileset must be defined in the DCE cell's filespace in order to define the mountpoints needed to export any other filesets from a DFS File Server. Refer to "Setting Up the root.dfs Fileset" on page 106 for more information.

A DFS mount point appears and functions like a regular directory but, structurally, it is a symbolic link which indicates the name of the fileset associated with a mount point. Mount points make the contents of a fileset visible and accessible to users in the DCE namespace.

To create mount points for filesets, use the following procedure:

1. Locate and change your directory to the junction to the DFS filespace, **root.dfs**. The junction is defined at **./.../cellname/fs**

```
$ cd ./.../abc.com/fs
```

2. Create directories for related filesets such as *user* for all user data sets or *src* for source files. The following example creates a directory called **user**, a user fileset, and a mount point for the fileset. If the fileset on which the aggregate resides has been exported to the file server, you can access the data in that fileset by changing to the directory of the newly created mount point.

```
mkdir user
cd user
fts create user.abc.com lfs1 -verbose
fts crmount -fileset user.abc.com -dir user.abc
cd abc
```

3. If you plan to use the DCE Local File System replication feature for filesets, create a read-only version of the **root.dfs** fileset. This is necessary as the Cache Manager will traverse a path name to locate a file, beginning with **root.dfs**, and will access the read-only version of the fileset where possible. If *any* fileset in the path name does *not* have a read-only version, the Cache Manager will access the read-write version and will *never* access a read-only version for any filesets it traverses for the remainder of the path.

Before replicating a **root.dfs** fileset, an explicit read-write mount point must first be created. If this is not done, it becomes impossible to access the read-write version of **root.dfs**. To anticipate the possible need for read-only filesets, create a read-write mount point for **root.dfs**. In the following example, **.rw** is used as the newly created read-write mount point. The **-rw** option specifies the type of mount point as read-write.

```
$ fts crmount .rw root.dfs -rw
```

Once you have replicated **root.dfs**, create additional mount points for other filesets by specifying the read-write path as follows: In the following, *name|ID* specifies the complete name or fileset number of the fileset to be mounted:

```
$ fts crmount -fileset name|ID -dir /:/.rw/user/name|ID
```

For further information regarding mount points, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Steps to Create and Export a DCE Local File System Aggregate

This section describes how to allocate, define, and export a DCE Local File System aggregate on OS/390. It also describes how to define a fileset in the DCE Local File System aggregate.

Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for further information on defining filesets.

1. Logon to OS/390 and OMVS

Logon to OS/390 and OMVS.

Optionally, logon to OS/390 and OMVS as **root** user with **UID = 0**.

Note: Unless otherwise noted, commands described in this section should be issued from OMVS.

2. dce_login as cell_admin

Login to DCE as a user with DCE cell administrator authorization by issuing the following command:

```
dce_login cell_admin
or
dce_login userid
```

3. Create VSAM Data Sets for the DCE Local File System Aggregate for root.dfs

To create a DCE Local File System aggregate on OS/390, you need to allocate VSAM linear data set(s) for the logical volume that will be associated with the device name for the aggregate.

To do this, you can make the necessary changes to the example control statements in the installation data set member SYS1.SIOESAMP(NEWAGGR) and submit the NEWAGGR job for execution.

Note: The following examples assume that the data sets OMVS.PRIV.DCELFS.AGGR001.LDS00001 and OMVS.PRIV.DCELFS.AGGR001.LDS00002 were created for the DCE Local File System aggregate.

4. Update the devtab File for the DCE Local File System Aggregate

In order to export the aggregate, you need to update the **/opt/dfslocal/var/dfs/devtab** file. Use OMVS **OEDIT** to update the **devtab** file.

Below is an example of the contents of a **devtab** file that defines the linear data set names that makes up the logical volume associated with the device name **/dev/lfs2/**.

Note: The device name **/dev/lfs2/** will be assigned the DCE Local File System aggregate name **lfs2** in the **dfstab** (see below).

The following is an example of the contents of a **dfstab** file that defines the logical volumes for two DCE Local File System aggregates with the device names **/dev/lfs1** and **/dev/lfs2**:

```
* Logical Volume - DCE LFS (root.dfs) - Device Name /dev/lfs1
define_lfs 1
OMVS.PRIV.DCELFS.AGGR.R00T.DFS
*
* Logical Volume - DCE LFS - Device Name /dev/lfs2
define_lfs 2
OMVS.PRIV.DCELFS.AGGR001.LDS00001
OMVS.PRIV.DCELFS.AGGR001.LDS00002
```

5. Update the dfstab File for the DCE Local File System Aggregate

The **/opt/dfslocal/var/dfs/dfstab** file needs to be updated in order to export the aggregate. Use OMVS **OEDIT** to update the **dfstab** file.

An example of the **dfstab** file entry is shown below:

- Associates the device name **/dev/lfs2/** with the DCE Local File System aggregate name **lfs2**,
- Defines the aggregate type **lfs**,
- Defines the aggregate ID **2**, and
- Identifies that the aggregate should be exported when the file is read by the DFS File Server.

The following example **dfstab** file entries define two DCE Local File System aggregates (**lfs1** and **lfs2**):

```
/dev/lfs1      lfs1    lfs  1
/dev/lfs2      lfs2    lfs  2
```

6. Export the DCE Local File System Aggregate

To enable DFS clients to access the filesets in the new DCE Local File System aggregate, the DFS File Server must export the aggregate.

To export aggregates that have entries in the **devtab** and **dfstab**, issue the following command from the OS/390 system operator's console:

```
modify dfs,start export
```

Or, issue the following command on OMVS while logged on as a **root** user with **UID = 0**:

```
dfsexport lfs2
```

7. Create a Fileset for the DCE Local File System Aggregate

The following example command can be issued for a DFS host OS/390 system with the DCE host name of **GDLCSSST** to create the fileset named **episet1** in the new DCE Local File System aggregate named **lfs2**. The fileset name and aggregate must be unique in the location and is also used as the second field of the associated **dfstab** file entry.

```
fts create -ftname episet1 -server ../../hosts/GDLCSSST -aggregate lfs2 -verb
```

An example of the output from issuing this command would be:

```
----- Creating a new FLDB entry for episet1 -----
Created the FLDB entry for fileset episet1
      readWrite   ID 0,,4   valid
      readOnly   ID 0,,5   invalid
      backup      ID 0,,6   invalid
number of sites: 1
  server          flags      aggr    siteAge principal      owner

gdlcstt.endicott.ib RW          lfs1     0:00:00 hosts/GDLCSSST <nil>
```

Fileset 0,,4 created on aggregate lfs2 of GDLCSSST

8. Create a DFS Mount Point for the DCE Local File System Fileset

The following is an example of the commands that can be issued on a DFS host OS/390 system to create a mount point for the **episet1** fileset:

```
cd /:/.rw
fts crmount -fileset episet1 -dir mvsepimtp
```

9. Set Replication Type for DCE Local File System Fileset

In order to replicate the **episet1** fileset defined for the new DCE Local File System aggregate, the following command can be issued:

```
fts setrepinfo -fileset episet1 -scheduled -maxage 1h -failage 2h \
-reclaimwait 5h -minrepdelay 5m
```

An example of the output from issuing this command would be:

```
fts setrepinfo: Using given value for maxage of 1:00:00
fts setrepinfo: Using given value for failage of 2:00:00
fts setrepinfo: Using given value for reclaimwait of 5:00:00
fts setrepinfo: Using given value for minrepdelay of 0:05:00
```

10. Add Replication Site for DCE Local File System Fileset

Add the replication site for the fileset to the File Set Location Database (FLDB). The following command can be issued for a DFS host OS/390 system with the DCE host name of **GDLCSSST**:

```
fts addsite -fileset episet1 -server ../../hosts/GDLCSSST \
-aggr lfs2 -maxsiteage 30m
```

An example of the output from issuing this command would be:

```
Added replication site GDLCSSST lfs2 for fileset episet1
```

11. Initiate Schedule Replication for the DCE Local File System Fileset

Initiate replication of the fileset defined in the DCE Local File System aggregate. The following command can be issued for a DFS host OS/390 system with the DCE host name of **GDLCSSST**:

```
fts update -fileset episet1 -server ../../hosts/GDLCSSST -verbose
```

An example of the output from issuing this command would be:

```
Calling REP_UpdateSelf(gdlcsst.endicott.ibm.com, {268458673,,1888327310}
{0,,56}, ...)...
fts update: Repserver on gdlcsst.endicott.ibm.com requested to update
fileset 0,56
```

Steps to Create and Export an HFS Aggregate

This section describes how to allocate, define, and export an HFS data set as a DFS aggregate on OS/390.

Note: An HFS data set defined as a DFS aggregate contains only a single DFS fileset.

Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for further information about defining filesets and the commands issued in the following instructions.

1. Logon to OS/390 and OMVS

Logon to OS/390 and OMVS.

Optionally, logon to OS/390 and OMVS as **root** user with **UID = 0**.

Note: Unless otherwise noted, commands described in this section should be issued from OMVS.

2. dce_login as cell_admin

Login to DCE as a user with DCE cell administrator authorization by issuing the following command:

```
dce_login cell_admin
or
dce_login userid
```

3. Create a Data Set

In order to export an HFS data set, you need to allocate a data set of type **HFS** that will be defined as a DFS aggregate.

Or, if an HFS data set already exists, then you can export it.

The examples in this section assume that the name of the HFS data set name used for the DFS aggregate is **OMVS.PRIV.HFS.DATASET**.

4. Update the devtab File for the HFS Aggregate

To export the HFS data set as a DFS aggregate, you need to update the **/opt/dfslocal/var/dfs/devtab** file. Use OMVS **OEDIT** to update the **devtab** file.

The following is an example of the contents of a **devtab** file that defines two DCE Local File System logical volumes and the HFS logical volume **/dev/ufs010** that includes the data set

OMVS.PRIV.HFS.DATASET:

```
* Logical Volume - DCE LFS (root.dfs) - Device Name /dev/lfs1
define_lfs 1
OMVS.PRIV.EPISODE.AGGR.ROOT.DFS
*
* Logical Volume - DCE LFS - Device Name /dev/lfs2
define_lfs 2
OMVS.PRIV.EPISODE.AGGR001.LDS00001
OMVS.PRIV.EPISODE.AGGR001.LDS00002
*
* Logical Volume - HFS - Device Name /dev/ufs010
define_ufs 10
OMVS.PRIV.HFS.DATASET
```

5. Create the FLDB Entry for the HFS Aggregate

Create the Fileset Location Database (FLDB) entry for the HFS aggregate by issuing an **fts crfldbentry** command. The **fts crfldbentry** command assigns the fileset name and creates the fileset and FLDB entry for the HFS aggregate. It also assigns a unique aggregate ID number that must be used as the fourth field of the associated **dfstab** file entry.

Following is an example of the **fts crfldbentry** command when DFS is on an OS/390 system with a DCE host name of **GDLCSSST**, the fileset name **OMVS.PRIV.HFS.DATASET** is selected for the HFS aggregate, and the aggregate ID **10** is specified by the **-aggrid** value. The fileset name (**-ftname** specification) is assigned by the location but the use of the HFS data set name is recommended as shown in the example.

```
fts crfldbentry -ftname OMVS.PRIV.HFS.DATASET \
               -server ./:/hosts/GDLCSSST -aggrid 10
```

An example of the output from issuing this command would be:

```
      readWrite  ID 0,,22  valid
      readOnly  ID 0,,23  invalid
      backup    ID 0,,24  invalid
number of sites: 1
  server      flags      aggr  siteAge principal      owner
gdlsst.endicott.ib RW      10    0:00:00 hosts/GDLCSSST <nil>
```

FLDB entry created for fileset hfsset1 (0,,22) on aggregate 10 of GDLCSSST

Note: The ID number for the **readWrite** entry (0,,22) shown above is the number that is to be used in the **dfstab** file entry for the HFS fileset (see step 6, **Update the dfstab File for the HFS Aggregate**).

6. Update the dfstab File for the HFS Aggregate

The **/opt/dfslocal/var/dfs/dfstab** file must be updated to export the fileset assigned to the HFS aggregate. Use OMVS **OEDIT** to update the **dfstab** file.

The following is an example of the contents of the **dfstab** showing the HFS fileset entry with device name **/dev/ufs010**, aggregate name **ufs10**, and the fileset ID number **0,,22**:

```
/dev/lfs001      lfs1      lfs  1
/dev/lfs002      lfs2      lfs  2
/dev/ufs010      ufs10     ufs  10    0,,22
```

7. Create an MVS Mount Point for the HFS Aggregate

If you want to access the HFS data set used for the DFS aggregate from OMVS, create a mount point for it and mount it in the local file system as you would for any other HFS data set.

8. Export the HFS Aggregate

In order to allow the DFS client to access the HFS aggregate, the aggregate needs to be exported. In order to export aggregates that have entries in the **devtab** and **dfstab**, issue the following command from the operator's console:

```
modify dfs,start export
```

Or, issue the following command from OMVS while logged on as a **root** user with **UID = 0**.

```
dfsexport ufs10
```

9. Create a DFS Mount Point for the (HFS) Fileset

In order to create a DFS mount point for the fileset assigned to the HFS aggregate, an **fts crmount** command must be issued.

The following is an example of the commands that can be used to mount the HFS fileset named **OMVS.PRIV.HFS.DATASET** at the mount point **/:.rw/mvshfsmtpt**:

```
cd /:/.rw
fts crmount -fileset OMVS.PRIV.HFS.DATASET -dir mvshfsmtpt
```

Steps to Create and Export an RFS Aggregate

This section describes how to define and export an RFS data set as a DFS aggregate on OS/390.

Note: An RFS data set defined as a DFS aggregate contains only a single DFS fileset.

Refer to the *OS/390 Distributed File Service DFS Administration Guide and Reference*, for further information about defining filesets and the commands issued in the following instructions.

1. Logon to OS/390 and OMVS

Logon to OS/390 and OMVS.

Optionally, logon to OS/390 and OMVS as **root** user with **UID = 0**.

Note: Unless otherwise noted, commands described in this section should be issued from OMVS.

2. dce_login as cell_admin

Login to DCE as a user with DCE cell administrator authorization by issuing the following command:

```
dce_login cell_admin
or
dce_login userid
```

3. Update the devtab File for the RFS Aggregate

To export the HFS data set as a DFS aggregate, you need to update the **/opt/dfslocal/var/dfs/devtab** file. Use OMVS **OEDIT** to update the **devtab** file.

The following is an example of the contents of a **devtab** file that defines two DCE Local File System logical volumes, an HFS logical volume and the RFS logical volume **/dev/ufs020** that includes the data set **OMVS.PRIV.RFS.DATASET**:

```
* Logical Volume - DCE LFS (root.dfs) - Device Name /dev/lfs1
define_lfs 1
OMVS.PRIV.EPISODE.AGGR.ROOT.DFS
*
* Logical Volume - DCE LFS - Device Name /dev/lfs2
define_lfs 2
OMVS.PRIV.EPISODE.AGGR001.LDS00001
OMVS.PRIV.EPISODE.AGGR001.LDS00002
*
* Logical Volume - HFS - Device Name /dev/ufs010
define_ufs 10
OMVS.PRIV.HFS.DATASET
*
* Logical Volume - RFS - Device Name /dev/ufs020
define_ufs 20 rfs
OMVS.PRIV.RFS.DATASET
```

4. Create the FLDB Entry for the RFS Aggregate

Create the Fileset Location Database (FLDB) entry for the RFS aggregate by issuing an **fts crfldbentry** command. The **fts crfldbentry** command assigns the fileset name and creates the fileset and FLDB entry for the HFS aggregate. It also assigns a unique aggregate ID value that must be used as the fourth field in the associated **dfstab** file entry.

Following is an example of the **fts crfldbentry** command when DFS is on an OS/390 system with a DCE host name of **GDLCST**, the fileset name **-ftname OMVS.PRIV.RFS.DATASET** is selected for the RFS aggregate, and the aggregate ID **-aggrid 20** is assigned. The fileset name (**-ftname** specification) is assigned by the location but the use of the HFS data set name is recommended as shown in the example.

```
fts crfldbentry -ftname OMVS.PRIV.RFS.DATASET \
               -server /./hosts/GDLCST -aggrid 20
```

An example of the output from issuing this command would be:

```
      readWrite  ID 0,,19  valid
      readOnly   ID 0,,20  invalid
      backup     ID 0,,21  invalid
number of sites: 1
  server      flags      aggr  siteAge principal      owner

gdlsst.endicott.ib RW      20      0:00:00 hosts/GDLCST <nil>
```

FLDB entry created for fileset hfsset1 (0,,19) on aggregate 20 of GDLCST

Note: RFS aggregate and returns a fileset unique ID number to be used in the **dfstab** file. The ID number for the **readWrite** entry (0,,19) shown above is the number that is to be used in the **dfstab** file entry for the HFS fileset (see step 5, **Update the dfstab File for the RFS Aggregate**).

5. Update the dfstab File for the RFS Aggregate

The **/opt/dfslocal/var/dfs/dfstab** file must be updated to export the fileset assigned to the RFS aggregate. Use OMVS **OEDIT** to update the **dfstab** file.

The following is an example of the contents of the **dfstab** showing the RFS fileset entry with device name **/dev/ufs020**, aggregate name **ufs20**, and the fileset ID number **0,,19**:

```
/dev/lfs1      lfs1    lfs  1
/dev/lfs2      lfs2    lfs  2
/dev/ufs010    ufs10   ufs  10    0,,22
/dev/ufs020    ufs20   ufs  20    0,,19
```

6. Export the RFS Aggregate

In order to allow DFS client to access the HFS aggregate, the aggregate needs to be exported. In order to export aggregates that have entries in the **devtab** and **dfstab**, issue the following command from the operator's console:

```
modify dfs,start export
```

Or, issue the following command from OMVS while logged on as a **root** user with **UID = 0**.

```
dfsexport ufs10
```

7. Create a DFS Mount Point for the RFS Fileset

In order to create a DFS mount point for the fileset assigned to the RFS aggregate, an **fts crmount** command must be issued.

The following is an example of the commands that can be used to mount the HFS fileset named **OMVS.PRIV.HFS.DATASET** at the mount point **/./rw/mvshfsmtpt**:

```
cd /./rw
fts crmount -fileset OMVS.PRIV.HFS.DATASET -dir mvshfsmtpt
```

Chapter 10. Deconfiguring DFS

Important Note to Users

Before deconfiguring any OS/390 DFS Client or Server processes on an OS/390 DCE host system, be sure to stop the DFS processes on that system (see Chapter 11, “Starting and Stopping DFS Components” on page 131).

Certain situations require that you deconfigure or remove configuration files for all of DFS or for a particular DFS component from an OS/390 DCE host system where DFS is configured.

Selected DFS components may require deconfiguration. If you want to reconfigure a particular component with new parameters, you must deconfigure it, removing the existing configuration, before setting up the new configuration. Or, for example, if a configuration of a component failed and it is only partially configured, you must remove the partial configuration before attempting configuration again.

Situations may require that you deconfigure all DFS components from the server. For example, if you want to transfer a server from one cell to another, you must remove the configurations for the old cell from the server before setting up the configurations for the new cell.

You may want to deconfigure an entire DCE cell. If you choose to do this, you should also unregister the DCE cell's name from the global namespace. Before deconfiguring DCE, you should always deconfigure the DFS Client and Servers.

Exercise caution in deconfiguring DFS components, especially if you are removing components which perform services required by other components. Deconfiguring a component partially or completely disables other components which are dependent upon it. For example, deconfiguring all Fileset Location Database servers, the basis of any cell's DFS system, disables the DFS functions for the cell.

To deconfigure you must set up a TSO user ID (see “Set Up the DFSCONF Administrator TSO User ID” on page 55) and make sure you are logged in to DCE (see “DCE Login Panel” on page 55) as a user with DCE cell administrator (**cell_admin**) authority.

DFS should be deconfigured prior to deconfiguring DCE on the host system. After DFS is deconfigured, you can deconfigure DCE. After DCE is reconfigured, you can then configure DFS on the host system following the steps described in Chapter 7, “Configuring DFS” on page 47. Special actions are required if DFS is configured on a host system and DCE is deconfigured and configured on the same host system with a different DCE host name (see “DFS Deconfiguration Steps for DCE Host Name Change” on page 128).

Note: Before any DFS deconfiguration is done, you should review the “DCE Deconfiguration Considerations for DFS” on page 118 and “Update (upserver) Deconfiguration Considerations” on page 118.

The actions associated with deconfiguring a DFS component are described in this chapter. The chapter includes:

- “DCE Deconfiguration Considerations for DFS” on page 118
- “Update (upserver) Deconfiguration Considerations” on page 118
- “DFS Component Deconfiguration” on page 119
- “Deconfigure Backup Database Server (bakserver)” on page 120
- “Deconfigure the Fileset Location Database Server (flserver)” on page 121
- “Deconfigure the Fileset Server (ftserver)” on page 123
- “Deconfigure the DFS File Server” on page 123

- “Using DFSCONF to Deconfigure DFS Servers” on page 124
- “Using DFSCONF to Deconfigure the DFS Client” on page 126
- “Removing DFS Servers from a Host System” on page 127
- “Remove the DFS Client from a Host System” on page 127
- “DFS Deconfiguration Steps for DCE Host Name Change” on page 128.

Note: It is recommended that you follow the deconfiguration steps outlined in “DFS Component Deconfiguration” on page 119.

DCE Deconfiguration Considerations for DFS

The following considerations apply to deconfiguring DFS on an OS/390 system:

1. If you need to reconfigure (deconfigure and configure) DCE on an OS/390 system, and DFS is also configured on the OS/390 system, you need to reconfigure both DFS and DCE. It is recommended that the reconfiguration steps be done in the following order:
 - Deconfigure DFS
 - Deconfigure DCE
 - Configure DCE
 - Configure DFS.
2. If DCE is reconfigured while DFS remains configured on an OS/390 DCE host system, it is recommended that DFS be reconfigured on that system.
3. It is recommended that you do not change the DCE host name when reconfiguring the DCE client if DFS is configured on an OS/390 system. Special DFS reconfiguration steps are required if the DCE host name is changed (see “DFS Deconfiguration Steps for DCE Host Name Change” on page 128).
4. After DFS is configured on a DCE host system, you can not reconfigure all the DCE Security Servers or all DCE Cell Directory Servers in a DCE cell without destroying the DFS authorization. The content of all the DFS administrative list file and ACL data for DCE Local File System files is invalidated. File data may not be accessible through DFS even if DFS is reconfigured. You should copy DCE Local File System data to another file system before you deconfigure the DCE Security Servers in a DCE cell.

If all the DCE Security Servers and DCE Cell Directory Servers reconfigure, you can attempt to re-establish DFS operations by:

- a. Deconfiguring the DFS servers using DFSCONF if this was not done prior to deconfiguring DCE.
- b. Configuring the DFS servers using DFSCONF.
- c. Deleting the DFS administrative list files and re-creating the them as described in Chapter 8, “Defining DFS Administrators and Users” on page 83. All the required DFS administrator principals and groups must also be added to the the DFS administrative list files.
- d. Redefining all DFS file ACL data as described in the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Update (upserver) Deconfiguration Considerations

The **upserver** deconfiguration considerations in this section only apply if you have an **upserver** running on the OS/390 host system. Before deconfiguring all DFS servers or just the DFS **upserver** from an OS/390 DCE host system, you may need to configure the **upserver** on another DCE host system in the DCE cell if DFS runs on more than system in the DCE cell.

You can ignore these **upserver** related instructions if:

- DFS is configured on a single DCE host system in the DCE cell.
- There is no **upserver** running in the DCE cell.
- The **upserver** is running on another DCE host system in the DCE cell.
- If the **upserver** remains configured on this OS/390 system after other components of DFS are deconfigured.

If the DFS **upserver** is running on the OS/390 system where DFS including the **upserver** is being deconfigured, the following steps must be taken before any further DFS deconfiguration steps are performed:

1. Stop the **upserver** on the OS/390 host system.
2. Stop the Update Clients on the other DFS host systems in the DCE cell.
3. Configure and start an **upserver** on another DCE host system in the DCE cell.
4. Start an **upclient** on the other DCE host systems in the DCE cell (on a system other than where the **upserver** is now running), identifying the new **upserver** when an **upclient** is started.

DFS Component Deconfiguration

This section identifies the DFS components that can be deconfigured on a OS/390 DCE host system and the sequence in which DFS component deconfiguration can be performed.

Deconfigure the DFS Client (DFSCM)

The DFS Client (**DFSCM**) can be deconfigured independent from DFS servers.

To deconfigure the **DFSCM** on the OS/390 system with DCE host name *dcehostname*, the following steps should be performed:

1. Use **DFSCONF** to deconfigure the **DFSCM** (see “Using DFSCONF to Deconfigure DFS Servers” on page 124).
2. If you plan to remove the **DFSCM** from the DCE host system, you may want to perform the additional steps described in “Remove the DFS Client from a Host System” on page 127.

Deconfigure DFS Servers

All DFS servers are dependent on the configuration of the DFS File Server on the DCE host system. The DFS File Server should be deconfigured last.

The Backup Database Server (**bakserver**) including the Backup Tape Coordinator (**BUTC**) Servers, the Fileset Location Database Server (**flserver**), and the Fileset Server (**ftserver**) can be deconfigured independently and only depend on the configuration of the DFS File Server on the DCE host system.

It is recommended that DFS servers be deconfigured as described in the sections referenced in the following list:

1. Deconfigure the Backup Database Server (**bakserver**) including the Backup Tape Coordinator (**BUTC**) Servers (see “Deconfigure Backup Database Server (bakserver)” on page 120).
2. Deconfigure the Fileset Location Database Server (**flserver**) (see “Deconfigure the Fileset Location Database Server (flserver)” on page 121).
3. Deconfigure the Fileset Server (**ftserver**) (see “Deconfigure the Fileset Server (ftserver)” on page 123).

4. Deconfigure the DFS File Server (see “Deconfigure the DFS File Server” on page 123).
5. If you plan to remove DFS servers from the DCE host system, you may want to perform the additional steps described in “Removing DFS Servers from a Host System” on page 127.

After DFS is deconfigured using **DFSCONF**, any DFS administrative list files and the **BosConfig** file still exist in the directory **/opt/dcelocal/var/dfs**. If you reconfigure DFS, these files are re-used by DFS unless you rename, move, or delete them.

Deconfigure Backup Database Server (bakserver)

To deconfigure the Backup Database Server (**bakserver**) and any Backup Tape Coordinator (**BUTC**) servers, the following steps should be performed:

1. Insure that the **boserver** is running on the host system.
2. Insure that the **bakserver** is running on the host system or in the DCE cell.
3. If any of the **butc01** through **butc08** servers are running the host system, issue the following OS/390 operator command:

```
modify dfs,stop butcnn
```

where: *nn* equals **01** through **08**.

4. Login as a user ID with DCE **cell_admin** authority from OMVS.
5. Issue the following command for each **BUTCnn** server configured on the DCE host system:

```
bak rmhost -tcid n
```

where *n* = the value 0-7 that identifies the **BUTC** process. Note that the **BUTC** process identifier values **0** through **7** relate to the **BUTC01** through **BUTC08** servers.

See “Completing the DFS Backup Tape Coordinator Configuration” on page 69 for more information on identifying **BUTC** servers.

For additional detailed information about the **bak rmhost** command, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

6. Optionally update the DFS Backup Tape Management file. For more information, refer to “Completing the DFS Backup Tape Coordinator Configuration” on page 69, step 6 on page 71.
7. Optionally update the **BUTCnn envar** files to reflect the IBM supplied values. For more information, refer to “Completing the DFS Backup Tape Coordinator Configuration” on page 69, step 5 on page 70.
8. If the **bakserver** is running on the host system, issue the following command from OMVS:

```
bos stop ./:/hosts/dcehostname bakserver
```

9. Issue the following command:

```
bos delete ./:/hosts/dcehostname bakserver
```

For additional detailed information about the **bos delete** command, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

10. Use **DFSCONF** to deconfigure the Backup Database Server (**bakserver**) on the OS/390 DCE host system (see “Using DFSCONF to Deconfigure DFS Servers” on page 124).
11. Complete the **bakserver** deconfiguration (see “Completing the Backup Database Server Deconfiguration” on page 121).

Completing the Backup Database Server Deconfiguration

The section describes additional deconfiguration steps that must be performed after using **DFSCONF** to deconfigure the Backup Database Server (**bkserv** or **bakserver**) (see “Using DFSCONF to Deconfigure DFS Servers” on page 124).

The additional steps that are required to complete the deconfiguration of the (**bkserv**) are:

1. Use the **dcecp -c rpcentry show ./:/subsys/dce/dfs/bak** command to update the entry for the RPC server group **./:/subsys/dce/dfs/bak**. The command forces CDS to update information that it caches from the entry for the group in the namespace.
2. Stop and restart the Backup Database Server process (**bakserver**) on each database server of that type. Restarting the existing database server processes causes the processes to read the updated RPC server group. This ensures that each Ubik coordinator agrees on the number and identities of the other database servers of its type, which is vital to Ubik's use of a quorum of database servers to maintain database consistency.
3. Remove the DFS server principal for the **admin.bak** list.

If there is only a single DCE host system in the DCE cell where DFS has been configured or if there is no DFS **upserver** running on another DCE host system in the DCE cell, you can skip this step. Refer to “Update (upserver) Deconfiguration Considerations” on page 118 for more information.

If there is an **upserver** running on another DCE host system in the DCE cell, use the **bos rmdadmin** command to cause the DCE cell System Control machine where the **upserver** is running to remove the abbreviated DFS server principal **hosts/dcehostname/dfs-server** for this DCE host system from its **admin.bak** administrative list. The **upserver** then distributes the updated list to the Update Clients in the DCE cell.

If you previously chose to add the full or abbreviated DFS server principal to a security group and included the group in the **admin.bak** list, you can use the **dcecp group remove** command to remove the DFS server principal from the group. Note that **DFSCONF** has already removed the abbreviated DFS server principal **hosts/dcehostname/dfs-server** from the security group **subsys/dce/dfs-bak-servers**.

4. Remove the **admin.bak** administrative list from the **/opt/dcelocal/var/dfs** directory on the database server to be removed. Modify the Update Server as necessary if the list is propagated from the cell's System Control machine.

For additional detailed information, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

After performing the actions described in this section, you should proceed to the next step described in “DFS Component Deconfiguration” on page 119.

Deconfigure the Fileset Location Database Server (flserver)

To deconfigure the Fileset Location Database Server (**flserver**) on the OS/390 system with DCE host name **dcehostname**, the following steps should be performed:

1. Insure that the **boserver** is running on the host system.
2. Login as a user ID with DCE **cell_admin** authority from OMVS.
3. If the **flserver** is running on the host system, issue the following command from OMVS:

```
bos stop ./:/hosts/dcehostname flserver
```

4. Issue the following command:

bos delete `./:/hosts/dcehostname flserver`

For additional detailed information about the **bos delete** command, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Note: After issuing the **bos delete** command for the **flserver**, if another **flserver** does not exist in the DCE cell, the **fts** commands shown for deconfiguring the DFS File Server (see “Deconfigure the DFS File Server” on page 123) can not be issued.

5. Use **DFSCONF** to deconfigure the Fileset Location Database Server (**flserver**) on the OS/390 DCE host system (see “Using DFSCONF to Deconfigure DFS Servers” on page 124).
6. Complete the **flserver** deconfiguration (see “Completing the Fileset Location Database Server Deconfiguration”).

Completing the Fileset Location Database Server Deconfiguration

The section describes additional deconfiguration steps that must be performed after using **DFSCONF** to deconfigure the Fileset Location Database Server (**flserver**), (See “Using DFSCONF to Deconfigure DFS Servers” on page 124 for more information.)

The additional steps that are required to complete the deconfiguration of the **flserver** are:

1. Use the **dcecp rpcentry show** command on the Fileset Database Server to update the entry for the appropriate RPC server group from CDS. The command forces CDS to update information that it caches from the entry for the group in the namespace.
2. Stop and restart the Fileset Database Server process (**flserver**) on each database server of that type. Restarting the existing database server processes causes the processes to read the updated RPC server group. This ensures that each Ubik coordinator agrees on the number and identities of the other database servers of its type, which is vital to Ubik's use of a quorum of database servers to maintain database consistency.
3. Remove DFS server principal for the **admin.fl** list

If there is only a single DCE host system in the DCE cell where DFS has been configured or if there is no DFS **upserver** running on another DCE host system in the DCE cell, you can skip this step. Refer to “Update (upserver) Deconfiguration Considerations” on page 118 for more information.

If there is an **upserver** running on another DCE host system in the DCE cell, use the **bos rmdadmin** command to cause the DCE cell System Control machine where the **upserver** is running to remove the abbreviated DFS server principal **hosts/dcehostname/dfs-server** for this DCE host system from its **admin.fl** administrative list. The **upserver** then distributes the updated list to the Update Clients in the DCE cell.

If you previously chose to add the full or abbreviated DFS server principal to a security group and included the group in the **admin.fl** list, you can use the **dcecp group remove** command to remove the DFS server principal from the group. Note that **DFSCONF** has already removed the abbreviated DFS server principal **hosts/dcehostname/dfs-server** from the security group **subsys/dce/dfs-fs-servers**.

4. Remove the **admin.fl** administrative list from the **/opt/dcelocal/var/dfs** directory on this DCE host system. Modify the Update Server as necessary if the list is propagated from the cell's System Control machine.

For additional detailed information, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

After performing the actions described in this section, you should proceed to the next step described in “DFS Component Deconfiguration” on page 119.

Deconfigure the Fileset Server (ftserver)

To deconfigure the DFS Fileset Server (**ftserver**), the steps shown in this section should be performed.

Note: You can skip the steps that require an **fts** command to be issued if the Fileset Location Database Server (**flserver**) has already been deconfigured on this host system and there is no other **flserver** running in the DCE cell. Refer to “Deconfigure the Fileset Location Database Server (flserver)” on page 121 for more information.

The steps to deconfigure the **ftserver** on an OS/390 system with DCE host name *dcehostname* are:

1. Insure that the **boserver** is running on the host system.
2. Insure that the **ftserver** is running the host system.
3. Login as a user ID with DCE **cell_admin** authority from OMVS.
4. List fileset entries from the Fileset Location Data Base (FLDB) for the host using the following command:

```
fts lsflldb -server /./hosts/dcehostname
```

5. Delete each fileset entry from the FLDB for *dcehostname* using the following command:

```
fts delfldbentry -fileset abc.fileset -server {ipaddress | /./hosts/dcehostname}
```

6. Delete the server entry from the FLDB for *dcehostname* using the following command:

```
fts delserverentry -server {ipaddress | /./hosts/dcehostname}
```

7. Delete the DFS customizable files described in Table 19 on page 150.

8. If the **ftserver** is running on the host system, issue the following command from OMVS:

```
bos stop /./hosts/dcehostname ftserver
```

9. Issue the following command:

```
bos delete /./hosts/dcehostname ftserver
```

For additional detailed information about the **bos** and **fts** commands, see *OS/390 Distributed File Service DFS Administration Guide and Reference*.

After performing the actions described in this section, you should proceed to the next step described in “DFS Component Deconfiguration” on page 119.

Deconfigure the DFS File Server

To deconfigure the DFS File Server the steps shown in this section should be performed.

Note: You can skip the steps that require an **fts** command to be issued if the Fileset Location Database Server (**flserver**) has already been deconfigured on this host system and there is no other **flserver** running in the DCE cell. Refer to “Deconfigure the Fileset Location Database Server (flserver)” on page 121 for more information.

The steps to deconfigure the DFS File Server on an OS/390 system with DCE host name *dcehostname* are:

1. Login as a user ID with DCE **cell_admin** authority from OMVS.
2. Verify that the DFS server entry for DCE host name *dcehostname* still exists by issuing the command:

```
$ fts lsserverentry -all
```

3. If it exists, delete the server entry from the FLDB for *dcehostname* using the following command:

```
$ fts delserverentry -server {ipaddress | /./hosts/dcehostname}
```

4. Use **DFSCONF** to deconfigure the DFS File Server (see “Using DFSCONF to Deconfigure DFS Servers”).

After performing the actions described in this section, you should proceed to the next step described in “DFS Component Deconfiguration” on page 119.

Using DFSCONF to Deconfigure DFS Servers

Note: Refer to “Using the DFS Configuration Program (DFSCONF)” on page 53 for information on how to use **DFSCONF** to DCE login as a cell administrator and select the option to **Deconfigure DFS Servers** from the DFSCONF Main Menu.

Using the **DFSCONF** panels to deconfigure your system automates some of the deconfiguration process. It is recommended that you use the **DFSCONF** panels to deconfigure DFS on your OS/390 DCE host system.

If you plan to completely deconfigure your system, it is recommended that the Backup Tape Coordinator, Backup Database server, and Fileset Database server be deconfigured before the DFS File Server is deconfigured. The DFS Client can be deconfigured before or after any of the DFS servers are deconfigured.

Selecting Deconfigure DFS Servers from the DFSCONF Main Menu displays the panel shown in Figure 9.

```
IOEBDCFG----- DECONFIGURE DFS SERVERS -----
COMMAND ===>

DCE Cell Name      ===> dcecellname
DCE Host Name      ===> dcehostname

Options:
  DFS File Server      ===> N
  DFS Backup Server    ===> N
  DFS Fileset Database Server ===> N

Enter END COMMAND to return to previous menu.

F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP  F10=LEFT   F11=RIGHT  F12=RETRIEVE
```

Figure 9. Deconfiguring DFS Servers Panel

The menu items are:

Menu Item	Description
DCE Cell Name	This is the DCE cell name used to previously configure DCE on this OS/390 system and is obtained from the DCE configuration file /opt/dcelocal/dce_cf.db . It normally should not be changed on this panel and is displayed for information purposes only.
DCE Host Name	This is the DCE host name currently defined by the DCE configuration on this OS/390 system and is obtained from the DCE configuration file /opt/dcelocal/dce_cf.db . It normally should not be changed on this panel but if you want to deconfigure the DFS servers using a different DCE host name you can change it. The DCE host name value is case-sensitive.
DFS File Server	To deconfigure the DFS File Server, change the option from N to Y . To initiate the deconfiguration of this server, press <Enter> after selecting other options.
Backup Server	To deconfigure the Backup Database Server (bakserver), and Backup Tape Coordinator (butc) servers, change the option from N to Y . To initiate the deconfiguration of this server, press <Enter> after selecting other options.
DFS Fileset Location Database Server	To deconfigure the Fileset Location Database server (flserver), change the option from N to Y . To initiate the deconfiguration of this server, press <Enter> after selecting other options.

After deconfiguration processing is initiated (by pressing **<Enter>**) for the selected servers, the servers are deconfigured in order, starting with the bottom option working up the list.

The program begins issuing the DCE administration commands that deconfigure the server (see Appendix A, “Example DFS Server Configuration Log File (dfsconf.log)” on page 141).

If not interrupted, and if successfully completed, the program displays a server Deconfiguration Ended....SUCCESSFULLY message for each selected server.

If interrupted, or if deconfiguration was not successful for a server, a server Deconfiguration Ended....UNSUCCESSFULLY message is displayed for the server.

To determine where the error occurred that caused the deconfiguration to fail, examine the **dfsconf.log** file, which by default is in your home directory, or at the location specified by the environment variable **_IOE_CFG_LOG_FILE**. Refer to Appendix A, “Example DFS Server Configuration Log File (dfsconf.log)” on page 141 for an example output of a **dfsconf.log** file. If the **dfsconf.log** file already exists, **DFSCONF** appends messages to the end of the log file.

When **DFSCONF** processing is complete, if you want to now (re-)configure the DFS server or servers on the DCE host system, refer to “DFS Configuration Main Menu” on page 56.

If you intend to remove the DFS Fileset Location Database Server (**flserver**) from this DCE host system, refer to “Completing the Fileset Location Database Server Deconfiguration” on page 122.

If you intend to remove the DFS Backup Database Server (**bakserver** or **bakserver**) from this DCE host system, refer to “Completing the Backup Database Server Deconfiguration” on page 121.

If you want to remove all DFS servers from the DCE host system, after completing the instructions in “Completing the Fileset Location Database Server Deconfiguration” on page 122 and “Completing the Backup Database Server Deconfiguration” on page 121, you should refer to “Removing DFS Servers from a Host System” on page 127.

After following the previous instructions in this section, DFS server deconfiguration is complete.

Using DFSCONF to Deconfigure the DFS Client

Note: Refer to “Using the DFS Configuration Program (DFSCONF)” on page 53 for information on how to use **DFSCONF** to DCE login as a cell administrator and select the option to **Deconfigure DFS Client** from the DFSCONF Main Menu.

The DFS client (**DFSCM**) should be deconfigured to remove it from a DCE host system.

Note: If you intend to simply change the form of **DFSCM** caching (e.g from memory caching to disk caching), there is no need to deconfigure the **DFSCM** before changing the disk configuration values. See “DFSCM Disk Caching” on page 75 for more information.

As the **DFSCM** does not require any DFS servers to be running on the same OS/390 system, the deconfiguration process for **DFSCM** is handled separately from the DFS servers.

Selecting DFS Client from the DFSCONF Main Menu displays the panel shown in Figure 10.

```
IOEBDCL----- DECONFIGURE DFS CLIENT -----
COMMAND ===>
DCE Cell Name ===> dcecellname
DCE Host Name ===> dcehostname

Options:
  DFS Client ===> N
Enter END COMMAND to return to previous menu.

F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP    F10=LEFT   F11=RIGHT   F12=RETRIEVE
```

Figure 10. Deconfiguring the DFS Client (DFSCM) Panel

The menu items are:

Menu Item	Description
DCE Cell Name	This is the DCE cell name used to previously configure DCE on this OS/390 system and is obtained from the DCE configuration file /opt/dcelocal/dce_cf.db . It normally should not be changed on this panel and is displayed for information purposes only.

DCE Host Name	This is the DCE host name currently defined by the DCE configuration on this OS/390 system and is obtained from the DCE configuration file /opt/dcelocal/dce_cf.db . It normally should not be changed on this panel but if you want to deconfigure the DFS client using a different DCE host name you can change it. The DCE host name value is case-sensitive.
DFS Client	To initiate the deconfiguration of the DFSCM , change the DFS Client option from N to Y and press <Enter> .

Note: After the **DFSCM** is deconfigured as described above, you should not restart the DFS client as described in “Starting and Stopping the DFS Client (DFSCM)” on page 131 before the **DFSCM** is (re)configured as described in Chapter 7, “Configuring DFS” on page 47.

When DFSCONF processing is complete, if you want to now (re-)configure the **DFSCM** on the DCE host system, refer to “DFS Configuration Main Menu” on page 56.

Or, if you intend to remove the **DFSCM**, from your system, refer to “Remove the DFS Client from a Host System.”

Removing DFS Servers from a Host System

To remove DFS servers from a OS/390 DCE host system, you should:

1. Deconfigure all the DFS servers as outlined in “DFS Component Deconfiguration” on page 119
2. Insure all the DFS customizable files described in Table 19 on page 150 are deleted
3. Delete any DCE LFS filesets allocated on the OS/390 host system for exporting to DFS clients.

Remove the DFS Client from a Host System

After deconfiguring DFS Client (**DFSCM**) as described in “Deconfigure the DFS Client (DFSCM)” on page 119, you can perform the following actions to remove the **DFSCM** from a host system:

If you intend to remove the **DFSCM** from your system, perform the following actions:

1. Remove the **BPXPRMxx FILESYSTYPE TYPE(DFSC)** entry in the data set that is allocated to the IEFPARM DD statement in the OS/390 system start-up procedure. Once removed, OS/390 does not attempt to start the DFS client during the next OS/390 initialization (IPL).
2. Delete the **/opt/dcelocal/etc/CacheInfo** file.
3. Delete the **/opt/dfslocal/home/dfscm/envvar** file.
4. If a DCE Local File System aggregate was allocated during the initial configuration for DFS client disk caching, de-allocate the aggregate or re-use the aggregate for other purposes as needed.

Note: After the **DFSCM** is fully deconfigured and removed from your system, you should not restart the DFS client as described in “Starting and Stopping the DFS Client (DFSCM)” on page 131 before OS/390 is re-IPL'ed.

DFS Deconfiguration Steps for DCE Host Name Change

To change a DCE host name for an OS/390 system, DCE must be reconfigured. If DFS is configured on a OS/390 DCE host system, it is recommended that DFS be deconfigured before DCE is deconfigured. Then DCE and DFS can then be configured using the new DCE host name.

Additionally, certain DFS administrative files must be changed and re-created to reflect the new DCE hostname. This section describes how to change the DCE hostname and the DFS configuration to reflect the new DCE host name. It includes the following:

- “Recommended Reconfiguration Steps When Changing DCE Host Name”
- “DCE Host Name Change While DFS is Configured”
- “DCE Host Name Change While DFS is Configured.”

Unless otherwise noted, more information on the commands and files referred to in this section can be found in the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Also, it is recommended that the command parameter specification *./hosts/dcehostname* should be used instead of the *ipaddress* parameter whenever possible.

If you need more information on deconfiguring DCE and reconfiguring DCE see the *OS/390 DCE Configuring and Getting Started* book. If you need more information on the DCE commands shown in this section (for example, **dcecp** and **cdscp**), see the *&euvmk001*..

For additional information on how to deconfigure DFS using **DFSCONF**, see Chapter 10, “Deconfiguring DFS” on page 117.

Recommended Reconfiguration Steps When Changing DCE Host Name

The recommended sequence of DFS and DCE reconfiguration steps are:

1. Use **DFSCONF** to deconfigure DFS using the current DCE host name.

Note: The DCE host name displayed on the DFSCONF panels is the DCE host name that must be used for the DFS deconfiguration in this scenario.

2. Deconfigure DCE.
3. Configure DCE using a new DCE host name.
4. Configure DFS using the new DCE host name.

Note: The new DCE host name that must be used for DFS configuration is displayed appropriately on DFSCONF panels.

5. Complete DFS deconfiguration by removing the DFS entries from various configuration files that contain the previous DCE host name and creating new configuration files entries that contain the new DCE host name (see “DCE Host Name Change While DFS is Configured”).

DCE Host Name Change While DFS is Configured

Although not recommended, if DCE is deconfigured and reconfigured with a new DCE host name (*newdcehostname*) while DFS remains configured on an OS/390 system with the previous DCE host name (*olddcehostname*), DFS does not operate properly.

Note: Make sure you review, “DCE Deconfiguration Considerations for DFS” on page 118, before you proceed with the following.

To restore DFS to an operational status, DFS must be reconfigured on the OS/390 DCE host system using the following sequence of DFS reconfiguration steps:

1. Deconfigure DFS

Use **DFSCONF** to deconfigure all DFS components that are configured on the OS/390 system.

Note: The DCE host name displayed on the DFSCONF panels is the *newdcehostname*. It must be replaced with the *olddcehostname* used for the DCE configuration. You can ignore any errors that occurred while running **DFSCONF**.

2. Configure DFS

Use **DFSCONF** to configure all DFS components that are required on the OS/390 system.

Note: It is the *newdcehostname* that must be used for this DFS configuration. Change the DCE host name on the panel if the new DCE host name is not displayed.

3. Start DFS Servers

At this time, you must start the DFS servers that were previously configured on this DCE host system using the OS/390 system operator command **start dfs**. Since the DFS administrative list files in the directory */opt/dfslocal/var/dfs* and the */opt/dcelocal/var/dfs/BosConfig* file should be intact from the previous configuration of DFS, the DFS servers initialize successfully.

4. Complete DFS deconfiguration

You must complete the DCE host name change for DFS by removing the DFS entries from various configuration files that contain the previous DCE host name, and by creating new configuration file entries that contain the new DCE host name.

The steps to complete DFS deconfiguration and reconfiguration after DCE is configured using the new DCE host name are listed below. Some of the steps to establish the DFS configuration file entries for the *newdcehostname* are the same as the additional steps used to complete the configuration of the DFS components that are described in Chapter 7, “Configuring DFS” on page 47.

The steps to complete DFS deconfiguration and reconfiguration after a DCE host name change are:

- a. Verify the correct DCE host name by issuing the following command:

```
dcecp -c host catalog | grep newdcehostname
```

Note: The **dcecp** command is described in the *OS/390 DCE Configuring and Getting Started* book.

- b. You must determine the TCP/IP address (*ipaddress*) that corresponds to the *newdcehostname* using the following command:

```
cdscp show object /./hosts/newdcehostname/self | grep ncadg
```

Note: The **cdscp** command is described in the *OS/390 DCE Configuring and Getting Started* book. Also, it is recommended that you remember the *ipaddress* in case you may required it at a later time.

- c. Using the TCP/IP address list fileset entries from the Fileset Location Data Base (FLDB) for the DCE host system using the following command:

```
fts lsflldb -server ipaddress
```

Note: You must remember the fileset names and fileset IDs for the filesets displayed if you have HFS or RFS filesets and you did not name your HFS or RFS filesets using the MVS data set name (*mvsdatasetname*) allocated for the fileset. In step 4i on page 130, you need to refer to this list to determine the correct fileset name to use for HFS or RFS filesets.

- d. Delete each fileset entry from the FLDB for the DCE host system using the following command:

```
fts delflldbentry -fileset abc.fileset -server ipaddress
```

- e. Remove the entry from the Fileset Location Database (FLDB) for the DFS servers that were configured on the OS/390 host system using the *olddcehostname*. Delete the server entry using the following command:

```
fts delserverentry -server ipaddress
```

- f. Add the server entry for *newdcehostname* using the following command:

```
fts crserverentry -server ./:/hosts/newdcehostname -principal hosts/newdcehostname
```

- g. Synchronize the FLDB with the fileset IDs using the following command:

```
fts syncfdb -server ./:/hosts/newdcehostname
```

Note: The **fts syncfdb** command sets HFS and RFS fileset names to the MVS data set name (*mvdsdatasetname*) allocated for the fileset. HFS and RFS fileset names that are different from the MVS data set name are not propagated by the **fts syncfdb** command.

- h. Display FLDB to see the current fileset names and fileset IDs using the following command:

```
fts lsfdb
```

Note: The HFS and RFS fileset names displayed are the MVS data set names from the **devtab** file (*/opt/dcelocal/var/dfs/devtab*).

- i. If you do not name your HFS or RFS filesets using the MVS data set name (*mvdsdatasetname*) allocated for the fileset, you must now rename the HFS or RFS filesets.

First, you must determine the fileset name that was previously used for each fileset before DFS was reconfigured. In step 4c, you listed and saved the previous fileset entries from the FLDB for the DCE host. This list is referred to as the *previous FLDB list*. In step 4h, you displayed the current fileset entries from the FLDB for all DCE hosts in the cell where DFS is exporting filesets. This display is referred to as the *current FLDB list*. Now, to rename the HFS or RFS filesets follow these steps:

- 1) In the current FLDB list, look in the **aggr** column to determine the aggregate name.
- 2) Look in the **dfstab** file to determine if the aggregate name has a file system type of **ufs**.
- 3) If the aggregate has a file system type of **ufs**, locate the aggregate name in the previous FLDB list and use the associated fileset name as the *previousfilesetname*. The *previousfilesetname* is used as the **-newname** specification below.
- 4) In the current FLDB list entry for this aggregate name, you can determine the current fileset name which was set to the *mvdsdatasetname* by the **fts syncfdb** command issued in step 4g. The *mvdsdatasetname* is used as the **-oldname** specification below.
- 5) Now, you can rename the HFS or RFS fileset in the FLDB using the following command:

```
fts rename -oldname mvdsdatasetname -newname previousfilesetname
```

Chapter 11. Starting and Stopping DFS Components

This chapter describes the procedures necessary to start and stop the DFS client (**DFSCM**) and the DFS servers on OS/390 DFS. On OS/390 DFS, the DFS client is implemented as a physical file system running as a colony address space, **DFSCM**. The DFS servers run in the DFS address space, **DFS**. A separate **DFSKERN** address space may optionally exist to provide better recovery capability for the server processes running in the **DFS** address space.

This chapter discusses the following topics:

- “Starting and Stopping the DFS Client (DFSCM)”
- “Starting and Stopping the DFS Servers” on page 133.

Starting and Stopping the DFS Client (DFSCM)

This section discusses the DFS client (**DFSCM**) address space and various ways of starting and stopping the DFS client on OS/390 DFS. It contains the following topics:

- “Who Can Start and Stop DFSCM?”
- “DFSCM Initialization”
- “Starting the DFS Client (DFSCM)” on page 132
- “Stopping and Restarting the DFSCM” on page 133.
- “DFSCM Relationship to DCE and DFS Servers” on page 133.

Who Can Start and Stop DFSCM?

A user with OS/390 operator privileges is the only one who can start or stop the DFS client (**DFSCM**).

DFSCM Initialization

The **DFSCM** is started as part of OS/390 initialization and can be stopped and restarted without stopping OS/390. Other OS/390 DFS servers do not need to be running on the local OS/390 system with the **DFSCM**. **DFSCM** is a colony address space that is automatically started by OMVS as a physical file system during system IPL when OMVS is started. The DFS client is identified as a physical file system in the system parmlib **BPXPRMxx** member **FILESYSTYPE TYPE(DFSC)** entry that is read during system initialization.

Note: The variable, *xx*, is a suffix used to uniquely identify system parmlib members.

On OS/390, the **FILESYSTYPE TYPE(DFSC)** entry in the system parmlib member **BPXPRMxx** causes OS/390 to initialize the **DFSCM**. Information is passed from the **FILESYSTYPE TYPE(DFSC)** entry to the **DFSCM**. Refer to “System Parmlib Member BPXPRMxx Entry for the DFSCM” on page 72 for more information.

In OS/390, the DFS client runs in the colony address space, **DFSCM**. **DFSCM** daemons **ioecmini**, **ioedfsd**, and **ioellogin** run as processes within the **DFSCM** address space and can not be controlled separately.

When initializing the **DFSCM**, OS/390 UNIX issues the **ioecmini** process. The **ioecmini** process performs several functions including:

- Starting the **ioedfsd** process which in turn, creates various worker threads.

- Mounting the HFS file system name specified in the system parmlib member **BPXPRMxx FILESYSTYPE TYPE(DFSC)** entry for the DFS client.
- Returning to OS/390 UNIX before it has completed initialization.
- Waiting for DCE to initialize before completing initialization and notifying OS/390 UNIX that the DFS client is available as a physical file system.

After **DFSCM** notifies OS/390 UNIX that the DFS client is available as a physical file system, **ioecmini** is subsequently invoked by OS/390 UNIX in cross memory mode by a program call as a result of a file system interface request when an OMVS user or application references a directory or file in the DFS (*/...*) namespace. More information on the DFS namespace follows.

The **ioecmini** process queues work for the **ioedfsd** process which does not run in cross memory mode. The **ioedfsd** process and its threads perform the necessary work to service the request and returns information to the **ioecmini** process which in turn, returns the information to OS/390 UNIX and the OMVS user or application. Additionally, requests are queued to the **ioellogin** process by the **ioedfsd** process to perform DCE login services.

Important Note to Users

The **ioedfsd** process is also unique in OS/390 in that it combines the functions of the traditional Cache Manager **dfsd** and **dfsbind** processes. The **dfsd** and **dfsbind** command options are described in the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

During initialization, the **DFSCM** also reads parameters from the local HFS **/opt/dcelocal/etc/CacheInfo** file and the **/opt/dfslocal/home/dfscm/envvar** file. A parameter specified in the **CacheInfo** file is overridden by the same parameter specified in the **envvar** file. Refer to “Completing the DFS Client (DFSCM) Configuration” on page 72, for more information on the **CacheInfo** and **envvar** file.

Important Note to Users

DFSCM information specified in the system parmlib member **BPXPRMxx** only takes effect for the **DFSCM** when OMVS is started. Also, the **DFSCM _EUV_HOME** environment variable must be specified in the system parmlib **BPXPRMxx FILESYSTYPE TYPE(DFSC)** entry.

Initialization processing for the **DFSCM** waits until DCE initialization processing is completed. The initialization process for the **DFSCM** assumes that the local directory, */...*, exists and mounts the DFS global namespace, identified locally by an OS/390 Hierarchical File System (HFS) file system name, at this directory. The HFS file system does not need to physically exist but the file system name should be unique as it identifies the DFS global namespace on the local OS/390 host system. Any information previously mounted or accessed using the */...* pathname is overlaid and unavailable while the **DFSCM** is running. Refer to the *OS/390 Program Directory* or to the *OS/390 Distributed File Service DFS Administration Guide and Reference* for more information on the local HFS file system used for DFS.

Starting the DFS Client (DFSCM)

DFSCM is started automatically, during the system IPL, as part of OMVS initialization. After the initialization has completed successfully you receive the following message:

```
IOEC04183I DFSD: initialization complete
```

In OS/390, you can customize and control several DFS client features by specifying **DFSCM** initialization parameters in the **/opt/dcelocal/etc/CacheInfo** file and **_IOE_CM_PARMS** environment variable (located

in `/opt/dfslocal/home/dfscm/envvar`). For more information, see “DFSCM Initialization” on page 131 and “Completing the DFS Client (DFSCM) Configuration” on page 72.

Stopping and Restarting the DFSCM

To **stop** the **DFSCM** address space, use the **stop** operator command. (Refer to *OS/390 MVS System Commands* book, GC28-1781, for more information on the **stop** operator command.)

```
stop dfscm
```

To **restart** the **DFSCM** address space, reply to the OS/390 operator console message:

```
*nn BPXF014D FILESYSTYPE DFSC TERMINATED.  REPLY 'R' WHEN READY TO RESTART.
```

DFSCM Relationship to DCE and DFS Servers

To configure and use the **DFSCM**, DCE must be running on the same host system. **DFSCM** waits for DCE to start before completing initialization. If DCE stops while the **DFSCM** is running, the **DFSCM** processing fails.

Starting and Stopping the DFS Servers

Note: In OS/390 DFS, the DFS servers are sometimes referred to as *daemons*.

This section describes the procedures necessary to start and stop the DFS servers or daemons on OS/390 DFS. For an overview of the DFS servers and processes, see “DFS Address Space, Processes, and PROCLIBs” on page 13.

This section includes the following topics:

- “Who Can Start and Stop DFS Server and boserver Processes?”
- “Order of Starting DFS Daemons” on page 134
- “Starting DFS Server with System START Command” on page 134
- “Controlling DFS Server Processes with System MODIFY Command” on page 135
- “Stopping the DFS Server with System STOP Command” on page 134
- “Starting the boserver Processes from OMVS” on page 136
- “Stopping All the boserver Processes from OMVS” on page 139
- “DFS Server Relationship to DCE” on page 139.

Who Can Start and Stop DFS Server and boserver Processes?

There are two types of users who can start or stop the DFS server OS/390 address space and the processes controlled by **DFSCNTL**, (see “Controlling DFS Server Processes with System MODIFY Command” on page 135):

- A user with OS/390 operator privileges.
- A user who has update privilege to the **DFSKERN.START.REQUEST** RACF facility. This facility is created during the installation of DFS. For more information on this RACF facility, refer to the *OS/390 Program Directory*.

The **boserver** process itself is controlled by the **DFSCNTL** process. But it, in turn, controls sub-processes (see “Starting the boserver Processes from OMVS” on page 136, for more information on these processes). To start or stop the **boserver** sub-processes from OMVS using the **bos create** or **bos stop** command, the issuer must be logged into DCE with DCE principal or DCE group that is listed in the **admin.bos** file on the DCE host where the **boserver** sub-process is running. (Note, you can also issue

bos commands with the **-localauth** option when you are logged on the OS/390 system as **root** user with **UID=0**. For further information, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.)

Order of Starting DFS Daemons

When DFS starts, the DFS daemons that are started automatically are controlled by the configuration type in the **/opt/dfslocal/etc/ioepdcf** file.

When DFS daemons are started manually, the successful start-up of some daemons depends on the availability of services provided by other daemons. This implies that the DFS daemons must be started in a particular order.

The following is the sequence in which DFS daemons should be started on a DCE host system:

1. **dfskern**
2. **export**
3. **boserver**
4. **bakserver**
5. **butcnn**
6. other DFS servers controlled by the **boserver** can be started in any order.

For example to start the **boserver**, the **dfskern** daemon must already be up and running.

If the DFS daemons are started using the **modify dfs,start all** OS/390 system operator command, DFS ensures that the correct starting order is followed.

Note that if the **/opt/dfslocal/etc/ioepdcf** file indicates that a **butc** daemon should be started automatically, and a **bakserver** is not already running in the DCE cell, the **butc** daemon fails to start automatically. If the **bakserver** is run on the same DCE host system with a **butc** daemon, the **butc** daemon must be started using the **modify dfs,start butcnn** command.

Make sure that the passwords of the DFS daemons are valid before starting them up. If the passwords have expired, the daemons cannot be started. For further information, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

Starting DFS Server with System START Command

The OS/390 system command, **start**, is used to start DFS in OS/390 DFS (for detailed information regarding this command, see *OS/390 Distributed File Service DFS Administration Guide and Reference*).

To start **dfs**, issue the following command:

```
start dfs
```

A message written by **dfsexport** specifies each aggregate that you have exported and that is attached to the file system. If there are problems in exporting an aggregate, **dfsexport** issues a message identifying the aggregate that failed to attach.

Stopping the DFS Server with System STOP Command

To stop the DFS address space, use the **stop** operator command to ensure the normal shutdown of the address space.

To stop the DFS address space and all DFS daemons, enter the following OS/390 operator commands:

modify dfs,start unexport (optional)
stop dfs

Note: You can issue the **modify** command with the **unexport** option to perform an orderly shutdown of the DFS server. The **modify dfs,start unexport** command insures updates to files made by DFS clients are written to the DFS server and makes the files in exported aggregates unavailable to the DFS clients. It is not necessary to issue the command if a normal DFS shutdown is being performed using the **stop dfs** command. It is shown here for your information. The **modify dfs,start export** command can be used to instruct the DFS file server to re-export aggregate file data. Refer to “Controlling DFS Server Processes with System MODIFY Command” for more information.

Controlling DFS Server Processes with System MODIFY Command

Once the DFS address space is started, the DFS Control Task, **DFSCNTL**, is running. Processes under its control can be stopped, restarted, or queried by issuing the OS/390 system command, **modify** (for detailed information regarding this command, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*). Processes that run under **DFSCNTL** are: **dfskern**, **dfsexport** (which includes the **export** and **unexport** processes), **boserver**, and **butc01** through **butc08**.

Note: In OS/390 DFS, the following subprocesses run under the control of the **boserver** process: **bkservr**, **flservr**, **ftservr**, **rpserver**, **upserver**, and **upclient**. These processes are started and stopped differently (see “Starting the boserver Processes from OMVS” on page 136 and “Stopping All the boserver Processes from OMVS” on page 139).

For example, to start the **butc01** daemon, enter the following:

```
modify dfs,start butc01
```

To start all the daemons, enter the following:

```
modify dfs,start all
```

Do not use the **modify** command to start additional DFS daemons while the DFS address space is still initializing. During initialization, **DFSCNTL** attempts to start all the processes that run under its direct control that have been configured on the OS/390 host system and have the appropriate configuration type in the **/opt/dfslocal/etc/ioepdcf** file (for further information, see “Customizing the ioepdcf File” on page 78). If you issue the **modify** command while DFS is initializing, the DFS daemons may be started out of order or stopped erroneously. This may lead to unexpected errors during initialization and cause DFS to end abnormally.

Note: You must wait until DFS has issued a log message indicating that DFS initialization has completed before using additional **modify** commands.

To stop the DFS daemons that run under **DFSCNTL** (**dfskern**, **export**, **unexport**, **boserver**, **butc01** thru **butc08**) but not the DFS address space, use the **stop all** command. For example:

```
modify dfs,stop all
```

The **stop all** command causes **DFSCNTL** to stop all daemons that it controls. The **boserver** is stopped, causing all daemons controlled by it to also be stopped.

You can use the **modify dfs** system command to stop any specific daemon that runs under **DFSCNTL**. For example, to stop the **boserver** and all its sub-processes, enter:

```
modify dfs,stop boserver
```

Starting the boserver Processes from OMVS

The **boserver** process, like all processes that run under the control of the Control Task, can be started automatically during system initialization. This can be done by specifying a **Y** for the **boserver's** configuration type in the **/opt/dfslocal/etc/ioepdcf** file (for further information, see “Customizing the ioepdcf File” on page 78). Processes that run under the control of the **boserver** are started using the **bos create** command and are automatically started when the **boserver** starts. These processes are:

- **bkserver**
- **flserver**
- **ftserver**
- **rpserver**
- **upserver**
- **upclient**.

Starting the **boserver** creates empty **/opt/dfslocal/var/dfs/BosConfig** and **/opt/dfslocal/var/dfs/admin.bos** files if the files do not exist.

Prior to starting any of the processes controlled by the **boserver**, consult the following:

- If you plan to start the **bkserver**, refer to “Completing the Backup Database Server (bakserver) Configuration” on page 68.
- If you plan to start the **flserver**, refer to “Completing the FLDB Server (flserver) Configuration” on page 61.
- If you plan to start the **ftserver**, refer to “Completing Fileset Server (ftserver) Configuration” on page 63.
- If you plan to start the **rpserver**, refer to “Completing Replication (repserver) Server Configuration” on page 65.
- If you plan to start the **upclient**, refer to “Completing Update Client (upclient) Configuration” on page 65.
- If you plan to start the **upserver**, refer to “Completing Update Server (upserver) Configuration” on page 66.

Important Note to Users

Do not run the **upserver** and **upclient** daemons on the same OS/390 machine.

To start any of the **boserver's child** processes, use the **bos create** command. To direct **STDOUT** and **STDERR** from the **child** processes to their own output DD name, a redirection parameter is needed for the **bos create** command. The parameters must also include the definition of an environment variable to define the program's home directory. This allows the server running under DCE to find the proper **krb5ccname** file and to perform its own **dce_login**. The path for the program must be substituted with the partitioned data set member name of the program to be run.

Besides starting the **child** process, the **bos create** command updates the **/opt/dfslocal/var/dfs/BosConfig** file and cause the **child** process to be started each time the **boserver** is restarted.

In the following example, of starting the **ftserver**, **/./hosts/DFSMVS** identifies the OS/390 host running the DFS server started task. The **ftserver simple** entry specifies the server process name and process type. The **ftserver** entry immediately following the first " (double quote) is the name of the member in the SIOELMOD data set in the OS/390 MVS partitioned data set (PDS) load library (the SIOELMOD data set was created during OS/390 DFS installation). For further information, refer to the *OS/390 Program Directory*. The portion of the command in double quotes specifies the program, **ftserver**, to be executed, sets a runtime parameter, **envar**, and redirects **STDOUT** for this process to the OS/390 ddname

dd:ftserver. In OS/390 DFS, **envvar**, the runtime parameter, must include **_EUV_HOME** which points to that process' home directory. The slash, /, before **>dd:ftserver** separates the runtime parameters from the OS/390 DFS program parameters. A redirection parameter to the server's own output ddname must be specified in OS/390 DFS. In this example, the redirection of **STDOUT** to **DDNAME=FTSERVER** and the redirection of **STDERR** to **STDOUT** is specified by: **dd:ftserver 2>&1**. It is important to note that OS/390 DFS may, depending on the processes to be created, require additional parameters.

```
$ bos create /./hosts/DFSMVS ftserver simple "ftserver \  
    envvar('_EUV_HOME=/opt/dfslocal/home/ftserver')/ >dd:ftserver 2>&1"
```

The process automatically begins on OS/390 after issuing this command and whenever the **boserver** is restarted.

Example of Creating, Starting, and Stopping the ftserver: An example command to create and start the **ftserver** process on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos create /./hosts/DFSMVS ftserver simple "ftserver \  
    envvar('_EUV_HOME=/opt/dfslocal/home/ftserver')/ >dd:ftserver 2>&1"
```

The **bos create** command updates the **/opt/dfslocal/var/dfs/BosConfig** file. As a result, the **ftserver** process starts on the OS/390 system and whenever the **boserver** is restarted.

An example command to stop the **ftserver** process on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos stop /./hosts/DFSMVS ftserver
```

An example command to restart the **ftserver** process on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos start /./hosts/DFSMVS ftserver
```

Example of Creating, Starting, and Stopping the bakserver: An example command to create and start the **bkserver** process (also known as the **bakserver**) on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos create /./hosts/DFSMVS bakserver simple "bkserver \  
    envvar('_EUV_HOME=/opt/dfslocal/home/bakserver')/ >dd:bkserver 2>&1"
```

Note: Enter the previous command by specifying **bakserver** and **bkserver** exactly as shown.

The **bos create** command updates the **/opt/dfslocal/var/dfs/BosConfig** file. As a result, the **bakserver** starts on the OS/390 system and whenever the **boserver** is restarted.

An example command to stop the **bakserver** process on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos stop /./hosts/DFSMVS bakserver
```

An example command to restarting the **bakserver** process on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos start /./hosts/DFSMVS bakserver
```

Example of Creating, Starting, and Stopping the flserver: An example command to create and start the **flserver** process on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos create /./hosts/DFSMVS flserver simple "flserver \  
    envvar('_EUV_HOME=/opt/dfslocal/home/flserver')/ >dd:flserver 2>&1"
```

The **bos create** command updates the **/opt/dfslocal/var/dfs/BosConfig** file. As a result, the **flserver** process starts on the OS/390 system and whenever the **boserver** is restarted.

An example command to stop the **flserver** process on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos stop /./hosts/DFSMVS flserver
```

An example command to restart the **flserver** process on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos start /./hosts/DFSMVS flserver
```

Example of Creating, Starting, and Stopping the repserver: An example command to create and start the **rpserver** process (also known as the **repserver**) on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos create /./hosts/DFSMVS repserver simple "rpserver \  
  envvar('_EUV_HOME=/opt/dfslocal/home/repserver')/ >dd:rpserver 2>&1"
```

Note: Enter the previous command by specifying **repserver** and **rpserver** exactly as shown.

The **bos create** command updates the **/opt/dfslocal/var/dfs/BosConfig** file. As a result, the **repserver** starts on the OS/390 system and whenever the **boserver** is restarted.

An example command to stop the **repserver** on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos stop /./hosts/DFSMVS repserver
```

An example command to restart the **repserver** on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos start /./hosts/DFSMVS repserver
```

Example of Creating, Starting, and Stopping the upserver: Here is an example of creating and starting an **upserver** process and enabling it to act as the system control machine for the specified administrative files. In this example, the administrative files (**admin.bak**, **admin.bos**, **admin.ft**, **admin.fl**, and **admin.up**) all reside in the directory **/opt/dcelocal/var/dfs/**. To create and start an **upserver** on an OS/390 system with a DCE host name of **DFSMVS**, issue the following command from OMVS:

```
$ bos create /./hosts/DFSMVS upserver simple "upserver \  
  envvar('_EUV_HOME=/opt/dfslocal/home/upserver')/ >dd:upserver 2>&1 \  
  /opt/dcelocal/var/dfs/ admin.bak admin.bos admin.ft admin.fl \  
  admin.up"
```

The **bos create** command updates the **/opt/dfslocal/var/dfs/BosConfig** file. As a result, the **upserver** process starts on the OS/390 system and whenever the **boserver** is restarted.

An example command to stop the **upserver** on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos stop /./hosts/DFSMVS upserver
```

An example command to restart the **upserver** on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos start /./hosts/DFSMVS upserver
```

Example of Creating, Starting, and Stopping the upclient: Here is an example of creating and starting the **upclient** process when the **upclient**, the **boserver**, and the **ftserver** are on the OS/390 system with a DCE host name of **DFSMVS** and the **upserver** is on the system with a DCE host name of *dcehostnameups*. To create and start the **upclient** from OMVS, issue the command:

```
$ bos create ../hosts/DFSMVS upclient simple "upclient \  
  envvar('_EUV_HOME=/opt/dfslocal/home/upclient')/ >dd:upclient 2>&1 \  
  ../hosts/dcehostnameups /opt/dcelocal/var/dfs/admin.bos \  
  /opt/dcelocal/var/dfs/admin.ft -f UpCLog"
```

The **bos create** command updates the */opt/dfslocal/var/dfs/BosConfig* file. As a result, the **upclient** process starts on the OS/390 system and whenever the **boserver** is restarted.

An example command to stop the **upclient** on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos stop ../hosts/DFSMVS upclient
```

An example command to restart the **upclient** on the system with a DCE host name of **DFSMVS** from OMVS is:

```
$ bos start ../hosts/DFSMVS upclient
```

Stopping All the boserver Processes from OMVS

Processes controlled by **boserver** can be shut down using the **bos stop** and **bos shutdown** commands from OMVS. The following command instructs the BOS Server running on **fs3** to stop all **child** processes and continue to run itself:

```
$ bos shutdown ../../abc.com/hosts/fs3
```

For further information, see the *OS/390 Distributed File Service DFS Administration Guide and Reference*.

DFS Server Relationship to DCE

To configure and use the DFS server, DCE must be running on the same host system. The DFS server cannot successfully start if DCE is not running. If DCE processing fails after the DFS server is initialized, the DFS server processing fails.

Appendix A. Example DFS Server Configuration Log File (dfsconf.log)

This appendix shows an example of the contents of an OS/390 DFS configuration log file **dfsconf.log** that is created in the home directory of the user running DFSCONF. The example shows the output of a DFS server configuration, DFS server deconfiguration, DFS client configuration and DFS client deconfiguration in that order. The output shown here is for reference only and the actual output that is created by DFSCONF in the **dfsconf.log** file may not be exactly as shown.

Notice: DFSCONF started at Fri Dec 19 15:37:13 1997.

Notice: Option IOEBMAIN.1 selected.
IOED01033I DFS configuration initiated at Fri Dec 19 15:37:18 1997.

Notice: Panel Data:
 <cellname> = !nomyn.endicott.ibm.com!
 <host name> = !DCEDFS7!
 <dfs server> = !1!
 <backup tape> = !1!
 <backup db> = !1!
 <fileset db> = !-2024565684!

IOED01016I Login for cell_admin at Fri Dec 19 15:37:22 1997.
dce_login cell_admin
EUVS24588I Attention - change current password.
EUVS24577I Login successful.
dce_login cell_admin admin

Beginning configuration of DFS server machine.

IOED01022I Configure Initial Registry Information

IOED01019I Adding necessary principals to registry database for DCEDFS7.
dcecp -c group catalog

dcecp -c group create subsys/dce/dfs-admin -inprojlist yes

dcecp -c group catalog

dcecp -c group create subsys/dce/dfs-fs-servers -inprojlist yes

dcecp -c group list subsys/dce/dfs-admin

dcecp -c group add subsys/dce/dfs-admin -member cell_admin

IOED01023I Creating Security Group
IOED01019I Adding necessary principals to registry database for DCEDFS7.
dcecp -c principal create hosts/DCEDFS7/dfs-server

IOED01020I Creating the dfs-server principal for DCEDFS7.
rgy_edit -up

IOED01021I Modify ACL for DCEDFS7.
dcecp -c acl modify

```
../../nomyn.endicott.ibm.com/sec/principal/hosts/DCEDFS7/dfs-server  
-add {group subsys/dce/dfs-admin rcDnfmag}
```

IOED01024I Configure Initial Name Service Information

Modifying the namespace for DFS server operation

```
dcecp -c directory create ../subsys/dce/dfs
```

Setting ACLs on the new entries for DFS server operation

```
dcecp -c acl modify ../subsys/dce/dfs  
-ic -add {group subsys/dce/dfs-admin rwcidta}
```

```
dcecp -c acl modify ../subsys/dce/dfs  
-io -add {group subsys/dce/dfs-admin rwcidt}
```

```
dcecp -c acl modify ../subsys/dce/dfs  
-add {group subsys/dce/dfs-admin rwcdda}
```

```
dcecp -c acl modify ../fs -e  
-add {group subsys/dce/dfs-admin rwcidt}
```

```
dcecp -c acl modify ../fs -e  
-add {group subsys/dce/dfs-fs-servers rwcidt}
```

Setting Security ACLs

```
acl_edit  
-addr 50d6e004-2b8f-1f97-95dc-001234567890@ncacn_ip_tcp:9.130.79.230  
group/subsys/dce/dfs-admin -m group:acct-admin:rctDnfmM
```

```
acl_edit  
-addr 50d6e004-2b8f-1f97-95dc-001234567890@ncacn_ip_tcp:9.130.79.230  
group/subsys/dce/dfs-fs-servers -m group:acct-admin:rctDnfmM
```

```
acl_edit  
-addr 50d6e004-2b8f-1f97-95dc-001234567890@ncacn_ip_tcp:9.130.79.230  
group/subsys/dce/dfs-fs-servers -m group:subsys/dce/dfs-admin:rctDnfmM
```

Configure Registry Information

```
dcecp -c group add subsys/dce/rpc-server-group  
-member hosts/DCEDFS7/dfs-server
```

Update Name Service Database

Exporting the DFS server endpoint mapper host binding.

```
dcecp -c rpcentry export ../../nomyn.endicott.ibm.com/hosts/DCEDFS7/self  
-i {e1af8308-5d1f-11c9-91a4-08002b14a0fa,3.0}  
-b {ncadg_ip_udp:9.130.79.230#135"}
```

Setting ACL for DFSTRACE Command Suite

```
dcecp -c acl modify ../../nomyn.endicott.ibm.com/hosts/DCEDFS7  
-add {user hosts/DCEDFS7/dfs-server rwdtcia}
```

End configuration of DFS File Server.

Beginning configuration of DFS BAK server machine.

```
dcecp -c group catalog
```

```
dcecp -c group create subsys/dce/dfs-bak-servers -inprojlist yes
```

```

dcecp -c rpcgroup list /.../nomyn.endicott.ibm.com/subsys/dce/dfs/bak

dcecp -c rpcentry create /.../nomyn.endicott.ibm.com/subsys/dce/dfs/bak

dcecp -c rpcprofile add /.../nomyn.endicott.ibm.com/cell-profile
  -member /.../nomyn.endicott.ibm.com/subsys/dce/dfs/bak
  -i {eb814e2a-0099-11ca-8678-02608c2ea96e,4.0} -a bak

dcecp -c acl modify ./subsys/dce/dfs/bak -e
  -add {group subsys/dce/dfs-bak-servers rwct}

acl_edit
  -addr 50d6e004-2b8f-1f97-95dc-001234567890@ncacn_ip_tcp:9.130.79.230
  group/subsys/dce/dfs-bak-servers -m group:acct-admin:rctDnfmM

acl_edit
  -addr 50d6e004-2b8f-1f97-95dc-001234567890@ncacn_ip_tcp:9.130.79.230
  group/subsys/dce/dfs-bak-servers -m group:subsys/dce/dfs-admin:rctDnfmM

dcecp -c rpcgroup add /.../nomyn.endicott.ibm.com/subsys/dce/dfs/bak
  -member /.../nomyn.endicott.ibm.com/hosts/DCEDFS7/self

dcecp -c group add subsys/dce/dfs-bak-servers
  -member hosts/DCEDFS7/dfs-server

```

End configuration of DFS BAK server.

Beginning configuration of DFS FLDB server machine.

```

dcecp -c rpcgroup add /.../nomyn.endicott.ibm.com/fs
  -member /.../nomyn.endicott.ibm.com/hosts/DCEDFS7/self

dcecp -c group add subsys/dce/dfs-fs-servers
  -member hosts/DCEDFS7/dfs-server

```

End configuration of DFS FLDB server.

IOED01060I This configuration was successful for this machine.

Notice: DFSCONF ended at Fri Dec 19 15:43:29 1997.

Notice: DFSCONF started at Fri Dec 19 15:16:31 1997.

Notice: Option IOEBMAIN.2 selected.

IOED01032I DFS deconfiguration initiated at Fri Dec 19 15:16:39 1997.

Notice: Panel Data:

```

<cellname>    = !nomyn.endicott.ibm.com!
<host name>   = !DCEDFS7!
<dfs server>  = !1!
<backup tape> = !1!
<backup db>   = !1!
<fileset db>  = !-2024565684!

```

```
IOED01016I Login for cell_admin at Fri Dec 19 15:16:43 1997.  
dce_login cell_admin  
EUVS24588I Attention - change current password.  
EUVS24577I Login successful.  
dce_login cell_admin admin
```

Beginning deconfiguration of DFS FLDB server machine.

```
dcecp -c rpcgroup remove ././fs  
-member ../nomyn.endicott.ibm.com/hosts/DCEDFS7/self
```

```
dcecp -c group remove subsys/dce/dfs-fs-servers  
-member hosts/DCEDFS7/dfs-server
```

End deconfiguration of DFS FLDB server.

Beginning deconfiguration of DFS BAK server machine.

```
dcecp -c rpcgroup remove ../nomyn.endicott.ibm.com/subsys/dce/dfs/bak  
-member ../nomyn.endicott.ibm.com/hosts/DCEDFS7/self
```

```
dcecp -c group remove subsys/dce/dfs-bak-servers  
-member hosts/DCEDFS7/dfs-server
```

End deconfiguration of DFS BAK server.

Beginning deconfiguration of DFS server machine.

```
dcecp -c group remove subsys/dce/rpc-server-group  
-member hosts/DCEDFS7/dfs-server
```

```
dcecp -c account delete hosts/DCEDFS7/dfs-server
```

```
dcecp -c keytab  
remove ../nomyn.endicott.ibm.com/hosts/DCEDFS7/config/keytab/self  
-member hosts/DCEDFS7/dfs-server
```

```
dcecp -c principal delete hosts/DCEDFS7/dfs-server
```

```
dcecp -c rpcentry unexport ../nomyn.endicott.ibm.com/hosts/DCEDFS7/self  
-i {e1af8308-5d1f-11c9-91a4-08002b14a0fa,3.0}
```

End deconfiguration of DFS File server machine.

IOED01061I This machine has been successfully deconfigured.

Notice: DFSCONF ended at Fri Dec 19 15:20:20 1997.

Beginning configuration of DFS Client machine.

IOED01060I This configuration was successful for this machine.

Notice: DFSCONF ended at Thu Jan 8 15:33:08 1998.

Notice: DFSCONF started at Thu Jan 8 15:32:34 1998.

Notice: Option IOEBMAIN.4 selected.
IOED01032I DFS deconfiguration initiated at Thu Jan 8 15:32:48 1998.

Notice: Panel Data:
 <cellname> = !gdlsst.endicott.ibm.com!
 <host name> = !GD LCSST!
 <client> = !!

Beginning deconfiguration of DFS Client machine.

IOED01061I This machine has been successfully deconfigured.

Appendix B. Directories and Files

This appendix lists the important OS/390 HFS files and subdirectories shipped as part of the OS/390 DFS product. In addition, symbolic links created during the installation process are identified.

Files and Directories in /opt/dfsglobal

Note: The symbolic link **/opt/dfsglobal** refers to the directory **/usr/lpp/dfs/global**.

Table 16 lists the directories and the files in the **/opt/dfsglobal** directory. It also gives a short description of the directories.

<i>Table 16. Files and Directories in /opt/dfsglobal</i>		
File or Directory	Created by	Description
/opt/dfsglobal/bin	Installation	Contains commands: bak, bos, dce_error, dfsexport, fts, growaggr, mapid, newaggr, salvage, scout, and udebug.
/opt/dfsglobal/examples	Installation	Contains example files for devtab, dfstab, ioepdcf, and rfstab. Also contains the example envvar files for the DFS client and server processes. The directory also contains the following program files: asc2eb.cmd and eb2asc.cmd . The asc2eb.cmd program file is used to convert ASCII characters to EBCDIC characters. The eb2asc.cmd program file is used to convert EBCDIC characters to ASCII characters.
/opt/dfsglobal/lib/nls/msg/En_US.IBM-1047	Installation	Contains DFS message catalogs.
/opt/dfsglobal/src/COPYRIGHT.DFS	Installation	Contains the OS/390 DFS copyright information.
/opt/dfsglobal/script/dfs_cpfiles	Installation	Contains the OS/390 shell script for the OS/390 Distributed File Service default configuration files creation program.

Symbolic Links to /etc/dfs Created During Installation

Table 17 lists the frequently used symbolic links created during installation.

<i>Table 17 (Page 1 of 2). Symbolic Links Created During Installation</i>	
Symbolic Link	Linked File
/bin/bak	../usr/lpp/dfs/global/bin/bak
/bin/bos	../usr/lpp/dfs/global/bin/bos
/bin/cm	../usr/lpp/dfs/global/bin/cm
/bin/dfsexport	../usr/lpp/dfs/global/bin/dfsexport
/bin/fts	../usr/lpp/dfs/global/bin/fts
/bin/growaggr	../usr/lpp/dfs/global/bin/growaggr
/bin/mapid	../usr/lpp/dfs/global/bin/mapid

Table 17 (Page 2 of 2). Symbolic Links Created During Installation

Symbolic Link	Linked File
/bin/newaggr	../usr/lpp/dfs/global/bin/newaggr
/bin/salvage	../usr/lpp/dfs/global/bin/salvage
/bin/scout	../usr/lpp/dfs/global/bin/scout
/bin/udebug	../usr/lpp/dfs/global/bin/udebug
/etc/ioepdcf	../etc/dfs/etc/ioepdcf
/opt/dcelocal/etc/CacheInfo	.././../etc/dfs/etc/CacheInfo
/opt/dcelocal/var/dfs	.././../etc/dfs/var/dfs
/opt/dfslocal	../etc/dfs
/opt/dfsglobal	../usr/lpp/dfs/global

Directories in /opt/dfslocal

Notes:

1. The symbolic link **/opt/dfslocal** refers to the directory **/etc/dfs**.
2. Prior to OS/390 Version 2 Release 6, some DFS symbolic links were created to link to files in the **/usr/lpp/dfs/local** path. These symbolic links must now link to files in the **/etc/dfs** path.

Table 18 lists the directories in **/opt/dfslocal** and a brief description of each.

Table 18 (Page 1 of 2). Directories in /opt/dfslocal

Name (Note: /opt/dfslocal-->/etc/dfs)	Created by	Description of Directories
/opt/dfslocal/etc	Installation	Contains the daemon configuration (ioepdcf) file.
/opt/dfslocal/home	Installation	Contains the home directories of the OS/390 DFS daemons (see the following entries).
/opt/dfslocal/home/bakserver	Installation	Home directory of the bakserver process.
/opt/dfslocal/home/boserver	Installation	Home directory of the boserver process.
/opt/dfslocal/home/butc01	Installation	Home directory of the butc01 process.
/opt/dfslocal/home/butc02	Installation	Home directory of the butc02 process.
/opt/dfslocal/home/butc03	Installation	Home directory of the butc03 process.
/opt/dfslocal/home/butc04	Installation	Home directory of the butc04 process.
/opt/dfslocal/home/butc05	Installation	Home directory of the butc05 process.
/opt/dfslocal/home/butc06	Installation	Home directory of the butc06 process.
/opt/dfslocal/home/butc07	Installation	Home directory of the butc07 process.
/opt/dfslocal/home/butc08	Installation	Home directory of the butc08 process.
/opt/dfslocal/home/daemonct	Installation	Home directory of the daemonct process.
/opt/dfslocal/home/dfscm	Installation	Home directory of the dfscm process.
/opt/dfslocal/home/dfscntl	Installation	Home directory of the dfscntl process.
/opt/dfslocal/home/dfsexport	Installation	Home directory of the dfsxport process.
/opt/dfslocal/home/dfskern	Installation	Home directory of the dfskern process.

Table 18 (Page 2 of 2). Directories in /opt/dfslocal

Name (Note: /opt/dfslocal-->/etc/dfs)	Created by	Description of Directories
/opt/dfslocal/home/flserver	Installation	Home directory of the flserver process.
/opt/dfslocal/home/ftserver	Installation	Home directory of the ftserver process.
/opt/dfslocal/home/growaggr	Installation	Home directory of the growaggr command.
/opt/dfslocal/home/newaggr	Installation	Home directory of the newaggr command.
/opt/dfslocal/home/repserver	Installation	Home directory of the rpserver process.
/opt/dfslocal/home/salvage	Installation	Home directory of the salvage command.
/opt/dfslocal/home/upclient	Installation	Home directory of the upclient process.
/opt/dfslocal/home/upserver	Installation	Home directory of the upserver process.
/opt/dfslocal/lib/nls/msg	Installation	Contains the DFS message catalogs.
/opt/dfslocal/syscall	Installation	Used to contain the socket for AFS_SYSCALL.
/opt/dfslocal/var/dfs	Installation	Contains administration and configuration files.
/opt/dfslocal/var/dfs/adm	Installation	Contains log files for the DFS client and server processes.
/opt/dfslocal/var/dfs/backup	Installation	Contains error and log files for the bakserver and butc processes

Customizable Files in /opt/dfslocal

Table 19 on page 150 identifies the files in the /opt/dfslocal subdirectories.

Notes:

1. The symbolic link **/opt/dfslocal** refers to the directory **/etc/dfs**.
2. The symbolic link **/opt/dfsglobal** refers to the directory **/usr/lpp/dfs/global**.

Many of these files are copied from the IBM supplied files in the **/opt/dfsglobal/examples** directory to the applicable operational directory by the **/opt/dfsglobal/scripts/dfs_cpfiles** program which is run as described in Chapter 6, "Post Installation Considerations" on page 41. These files can be updated by an administrator using an editor such as **OEDIT** to specify environment variable (**envvar**) values and other configuration information for the DFS server and DFS client (DFSCM) processes.

Other files are usually created and maintained by administrator commands such as the **bos addadmin** command. These files should only be updated using the commands.

There are also some files in **/opt/dfslocal** subdirectories that are created and maintained by the DFS server and DFSCM process during normal operations. These files should not be directly updated or customized.

Finally, some of the **/opt/dfslocal** files apply to either the DFS support for DCE, the SMB File/Print support, or to both DFS and SMB.

Table 19 on page 150 describes the **/opt/dfslocal** files in this manner.

Table 19 (Page 1 of 2). OS/390 DFS Customizable Files

File (Note: /opt/dfslocal-->/etc/dfs)	Created By	Customizable	Applicability
/opt/dfslocal/etc/CacheInfo	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/etc/ioepdcf	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/home/bakserver/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/bakserver/krb5ccname	bakserver processing	NO	DFS
/opt/dfslocal/home/boserver/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/boserver/krb5ccname	boserver processing	NO	DFS
/opt/dfslocal/home/butc01/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/butc01/krb5ccname	butc01 processing	NO	DFS
/opt/dfslocal/home/butc02/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/butc02/krb5ccname	butc02 processing	NO	DFS
/opt/dfslocal/home/butc03/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/butc03/krb5ccname	butc03 processing	NO	DFS
/opt/dfslocal/home/butc04/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/butc04/krb5ccname	butc04 processing	NO	DFS
/opt/dfslocal/home/butc05/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/butc05/krb5ccname	butc05 processing	NO	DFS
/opt/dfslocal/home/butc06/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/butc06/krb5ccname	butc06 processing	NO	DFS
/opt/dfslocal/home/butc07/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/butc07/krb5ccname	butc07 processing	NO	DFS
/opt/dfslocal/home/butc08/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/butc08/krb5ccname	butc08 processing	NO	DFS
/opt/dfslocal/home/daemonct/envar	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/home/dfscm/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/dfscm/krb5ccname	DFSCM processing	NO	DFS
/opt/dfslocal/home/dfscntl/envar	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/home/dfsexport/envar	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/home/dfskern/dfsldmap	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/dfskern/envar	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/home/dfskern/smbldmap	dfs_cpfiles Program	OEDIT	SMB
/opt/dfslocal/home/flserver/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/ftserver/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/growaggr/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/newaggr/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/repserver/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/salvge/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/upclient/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/home/upserver/envar	dfs_cpfiles Program	OEDIT	DFS
/opt/dfslocal/syscall/rendezvous	Server processing	NO	DFS & SMB
/opt/dfslocal/var/dfs/admin.bak	bos addadmin command	bos addadmin command	DFS
/opt/dfslocal/var/dfs/admin.bos	bos addadmin command	bos addadmin command	DFS
/opt/dfslocal/var/dfs/admin.fl	bos addadmin command	bos addadmin command	DFS
/opt/dfslocal/var/dfs/admin.ft	bos addadmin command	bos addadmin command	DFS
/opt/dfslocal/var/dfs/admin.up	bos addadmin command	bos addadmin command	DFS

| Table 19 (Page 2 of 2). OS/390 DFS Customizable Files

File (Note: /opt/dfslocal-->/etc/dfs)	Created By	Customizable	Applicability
/opt/dfslocal/var/dfs/cmattr	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/var/dfs/devtab	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/var/dfs/dfstab	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/var/dfs/hfstab	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/var/dfs/NoAuth	touch command	OEDIT	DFS
/opt/dfslocal/var/dfs/rfstab	dfs_cpfiles Program	OEDIT	DFS & SMB
/opt/dfslocal/var/dfs/smbtab	dfs_cpfiles Program	OEDIT	SMB

Appendix C. Environment Variables in DFS

Environment variables affect the behavior of the DFS components. In DFS, the **envar** file of a client or server process can be created to contain the declarations of the environment variables.

The DCE environment variables affect the administration of DFS. This appendix describes the DCE and DFS environment variables DFS server processes and DFS client processes use, and tells you how to set them.

Environment variables are set using the following syntax:

VARIABLE_NAME=value

Environment Variables

Table 20 lists the OS/390 DCE and DFS environment variables with their descriptions. See the *OS/390 DCE Administration Guide* for a complete list of DCE environment variables. This table also includes the Distributed File Service environment variables specific for SMB support. You can refer to the *OS/390 Distributed File Service SMB Administration Guide and Reference* for more information.

Notes:

1. Setting these environment variables is optional. In most cases, the default values are used if environment variables are not set.
2. In the following table all the examples should be entered on one line, with no blanks, even though some of them appear on multiple lines.

Table 20 (Page 1 of 19). Environment Variables	
Name	Description
DCE Environment Variables	
_EUV_AUTOLOG	Determines if single sign-on processing is ignored when an OS/390 user invokes an application that invokes the DFS client. The valid value is: NO Do not enable single sign-on processing. Any other value causes DCE autologin processing to be attempted. Note: If _EUV_AUTOLOG=NO is specified, DCE autologin processing does not occur if the DFS namespace is accessed before a dce_login is performed. Refer to the <i>OS/390 Distributed File Service DFS Administration Guide and Reference</i> for more information on DFS client and DFS autologin.
_EUV_ECHO_STDIN	Used by the RPC, CDS, and DTS control programs, as well as the Registry Editor and ACL Editor to echo input commands to the standard output file when these commands are run in batch. The default value is 0 (disabled).
_EUV_ENVAR_FILE	Defines the name of the environment variable file. The default value is \$HOME/envar , where \$HOME is your home directory.
_EUV_FTRACE	Activates function tracing within DCE or user code compiled with the TEST(NONE) compiler option.
_EUV_HOME	Used to override the home directory value that was specified in the POSIX segment of a RACF user ID. The default value is your home directory.
_EUV_SEC_KRB5CCNAME_FILE	Specifies the file that contains the declaration of the KRB5CCNAME variable. The default value is \$HOME/krb5ccname .

Table 20 (Page 2 of 19). Environment Variables

Name	Description
_EUV_SVC_DBG_MSG_LOGGING	Turns debug tracing off.
_EUV_SVC_MSG_LEVEL	A DCE environment variable that sets the minimum severity level of messages that are displayed. For the Distributed File Service, the value VERBOSE is recommended. This is the default value if this envar file is not specified. For more information on this environment variable see the <i>OS/390 DCE Administration Guide</i> .
_EUV_SVC_MSG_LOGGING	A DCE environment variable used to control where messages are routed and displayed. For the Distributed File Service, the value CONSOLE_LOGGING is recommended and supplied in the sample envar file for each of the Distributed File Service processes. For more information on this environment variable see the <i>OS/390 DCE Administration Guide</i> .
BIND_PE_SITE	Determines whether the Security Server is located using the namespace or by reading the pe_site file. The default value is 0 , meaning the Security server is located by using a regular CDS query.
DCE_START_SOCKET_NAME	The path used as the well-known socket name by the DFS servers during daemon initialization. Default Value The default value for the DFS server is /opt/dfslocal/home/dfs kern/ioepk.soc Expected Value Character string. Example DCE_START_SOCKET_NAME= /opt/dfslocal/home/dfscntl/start.sock Where Variable is Used The DFS server daemons.
LIBPATH	The path names used to find DLLs. Default Value None Expected Value Character string Example LIBPATH=/usr/lib:/usr/lpp/Printsrv/lib Where Variable is Used DFSKERN
KRB5CCNAME	Specifies the name of the user's Security credentials cache file. This variable is automatically managed within the DCE library and should not be set by the user. In OS/390 DFS, KRB5CCNAME is not an actual environment variable. However, as KRB5CCNAME is an environment variable in other implementations of DCE, it is included here for reference purposes.
NLSPATH	A POSIX environment variable used by DCE that sets the search path for message catalogs. The default value is: /usr/lib/nls/msg/En_US.IBM-1047/%N: /usr/lib/nls/msg/%L/%N: /usr/lib/nls/msg/prime/%N
SVC_CDS_DBG	Sets the CDS component debug level.
SVC_DTS_DBG	Sets the DTS component debug level.
SVC_PLT_DBG	Sets the Platform component debug level.
SVC_RPC_DBG	Sets the RPC component debug level.
SVC_SEC_DBG	Sets the Security component debug level.
TZ	Sets the time zone used by DTS. The default value is localtime .
DFS Environment Variables	

Table 20 (Page 3 of 19). Environment Variables

Name	Description
_IOE_BACKUP_TAPE_CAPACITY	Specifies the size, in bytes, of the maximum size backup tape data set that is created during backup. It is desirable that this value equal the actual capacity of your tape volumes (or a little less). Default Value 160M (the size of a 3480 tape cartridge) Expected Value Size in bytes, of the maximum size backup tape data set. The value may be a whole positive integer followed by a 'K' (denoting kilobytes), or a whole positive integer followed by an 'M' (denoting megabytes). Example _IOE_BACKUP_TAPE_CAPACITY=180M Where Variable is Used BUTCnn (valid entries for nn are 01 through 08)
_IOE_BOSERVER_ENVAR	Specifies where to find the environment variable file during a boserver restart. Default Value ENVAR('_EUV_HOME=/opt/dfslocal/home/boserver') Expected Value Character string. Example _IOE_BOSERVER_ENVAR=ENVAR('_EUV_HOME=/opt/dfslocal/home/boserver') Where Variable is Used BOSERVER
_IOE_BOSERVER_OUTPUTDD	The data definition name (ddname), statically allocated in the DFS JCL procedure, which will receive the boserver output (before it switches to its log file). Default Value BOSERVER Expected Value Character string, eight characters or less. Example _IOE_BOSERVER_OUTPUTDD=BOSRV Where Variable is Used BOSERVER
_IOE_BOSERVER_PGMNAME	The OS/390 PDS member name of the boserver program to be used when restarting the boserver process. Default Value BOSERVER Expected Value Character string, eight characters or less. Example _IOE_BOSERVER_PGMNAME=NEWBOSRV Where Variable is Used BOSERVER
_IOE_BUTC_DISKUNIT	The unit name to be used when dynamically allocating a data definition for disk data sets. This is the unit name passed to dynamic allocation when allocating the disk device. Default Value SYSDA Expected Value Character string, six characters or less. Example _IOE_BUTC_DISKUNIT=3390 Where Variable is Used BUTCnn (valid entries for nn 01 through 08).

Table 20 (Page 4 of 19). Environment Variables

Name	Description
_IOE_BUTC_DUMP_HLQ	The high level qualifier for the dump (backup) data set names. Default Value DFSBKUP Expected Value Character string, 17 characters or less. Example _IOE_BUTC_DUMP_HLQ=DFSBKUP.VER1 Where Variable is Used BUTCnn (valid entries for nn 01 through 08).
_IOE_BUTC_KSDS	The name of the virtual storage access method (VSAM) DFS tape backup management file. Default Value DFSBUTC.KSDS.LABEL Expected Value Character string, 44 characters or less. Example _IOE_BUTC_KSDS=DFSBKUP.KSDS.LABEL Where Variable is Used BUTCnn (valid entries for nn 01 through 08).
_IOE_BUTC_TAPEUNIT	The unit name to be used when dynamically allocating a data definition for tape drives. This is the unit name passed to dynamic allocation when allocating the tape device. Default Value 3490 Expected Value Character string, six characters or less. Example _IOE_BUTC_TAPEUNIT=CART Where Variable is Used BUTCnn (valid entries for nn 01 through 08).
_IOE_CFG_CELL_ID	Specifies the DCE principal name of the administrator who is performing the OS/390 DFS configuration. Default Value cell_admin Expected Value Character string, 132 characters or less. Example _IOE_CFG_CELL_ID=cell_admin Where Variable is Used DFSCONF
_IOE_CFG_INFORM_LEVEL	Specifies the level of information that is displayed on the screen from the configuration of the host system. The valid values are: 0 Only messages regarding the progress of configuration are displayed. 1 Progress messages and the commands used by the configuration written. This is the default value. 2 Progress messages, command, and the output from most commands are written. Default Value 1 Expected Value 0, 1, or 2. Example _IOE_CFG_INFORM_LEVEL=2 Where Variable is Used DFSCONF

Table 20 (Page 5 of 19). Environment Variables

Name	Description
_IOE_CFG_LOG_FILE	Specifies the name of the configuration log file. Default Value dfsconf.log and is initially located in the home directory of the user running DFSCONF. Expected Value Character string, 132 characters or less. Example _IOE_CFG_LOG_FILE=/home/dfsadmin/dfsconf.log Where Variable is Used DFSCONF
_IOE_CM_ATTRIBUTES_FILE	Specifies the name of the file that contains the definition of file name suffixes that controls whether file data should be translated by the DFS client. Default Value None Expected Value Character string, 132 characters or less. Example _IOE_CM_ATTRIBUTES_FILE=/etc/httpd.conf Where Variable is Used DFS client processes Notes This file contains AddType statements in the same format as the WebSphere Application Server httpd.conf file. All statements other than AddType are ignored. IBM supplies an example file in /opt/dfsglobal/examples/cmattr. This file can be copied and modified appropriately. If you do not use the Webserver's http.conf file, it is recommended that you store the file in the /opt/dfslocal/home/dfscm directory. If the _IOE_CM_ATTRIBUTES_FILESETS environment variable (in the DFS client processes) exists, it limits the control of the file name suffix translation to the filesets listed in that file.
_IOE_CM_ATTRIBUTES_FILESETS	Specifies the name of the file that contains the definition of filesets that are under control of the cmattr file. It limits the effect of the cmattr file. Only data in these filesets are translated based on the cmattr file. Default Value None Expected Value Character string, 132 characters or less. Example _IOE_CM_ATTRIBUTES_FILESETS=/opt/dfslocal/home/dfscm/cmattr_filesets Where Variable is Used DFS client processes Notes If the cmattr_filesets file is empty, then no filesets are translated based on the cmattr file. If the cmattr_filesets file is non-existent (and a cmattr file exists), then all filesets are translated based on the cmattr file.

Table 20 (Page 6 of 19). Environment Variables

Name	Description
_IOE_CM_DIRCACHE_SIZE	Specifies the size, in bytes, of the directory cache for the DFS client. Default Value 1M Expected Value Size, in bytes, of the cache. The value may be a whole positive integer followed by a 'K' (denoting kilobytes), or a whole positive integer followed by an 'M' (denoting megabytes). Example _IOE_CM_DIRCACHE_SIZE=2M Where Variable is Used DFS client processes. Notes The minimum allowable size of the DFS client directory cache is one megabyte. This value will be used for the directory cache even if a lower value is specified. The maximum size is limited only by the virtual storage size of DFSCM. Values for the directory cache size will be rounded upward to a 128K boundary.
_IOE_CM_LOGIN_PROCESSES	The DFS environment variable controls the number of ioelogin processes set up to handle DCE login requests. Default Value 1 Expected Value A number greater than zero. Example _IOE_CM_LOGIN_PROCESSES=2 Where Variable is Used DFS client processes.
_IOE_CM_PARMS	The DFS client initialization parameters or options that take effect when the DFS client is started. The possible parameters are a subset of the dfsbind and the dfsd command options described in the <i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>. Default Value For the default values, see the dfsbind and dfsd command options in the "Commands" section of the <i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>. Expected Value Character string of DFS client initialization parameters (up to 1024 characters in length). Example _IOE_CM_PARMS=-mountfilesystem IOE_DFS_CLIENT_DATA -cachedir 11 Where Variable is Used DFS client processes.
_IOE_CM_REQUEST_THREADS	The DFS environment variable controls the number of DFSCM ioedfsd process threads set up to handle file requests. Default Value 5 Expected Value A number greater than zero. Example _IOE_CM_REQUEST_THREADS=7 Where Variable is Used DFS client processes.

Table 20 (Page 7 of 19). Environment Variables

Name	Description
_IOE_DAEMONS_IN_AS	The DFS environment variable controls whether the DFSKERN process runs in its own address space or in the DFS Server Address Space. Default Value The default is ". If " is specified or the environment variable is not specified, DFSKERN runs in the DFS Server Address Space. Expected Value DFSKERN or " Example _IOE_DAEMONS_IN_AS=DFSKERN In this case, DFSKERN runs in its own address space. Where Variable is Used DFSCNTL
_IOE_DFS_MODIFY_PATH	The path used as the well-known socket name by the DFS programs when registering with DFSCNTL to receive modify commands (F DFS, QUERY DFSKERN). Default Value The default value of the DFS server is /opt/dfslocal/home/dfscntl/modify.rendezvous Expected Value Character string. Example _IOE_DFS_MODIFY_PATH= /opt/dfslocal/home/dfscntl/test.rendezvous Where Variable is Used All programs. If the default is not being used, this environment variable must be coded in the envvar file for each process.
_IOE_DIRECTORY_CACHE_SIZE	The number of 512 byte blocks used to cache HFS and RFS directory entries. Default Value 768 Expected Value The value specified must be a numeric value greater than or equal to 768. Example _IOE_DIRECTORY_CACHE_SIZE=1024 Where Variable is Used DFSKERN
_IOE_EPI_CACHE_SIZE	The size, in bytes, of the DCE Local File System metadata buffer cache. This environment variable is used to increase the size of the buffer cache and to potentially improve performance. Default Value one megabyte. This value is also the minimum value. Expected Value Size, in bytes, of the cache. The value expected may be a whole positive integer followed by a 'K' (denoting kilobytes), or a whole positive integer followed by a 'M' (denoting megabytes). Example _IOE_EPI_CACHE_SIZE=8M Where Variable is Used DFS client processes and DFSKERN Notes The minimum allowable size of the DCE Local File System buffer cache is one megabyte. This value will be used for the buffer cache even if a lower value is specified. The maximum size is limited only by the virtual storage size of DFSKERN. Values for the buffer cache size will be rounded upward to a 128K boundary.

Table 20 (Page 8 of 19). Environment Variables

Name	Description
_IOE_HFS_ATTRIBUTES_FILE	Specifies the pathname of the hfsattr file that contains the definition of file name suffixes that controls whether file data should be translated by the DFS server. Default Value None. Expected Value Character string, 132 characters or less. Example _IOE_HFS_ATTRIBUTES_FILE=/etc/httpd.conf Where Variable is Used DFSKERN Notes This file contains AddType statements in the same format as the IBM HTTP Server's http.conf file. All statements other than AddType are ignored. IBM supplies an example file in /opt/dfsglobal/examples/cmattr. This file can be copied and modified appropriately. (Only one example file is provided for cmattr and hfsattr since they are the same format.) If you do not use the Webserver's httpd.conf file, it is recommended that you store this file in the /opt/dfslocal/home/dfskern directory.

Table 20 (Page 9 of 19). Environment Variables

Name	Description
_IOE_HFS_TRANSLATION	A DFS server variable that controls the conversion of HFS data to the appropriate data format. Converts incoming data from ASCII ISO 8859-1 to the local OS/390 code page. Converts outgoing data from the local OS/390 code page to ASCII ISO 8859-1. Default Value Off Expected Value On, Off, or Auto On means that all data will be translated. Off means that no data will be translated. Auto means that data that is received by (read) or sent to (written) the DFS server is examined to determine if the data is text or not. When data is received by (read) the DFS server, the first 255 bytes of data are compared against a table of valid text (EBCDIC) characters. If all 255 characters are deemed to be valid text characters, the file is marked for translation and all data is translated. Otherwise, the data is not translated. When data is sent from (written) the DFS server, the first 255 bytes of data are compared against a table of valid text (ASCII) characters. If all 255 characters are deemed to be valid text characters, the file is marked for translation and all data is translated. Otherwise, the data is not translated. Example _IOE_HFS_TRANSLATION=ON Where Variable is Used DFSKERN Notes For HFS, the following sequence is used by the DFS server to determine whether to translate file data: If the file exists and has a non-zero file format attribute (ST_FILEFMT) field in the BPXYSTAT structure, then a value of 1 (FTFFBINARY) causes no translation to be done. A value greater than 1 causes translation to be done. If the file does not exist or has a zero file format attribute, then if an hfsattr file is specified (by the DFSKERN _IOE_HFS_ATTRIBUTES_FILE environment variable), and the file name suffix matches a directive in the hfsattr file, then that controls whether translation occurs. The fileset's devtab translation parameter (text, binary, or auto) is used when no hfsattr file is specified, or if the file name suffix does not match any of the directives in the hfsattr file, or the file name has no suffix. The DFSKERN _IOE_HFS_TRANSLATION environment variable is used if the fileset's devtab entry does not have a translation parameter. The ST_FILEFMT field in the BPXYSTAT structure for HFS files is set if it is not already set and either an hfsattr file was used to determine whether to translate or auto (on the fileset or globally) was used to determine whether to translate. This variable does not apply to OS/390 DFS client processing. DFS client translation is controlled by the dfsd command -translation option described in the <i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>.

Table 20 (Page 10 of 19). Environment Variables

Name	Description
_IOE_LFS_SYNC_INTERVAL	The number of seconds between synchronization operations for the DCE Local File System. Default Value 30 Expected Value Numeric, greater than zero. Example _IOE_LFS_SYNC_INTERVAL=60 Where Variable is Used DFSKERN
_IOE_MVS_DFSDFLT	The name of the RACF-defined anonymous user that is associated with unauthenticated DCE users attempting access to exported HFS files. This ID must be RACF-defined with an OS/390 UNIX segment. Default Value There is no default value for this variable. Expected Value Character string, eight characters or less. Example _IOE_MVS_DFSDFLT=DFSDFLT Where Variable is Used DFSKERN (used when exporting HFS file systems).
_IOE_MVS_IDMAP	The name of the identity mapping output file that establishes the relationship between DCE user IDs and OS/390 user IDs (Resource Access Control Facility (RACF)). Default Value There is no default value for this variable. Expected Value Character string describing the map file to be used. Example _IOE_MVS_IDMAP=/opt/dfslocal/home/dfskern/mapid.output Where Variable is Used DFSKERN (used when exporting HFS file systems).
_IOE_MVS_IDMAP_SAF	The DFS Server variable establishes whether a System Access Facility (such as, RACF) is used for UUID to OS/390 ID mapping (YES) or if the identity mapping output file is used (NO). If YES is specified, the _IOE_MVS_IDMAP environment variable is ignored. Default Value NO Expected Value NO or YES Example _IOE_MVS_IDMAP_SAF=YES Where Variable is Used DFSKERN
_IOE_MVS_SERVER	The name of the OS/390 server used for identity mapping and registering with OS/390 UNIX as a DFS server. Default Value DFS Expected Value Character string, maximum 32 characters. Example _IOE_MVS_SERVER=MVS/DFS01 Where Variable is Used DFSKERN

Table 20 (Page 11 of 19). Environment Variables

Name	Description
_IOE_PROTOCOL_RPC	An environment variable that controls whether the DCE DFS protocol is supported (using DCE RPC). Default Value ON Expected Value ON or OFF Example _IOE_PROTOCOL_RPC=OFF Where Variable is Used DFSKERN Notes This environment variable also affects DCE applications and commands. It should not be used in any other processes including shell user processes.
_IOE_PROTOCOL_SMB	An environment variable that controls whether the SMB protocol is supported (using TCP/IP). Default Value OFF Expected Value ON or OFF Example _IOE_PROTOCOL_SMB=ON Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_RFS_ALLOC_TIMEOUT	Specifies the time period (in seconds) that an RFS data set will remain allocated after there has been no access to the data set through the DFS/SMB server. After this time period, the data set will be deallocated and will be available to other applications such as ISPF. Default Value 300 (5 minutes) Expected Value A number greater than or equal to 30. Example _IOE_RFS_ALLOC_TIMEOUT=600 Where Variable is Used DFSKERN
_IOE_RFS_ATTRIBUTES_FILE	The name of the file that contains the table for describing the attributes used to manipulate record files in the DFS Server. The value can be the name of the HFS file, a fixed-block partitioned data set, or a fixed-block sequential data set with a record length of 80. Default Value /opt/dfslocal/var/dfs/rfstab Expected Value Character string describing the attributes file being used (maximum 255 characters). Example _IOE_RFS_ATTRIBUTES_FILE=/'NFSADMIN.NFSS(NFSSATT)' Where Variable is Used DFSKERN Notes This DFS server variable can be overridden for specific filesets in the devtab file. For more information, see the <i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>.

Table 20 (Page 12 of 19). Environment Variables

Name	Description
_IOE_RFS_STATUS_REFRESH_TIME	The DFS Server refreshes its cache of exported record data set names and attributes at a regular interval. The time is specified in seconds. Default Value 600 seconds (10 minutes) Expected Value A number greater than zero. Example _IOE_RFS_STATUS_REFRESH_TIME=360 Where Variable is Used DFSKERN
_IOE_RFS_TRANSLATION	A DFS server variable that controls the conversion of RFS data to the appropriate data format. Converts incoming data from ASCII ISO 8859-1 to the local OS/390 code page. Converts outgoing data from the local OS/390 code page to ASCII ISO 8859-1. Default Value Off Expected Value On or Off Example _IOE_RFS_TRANSLATION=ON Where Variable is Used DFSKERN Notes This DFS server variable can be overridden for specific filesets in the devtab file. If an individual fileset is set to binary, in the devtab file, no translation will be performed. For more information, see the <i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>. This variable does not apply to OS/390 DFS client processing. DFS client translation is controlled by the dfsd command -translation option described in the <i>OS/390 Distributed File Service DFS Administration Guide and Reference</i>.
_IOE_RFS_WORKER_THREADS	A DFS server variable that controls the number of threads to be started in dfskern to service open and close requests for RFS files. Default Value 1 Expected Value A number greater than zero. Example _IOE_RFS_WORKER_THREADS=3 Where Variable is Used DFSKERN
_IOE_SERVER_NAME	The network name of the system. This environment variable should be used if there are several network connections on the OS/390 system and the primary connection is not being used for the DFS system. Default Value The name that is returned from the TCP/IP function, gethostname(). The name is the default name as determined from the TCP/IP TCPDATA file and may not reflect the actual network name for the device being used for the DFS network. Expected Value Character string containing the network name of the system without a domain (note that TCP/IP returns the name in upper case so this name should be in upper case also). Example _IOE_SERVER_NAME=RANDOM Where Variable is Used All programs that use the TCP/IP function, gethostname().

Table 20 (Page 13 of 19). Environment Variables

Name	Description
_IOE_SMB_BROWSE_INTERVAL	Specifies the Browser announcement interval (in milliseconds). Default Value 72000 (1.2 minutes) Expected Value A number between 0 and 720000 (12 minutes). Example _IOE_SMB_BROWSE_INTERVAL=90000 Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_CALLBACK_POOL	Specifies the number of secondary pool threads for processing SMB callback requests. Default Value 2 Expected Value A number greater than 0. Example _IOE_SMB_CALLBACK_POOL=3 Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_CLEAR_PW	Specifies the OS/390 SMB server policy for flowing the SMB password in the clear. REQUIRED specifies that passwords in the clear are required. ALLOWED specifies that passwords in the clear are allowed. Authentication is attempted using encrypted passwords if the client supports it. Otherwise, authentication is attempted assuming an unencrypted (clear text) password was used. NOTALLOWED specifies that passwords in the clear are not allowed. Authentication is attempted using encrypted passwords only. Default Value REQUIRED Expected Value REQUIRED, ALLOWED, NOTALLOWED Example _IOE_SMB_CLEAR_PW=NOTALLOWED Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_COMPUTER_NAME	Specifies the name to be used by SMB redirectors (that is, clients) to contact this server. If a Windows Internet Naming Service (WINS) server is available (see the _IOE_SMB_PRIMARY_WINS environment variable), the SMB computer name is used to identify this server to the WINS server. Default Value The TCP/IP hostname of this system. Expected Value Character string, 15 characters or less. Example _IOE_SMB_COMPUTER_NAME=OS390DATA1 Where Variable is Used DFSKERN Note This is used for SMB support only.

Table 20 (Page 14 of 19). Environment Variables

Name	Description
_IOE_SMB_DESCRIPTION	Specifies the description of this server that appears on the PC. Default Value DFS/MVS CIFS Server Expected Value Character string, 40 characters or less. Example _IOE_SMB_DESCRIPTION=OS390 File Server Note in this example, there are embedded blanks. Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_DIR_PERMS	The permissions for a directory created by the SMB server at the request of an SMB client. Default Value 777 Expected Value The value specified must be a three digit numeric value where each digit is greater than or equal to 0 and less than or equal to 7. Example _IOE_SMB_DIR_PERMS=755 Where Variable is Used DFSKERN Note If the directory is specified as Read-only, the write permissions (222) will be turned off. In this case, 755 would become 555.
_IOE_SMB_DOMAIN_NAME	Specifies the name to be used as the Domain name for this server. Default Value None Expected Value Character string, 15 characters or less. Example _IOE_SMB_DOMAIN_NAME=OS/390DOMAIN1 Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_FILE_PERMS	The permissions for a file created by the SMB server at the request of an SMB client. Default Value 777 Expected Value The value specified must be a three digit numeric value where each digit is greater than or equal to 0 and less than or equal to 7. Example _IOE_SMB_FILE_PERMS=750 Where Variable is Used DFSKERN Note If the file is specified as Read-only, the write permissions (222) will be turned off. In this case, 750 would become 550.

Table 20 (Page 15 of 19). Environment Variables

Name	Description
_IOE_SMB_IDMAP	Specifies the location of the smbidmap file. The smbidmap file contains the mapping of SMB user IDs to OS/390 user IDs. Default Value None Expected Value Character string specifying the path name of the smbidmap file. Example _IOE_SMB_IDMAP=/opt/dfslocal/home/dfskern/smbidmap Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_IDLE_TIMEOUT	Specifies how long (in seconds) an SMB session can remain inactive before it is terminated. Default Value 400 Expected Value A number between 0 and 4294967295. Example _IOE_SMB_IDLE_TIMEOUT=4000 Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_MAIN_POOL	Specifies the number of primary pool threads for processing SMB requests. Default Value 14 Expected Value A number greater than 0. Example _IOE_SMB_MAIN_POOL=20 Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_MAXXMT	Specifies the maximum server buffer size that is returned on the SMB Server Negotiate response. Default Value 65535 bytes Expected Value A number between 1024 and 65535. Example _IOE_SMB_MAXXMT=8192 Where Variable is Used DFSKERN Note This is used for SMB support only.

Table 20 (Page 16 of 19). Environment Variables

Name	Description
_IOE_SMB_NT_SMBS	Specifies whether new SMBs in the NT protocol dialect are to be accepted from PC clients. There may be some workloads that perform better with this set to OFF. Default Value ON Expected Value ON or OFF Example _IOE_SMB_NT_SMBS=OFF Where Variable is Used DFSKERN
_IOE_SMB_OPLOCK_TIMEOUT	Specifies the Opportunistic Lock Timeout period (in seconds) Default Value 300 Expected Value A number between 0 and 4294967295. Example _IOE_SMB_OPLOCK_TIMEOUT=60 Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_OPLOCKS	Specifies whether the server allows clients to use Opportunistic Locks on files. This allows some SMB clients to cache data at the client. This can improve performance. Default Value ON Expected Value ON or OFF Example _IOE_SMB_OPLOCKS=OFF Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_PRIMARY_WINS	Specifies the IP address of the Windows Internet Naming Service (WINS) server that this server announces itself to and forwards WINS requests to. Default Value None Expected Value An IP address (<i>n.n.n.n</i> where <i>n</i> is a number between 0 and 255). Example _IOE_SMB_PRIMARY_WINS=9.120.44.55 Where Variable is Used DFSKERN Note This is used for SMB support only.

Table 20 (Page 17 of 19). Environment Variables

Name	Description
_IOE_SMB_PROTOCOL_LEVEL	Specifies the highest level of SMB protocol dialect that will be negotiated with the PC client. NT is the highest dialect that the SMB server supports. LANMAN is the next lower dialect supported by the SMB server. Normally, this value should remain NT. Default Value NT Expected Value NT or LANMAN Example _IOE_SMB_PROTOCOL_LEVEL=LANMAN Where Variable is Used DFSKERN
_IOE_SMB_RAW	Specifies whether raw mode is supported in the SMB Server Negotiate response. Default Value ON Expected Value ON or OFF Example _IOE_SMB_RAW=OFF Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_SCOPE	Specifies the Scope ID for the Windows Internet Naming Service (WINS) server. The Scope ID defines a group of computers that recognize a registered NetBOIS name. (This should normally be omitted.) Default Value None Expected Value Character string, 224 characters or less. Example _IOE_SMB_SCOPT=MYDEPARTMENTSCOPE Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_SECONDARY_WINS	Specifies the IP address of the Windows Internet Naming Service (WINS) server that this server announces itself to and forwards WINS requests to if the Primary WINS server does not respond. Default Value None Expected Value An IP address (<i>n.n.n.n</i> where <i>n</i> is a number between 0 and 255). Example _IOE_SMB_SECONDARY_WINS=9.120.66.77 Where Variable is Used DFSKERN Note This is used for SMB support only.

Table 20 (Page 18 of 19). Environment Variables

Name	Description
_IOE_SMB_TOKEN_FILE_MAX	Specifies the maximum number of files that the SMB token cache should keep tokens for. Default Value 1024 Expected Value A number greater than or equal to 1024. Example _IOE_SMB_TOKEN_FILE_MAX=2048 Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_SMB_WINS_PROXY	Specifies whether this server can act as a Windows Internet Naming Service (WINS) server proxy. WINS requests sent to this server will be forwarded to a WINS server on another computer (see the _IOE_SMB_PRIMARY_WINS environment variable). Default Value OFF Expected Value ON or OFF. Example _IOE_SMB_WINS_PROXY=ON Where Variable is Used DFSKERN Note This is used for SMB support only.
_IOE_TKM_MAX_TOKENS	Specifies the maximum number of tokens that can be held in the DFS server memory. Default Value 1024 Expected Value A positive number. The minimum value is 1024. Example _IOE_TKM_MAX_TOKENS=2048 Where Variable is Used DFSKERN
_IOE_TKCGLUE_CACHE_SIZE	Specifies the number of file tokens for local HFS access that can be cached. Default Value 4096 Expected Value A number greater than or equal to 4096. Example _IOE_TKMGLUE_CACHE_SIZE=8192 Where Variable is Used DFSKERN
_IOE_TKMGLUE_SERVER_THREADS	The number of threads to be started in dfskern to service token requests from the glue layer. Default Value 5 Expected Value The value specified must be a numeric value greater than or equal to 5. Example _IOE_TKMGLUE_SERVER_THREADS=8 Where Variable is Used DFSKERN

Table 20 (Page 19 of 19). Environment Variables

Name	Description
_IOE_VM_CACHE_SIZE	Specifies the size, in bytes, of the virtual memory used by the DFS client and DFSKERN for the DCE Local File System. Default Value 1M Expected Value A positive number. The value may be a whole positive integer followed by a 'K' (denoting kilobytes), or a whole positive integer followed by an 'M' (denoting megabytes). Example _IOE_VM_CACHE_SIZE=2M Where Variable is Used Can be specified independently in the DFS client processes and DFSKERN Notes The virtual memory is used for data in the DCE Local File System. Metadata for the DCE Local File System is cached under control of the _IOE_EPI_CACHE_SIZE environment variable.
_IOE_VM_MAX_FILES	Specifies the maximum number of files that can be contained in the virtual memory used by the DFS client and DFSKERN for the DCE Local File System. Default Value 1024 Expected Value A positive number. The minimum value is 1024. Example _IOE_VM_MAX_FILES=2048 Where Variable is Used Can be specified independently in the DFS client processes and DFSKERN.
_IOE_VNODE_CACHE_SIZE	The size of the HFS and RFS vnode cache, that is, the number of vnodes. Default Value 4096 Expected Value The value specified must be a numeric value greater than or equal to 2048. Example _IOE_VNODE_CACHE_SIZE=6144 Where Variable is Used DFSKERN

Appendix D. DFS/SMB Performance and Tuning Guidelines

This appendix describes performance and tuning guidelines.

Who Should Read This

This information is provided for administrators of the Distributed File Service (DFS/SMB) server or client for OS/390. Customers of DFS/SMB that run large installations may need to tune the server or the client for optimal performance.

Introduction

The Distributed File Service (DFS/SMB) performance, like most file system clients and servers, is dependent on many factors. DFS/SMB support provides performance information to help the administrator determine sources of bottlenecks and determine the affect of a hardware or software change on DFS/SMB performance. Environment variables and the output of QUERY commands provide DFS/SMB with many tuning knobs that can be adjusted. Proper settings for DFS tuning options is essential.

In this section, the DFS server refers to the file server that runs to support either OS/390 DCE DFS or SMB clients (or both). The DFS client refers to the OS/390 DCE DFS client or a DCE DFS client running on a different system or platform.

This document provides you with a description of the various pieces of performance information you can extract from the DFS/SMB file server or DFS client. You can use this information in determining the source of a bottleneck. This document also provides information on the various DFS file server tuning options, how to set them, and how the settings may affect performance or storage consumption. Percentages listed in this appendix are guidelines. Attainable values may depend on your particular installation. Figure 11 on page 174 displays the DFS/SMB server components.

DFS/SMB Server Components

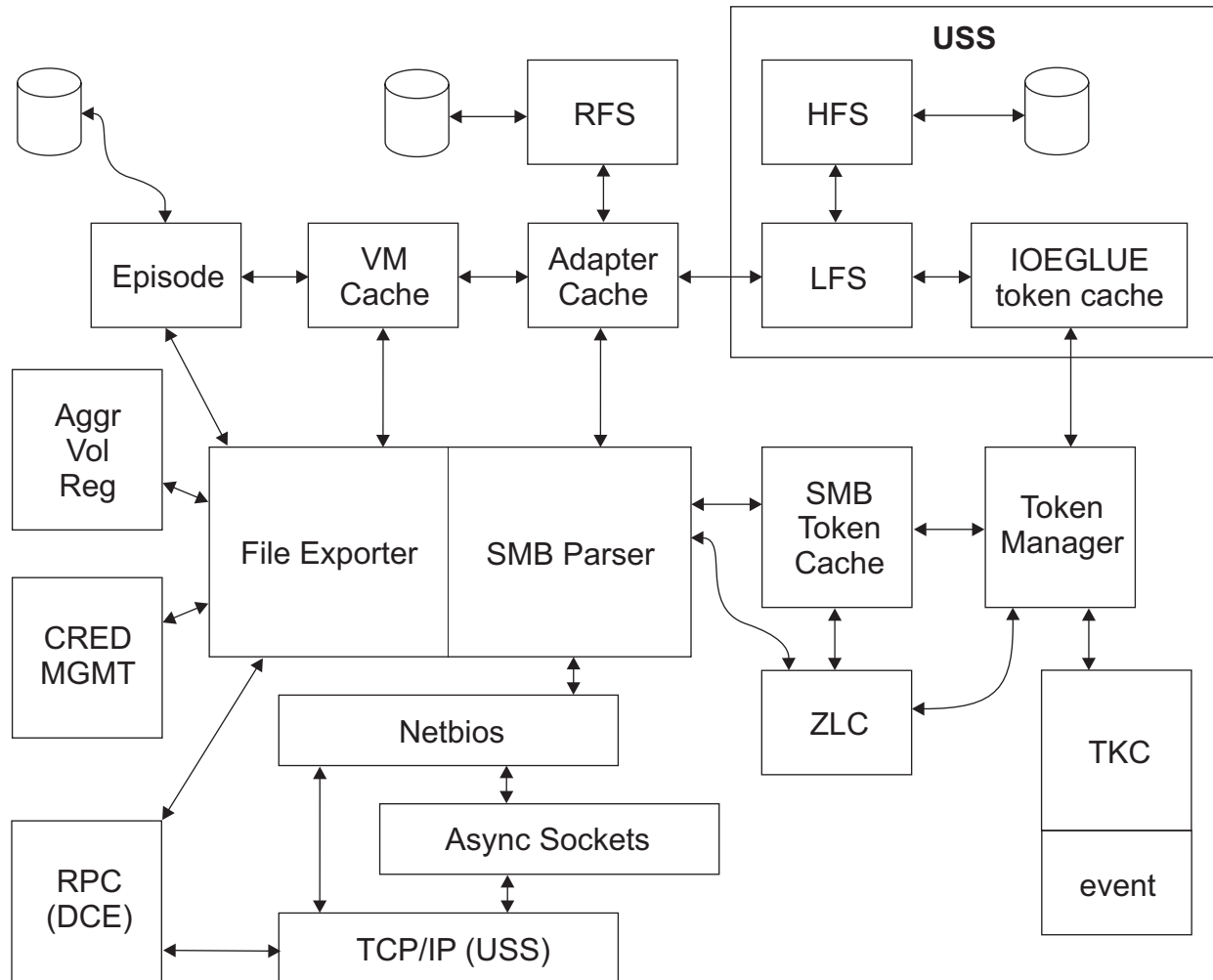


Figure 11. DFS/SMB Server Components

Tuning Considerations and Recommendations

DFS/SMB is a large and complex product with many components. OS/390 Distributed File Service provides a QUERY command that provides information for each of these components. It can be used by IBM service personnel if a performance problem is reported. A subset of this information is useful to an administrator in gauging DFS performance and determining the affect of a change to a tuning option.

Tuning DFS/SMB means:

- Determining the size of thread pools responsible for handling work
- Determining the amount of storage to reserve for various types of caches
- Balancing the data I/O load among disks and the networks attached to the machine.

Many of the DFS default settings or values work for most installations, but there are some defaults that may not be sufficient for larger installations. The most important tuning options are described here.

Threading: If DFS has a unit of work that needs to be dispatched on a processing thread and there are no available threads, that work needs to be placed on a queue to wait for an available service thread. The queue wait time increases server or client response time, therefore ensuring that DFS has enough defined processing threads. DFS default values for the number of processing threads is sufficient for smaller installations, but may be inadequate for larger production use. Each processing thread requires a certain amount of storage, therefore increasing the number of threads increases the DFS storage requirements.

In the sections that follow, the tuning options that control threading are described.

_IOE_SMB_MAIN_POOL

This dfskern environment variable controls the number of threads available to process incoming SMBs for the DFS file server. This is an important tuning option for customers running SMB workloads.

_IOE_RFS_WORKER_THREADS

This dfskern environment variable controls the number of threads available to process RFS file system dataset open or close requests. This is important for customers using RFS on a larger scale.

_IOE_CM_REQUEST_THREADS

This dfskern environment variable controls the number of threads available to process local user requests that requires an RPC. This is important for DFS client workloads where a large number of users are accessing the DFS file space at one time.

-mainprocs

This dfskern command line option controls the number of threads available to process remote DCE/RPC requests for the DFS file server. This is an important tuning option for customers running DCE/DFS workloads. It also controls the number of threads available to process asynchronous file RPCs for DFS client workloads.

Caching: DFS clients cache data to reduce the need to make RPC calls to remote servers. RPCs take a "long" time to complete relative to the amount of time it takes to process a client request when the information is in the DFS client cache. Ensuring the caches are large enough to support the given workload is important to providing good performance for DFS client workloads.

DFS servers cache file system data in memory to reduce the need to perform disk I/O and to allow disk I/O to be performed more efficiently. DFS clients that use disk caching also need to determine the amount of cache storage for the LFS utility aggregate used to house the client cache files. Therefore the LFS cache options for the server are also relevant to DFS client installations that use disk caching.

Providing sufficient caches is essential to good performance. The following are some DFS client tuning options:

-blocks Number of 1K blocks in the DFS client file cache.

-files Number of files to cache. This is the number of client files used for disk caching, not relevant for memory cache clients.

-stat This sets the size of the file status information cache.

-memcache

This parameter indicates if a memory cache is used instead of a disk cache. The default is disk cache.

-namecachesize

This is the size of the name lookup cache.

| **_IOE_CM_DIRCACHE_SIZE**

| This dfscm environment variable sets the size of the client cache used to store directory data.

| For file servers or clients using disk caching, the following tuning options are important:

| **_IOE_VM_CACHE_SIZE**

| This environment variable indicates how much file data to cache for DFS servers or DFS clients using disk caching. This is applicable to all server workloads, SMB or DCE, and with all file systems, HFS, RFS, and LFS.

| **_IOE_EPI_CACHE_SIZE**

| This environment variable indicates how much file system metadata to cache in memory for DFS servers or DFS clients using disk caching. This is applicable to all DFS client disk cache workloads and all server LFS workloads.

| **bufhigh** This dfskern environment variable indicates how much data to store in the RFS dataset cache. It is used for RFS workloads. High RFS file read/write activity may require an increase in the size of this cache.

| **I/O Balancing:** There are no server tuning options that can be used to balance I/O among disks or networks. That is a function of the physical placement of the hardware and where file systems reside. However, the DFS/SMB QUERY commands can possibly point out a bottleneck in a particular disk or the overall network. I/O response times reported by DFS/SMB may show the affect of a bottleneck or the affect of the removal of a bottleneck.

| **QUERY Command**

| The DFS file server and client provide a query command that is used to capture various performance information. The output of the QUERY command goes to the system log and the job output. The syntax of the command for the DFS file server is:

| F DFS,SEND DFSKERN,QUERY,<REPORT>

| The syntax of the query command for the DFS client is:

| F DFSCM,QUERY,<REPORT>

| Where <REPORT> is the name of a DFS performance report. The following is a list of all the available DFS/SMB performance reports:

| **Reports Available for the DFS/SMB Server**

| The following reports are for the DFS/SMB Server.

| **AFS4INT** This report shows the number and type of remote DCE/DFS client RPC requests made to the server, the average DFS file server response times to process those requests, and the amount of bytes transferred to or from the file server.

| **SMBINT** For the DFS file server, this report shows the number and type of remote Windows client SMB file requests made to the server, the average DFS file server response and CPU times to process those requests, and the TCP/IP send and receive times that can be used to determine network overhead in DFS file server response times.

| **SMBPRT** This report shows the number and type of calls made to the Infoprint Server and the average response time and CPU time of the Infoprint Servers per call. This is used to determine the performance of the Infoprint Server.

SMBCOMM

This report shows the TCP/IP asynchronous socket calls made by the file server for SMB client communications and the amount of queuing for incoming SMB requests.

VM

This report shows the storage usage and performance of the DFS file data cache that is stored in virtual memory. It is used to reduce disk I/O rates and physical file system calls.

ADAPTERS

This report shows the storage usage and performance of the DFS file server file status, directory contents, file name lookup, and file security cache used by both the HFS and RFS physical file systems. It also shows the number of calls made by the DFS protocols (DCE or SMB) to the HFS and RFS physical file systems, the number of calls to the DFSMS HFS file system, and the average response time of each call.

RFS

This report shows the access method counts and response times for the RFS physical file system.

LFS

This report shows the number of calls made by the DFS protocols to the local file system (Episode) along with the performance of the Episode metadata cache. It also shows I/O rates to each disk, the amount of data transferred, the number of I/O waits, and average I/O wait time.

TKM

This report shows the token manager statistics for the DFS file server. It includes the size of the token manager cache, the token manager request rates, the number of token revoke RPC calls to clients, and the average response time for a token revoke RPC call.

SMBTKC

This report shows the storage usage and performance of the SMB token cache used to obtain tokens for remote SMB clients. This ensures that SMB and DCE client caches remain synchronous. It also shows SMB opportunistic lock callbacks and average callback response time.

LOCKING

This report shows the DFS lock facility statistics. It includes the number of lock waits, the average lock wait time, and the time threads sleep waiting for certain specific events.

STORAGE

This report shows the total storage used by the DFS file server. It includes storage obtained from the LE heap, storage obtained from MVS subpools, and TCB owned storage.

ALL

This option shows all of the reports described above.

Reports Available for the DFS Client

The following reports are for the DFS Client.

AFS4INT

This report shows the number and type of remote DCE/DFS RCP requests made by the client, the average DFS file server response and CPU times to process those requests, the amount of bytes transferred to or from the client, and the RPC counts and times for remote servers and servers on the local machine.

CMSERV

This report shows the external call rate, DFS client response times for each type of call, and client service thread queueing performance.

CMCACHE

This report shows the storage usage and performance of the DFS/SMB client file status, file data, directory data, name lookup caches, and DFS client background I/O statistics.

VM

This report shows the storage usage and performance of the DFS server file data cache that is stored in the server's virtual memory. It is used to reduce disk I/O rates and physical file system calls. This report is only meaningful if the OS/390 client uses disk caching.

| **LFS** This report shows the number of calls made by the DFS protocols (DCE or SMB) to the local file system (Episode) along with the performance of the Episode metadata cache. It also shows I/O rates to each disk, the amount of data transferred, and the number of I/O waits, and average I/O wait time. This report is only meaningful if the OS/390 client uses disk caching.

| **LOCKING** This report shows the DFS lock facility statistics. It includes the number of lock waits, the average lock wait time, and the time threads sleep waiting for certain specific events.

| **STORAGE** This report shows the total storage used by the DFS/SMB client. It includes storage obtained from the LE heap, storage obtained from MVS subpools, and TCB owned storage.

| **ALL** This option shows all of the reports described above. Following is an example of the command syntax using the ALL option:

| F DFS,SEND DFSKERN,QUERY,ALL

| RESET Command

| The RESET command clears the statistics monitored by the DFS/SMB file server or client. The QUERY command always displays the statistics since client or server startup or since the last RESET. This allows an administrator to query performance just for a given time period of day, such as during peak usage times. The syntax for the server is:

| F DFS,SEND DFSKERN,RESET,<REPORT>

| The syntax for the client is:

| F DFSCM,RESET,<REPORT>

| where <REPORT> is the name of a report as described above. The following example clears the statistics kept for the AFS4INT report.

| F DFSCM,RESET,AFS4INT

| Data Normalization

| When analyzing performance information, you can normalize a performance metric with respect to the external request rate. For example, the SMBINT report records the number of SMBs received by the file server and the average SMB response time. Additionally, DFS/SMB provides many other fields that show count and average response times. When analyzing data, it is often easier to normalize a statistic per the average request response time. Following is an example.

| From the SMBINT report the following was shown from a QUERY command:

| Total SMB calls 5649396
| Average DFS Response Time per SMB 2.158

| Additionally, from the same QUERY command, the LOCK report showed the following:

| Total waits for locks: 1366750
| Average lock wait time: 0.535 (msecs)

| Therefore, to get average DFS/SMB lock wait time per SMB you could perform the following calculation:

| Avg. Lock Waits per SMB = $\frac{\text{Total waits for locks}}{\text{Total SMB calls}}$ = 0.242

| Avg. DFS lock wait time per SMB = Avg. Lock Waits per SMB X Average lock wait time
| = 0.522

- | DFS lock wait time is approximately 25% of the overall response time in the previous example.
- | This same technique can be used to determine I/O waits per SMB or DCE RPC call for the LFS file system, or HFS call time per SMB or DCE/RPC by looking at data from the LFS and ADAPTERS report, respectively.

| DFS Server Tuning

- | This section discusses Distributed File Service server tuning.

| Workloads

- | DFS file server performance is controlled by many factors. Some factors are external, such as the requests the clients present to the server. The client workload determines much of the file server performance characteristics. For example, are there many large file reads or writes?, do the clients request many metadata operations such as file renames, deletions, hard links, and so forth. Other factors include the file read to write ratio (is the workload write intensive?). The type of the client (DCE or SMB) also determines the workload seen by the server, along with the location of the client (is the client on same machine as the server or not?).

| Hardware

- | Hardware greatly affects file server performance. Bottlenecks in disk I/O could greatly increase server response times. An I/O bottleneck could exist because of improper balancing of I/O amongst the server's disks. It could also be a system's disks, channels, or control units cannot handle the disk I/O rates presented by the various physical file systems that DFS/SMB supports. DFS file server cache sizes may also affect disk I/O rates or result in increased or decreased I/O waits (DFS/SMB uses asynchronous I/O whenever possible).

- | Another possibility for contention is the network. Network transfer rates can represent a large portion of the end user response time. It could be, greater than the client and server portion of the end user response time. Network performance is especially important if large files are constantly read from or written to the file server.

| SMB Workloads

- | SMB (Windows) clients often cache less data than DCE clients. For example, they do not cache file status or directory contents nearly as often as DCE clients do, therefore SMB workloads characteristically send much more file lookups and directory reads than DCE clients. While SMB workloads have much higher message rates per client user task or command, DFS/SMB caches data to turn around SMB requests as fast as possible.

- | **Tuning Options:** The following tuning options are available that control the type of SMBs that flow from clients to servers. These are all environment variables that would be specified in the dfskern environment variable file.

| _IOE_SMB_RAW

- | This option controls whether the client can use the raw mode SMBs that send data in large amounts to the server. Enabling raw mode improves large file read and write performance but the administrator can disable this if desired. Enabling raw mode allows the client or server to send or receive up to 64K of data in one packet. Additionally, the transmission is optimized. For some networks disabling raw mode improved performance and therefore DFS/SMB allows the administrator to disable raw mode transmission. The possible values are:

| **ON** This is the default. Clients send raw mode SMBs to the server.

| **OFF** This disables raw mode and no client sends raw mode SMBs to the server.

| **_IOE_SMB_MAXXMT**

| This option controls how much data can be sent in one SMB. This is specified as a whole integer and the default is 65535. This variable controls how much data a client or the server sends in one SMB and determines how large a single file read or write request can be. Any time a client desires to send more data in one SMB it uses raw mode transmission for the read or write. The maximum possible value is 65535.

| **Note:** A good test to determine the optimal settings for your network would be to make a simple test to copy large files to or from an idle server and track the response time. The test could be repeated with different options enabled at the server.

| **_IOE_SMB_OPLOCKS**

| This option controls whether clients are allowed to request an opportunistic lock when a file is opened for read or write. Opportunistic locks allow clients to cache the file data in certain cases, cache byte range lock requests, and perform read-ahead and write-behind optimizations. It is recommended that you run with oplocks enabled, which is the default. The possible values are:

| **ON** This is the default.

| **OFF** Sets oplocks off. Clients will not cache files and will not perform read-ahead or write-behind.

| **_IOE_SMB_PROTOCOL_LEVEL**

| This option determines the level of protocol negotiated with remote clients. It is the dialect that the client and server use to communicate with each other. The default is the NT protocol which is the highest level available. The following values are allowed:

| **NT** This is the highest protocol level and the default.

| **LANMAN** This is the next highest level and can be set by the administrator.

| **_IOE_SMB_NT_SMBS**

| This option determines the NT capabilities for the NT protocol. The possible values are:

| **ON** Turns on the NT capabilities. This is the default.

| **OFF** Turns off the NT capabilities. For some workloads, performance is better when NT capabilities are turned off.

| **Diagnostics:** The following sample output is from the SMBINT report. It can be used by an administrator to determine the amount and type of file related calls made to the DFS file server by SMB clients.

SMB Call	Count	Average DFS Time (msecs) Response Time	CPU Time
smb_mkdir	20	144.581	31.926
smb_rmdir	20	96.246	19.172
smb_close	1779	3.310	2.195
smb_unlink	207	151.634	35.641
smb_mv	15	133.500	14.208
smb_getattr	1872	11.284	0.778
smb_setattr	469	8.206	0.872
smb_write	8644	343.186	12.129
smb_chkpath	45	6.176	0.516
smb_readBraw	154	6.149	4.993
smb_writeBraw	21	274.119	63.503
smb_setattrE	404	46.993	0.544
smb_getattrE	445	0.556	0.445
smb_lockingX	37	0.305	0.245
smb_openX	2008	30.339	6.214
smb_readX	20883	4.295	1.850
smb_trans2	2219	10.424	0.636
T2: find first	941	12.764	0.366
T2: find next	20	0.173	0.151
T2: qfsinfo	344	0.030	0.027
T2: qpathinfo	540	5.778	0.183
smb_findclose	20	3.793	3.670
smb_negprot	0	0.000	0.000
smb_sesssetupX	0	0.000	0.000
Total SMB calls	39262		
Average DFS Response Time per SMB		82.411	
Average DFS CPU Time per SMB		4.442	
TCP/IP Session read count:	47751	Avg Time:	0.136 (msecs)
TCP/IP Session write count:	39301	Avg Time:	0.252 (msecs)
Average TCP/IP call time per SMB		0.418	

This report shows the workload presented to the server from the SMB clients. It shows the type and performance of each SMB and an overall average. Additionally, the session read and write counts show performance of the TCP/IP calls made to read or write data on the client socket. This time is included in the overall SMB response time but is not controllable by the DFS file server.

The administrator could use this information in the report to determine the affect of the setting of a tuning option or a hardware change on server SMB performance. Excessive TCP/IP send/receive times relative to the average SMB response time may indicate a network bottleneck. If the TCP/IP send plus receive time is greater than 50% of the overall SMB average response time, a network bottleneck is probably present.

Note: Only the SMB types that were received by the server since the last statistics reset are shown.

The following sample output is from the SMBPRT report. The SMBPRT report shows the calls made to the Infoprint Server on behalf of print requests received by the DFS/SMB server, the average response time, and CPU time of each call. This information can be used to determine the performance of the Infoprint Server and the portion of the DFS/SMB response time represented by Infoprint Server calls.

Infoprint Server API	Average Print Time (msecs)			CPU Time
	Count	Failed	Response Time	
AbortPrintFile	0	0	0.000	0.000
BeginEnumJobs	30	0	1.395	0.496
BeginEnumPrinters	0	0	0.000	0.000
CancelJob	1	0	23.560	2.879
ClosePrintFile	3	0	31.473	2.806
CreatePrintFile	3	0	15.222	2.475
EndEnumJobs	30	0	0.042	0.041
EndEnumPrinters	0	0	0.000	0.000
EnumJobs	60	0	0.574	0.314
EnumPrinters	0	0	0.000	0.000
GetJobInfo	8	0	1.422	0.798
GetPrinterInfo	110	3	1.886	0.530
HoldJob	0	0	0.000	0.000
InitAPI	2	1	558.078	37.522
ReleaseJob	0	0	0.000	0.000
SetTerminationHandler	1	0	0.016	0.016
TermAPI	0	1	0.000	0.000
WritePrintFile	55	0	0.271	0.253
Total Infoprint Server calls	303			
Average Response Time per call	5.251			
Average CPU Time per call	0.684			

The calls to CreatePrintFile, ClosePrintFile, and WritePrintFile occur when files are being printed by PC clients on Infoprint Server managed printers. The response time listed is the average time the print request actually spent in the Infoprint server, processing the request, or waiting for a request to complete.

Having a PC client printer queue window open results in a high volume of GetPrinterInfo and xxxEnumJobs requests to be made to the Infoprint Server. A high value for this number can represent the fact that these windows are left open for extended periods of time. Doing this can lead to unnecessary network traffic, and network contention.

Each of these remote printer queue requests uses a thread represented by the dfskern environment variable **_IOE_SMB_MAIN_POOL** to execute. The system administrator should use this information to ensure that enough threads are present to process these requests along with the file related requests from PC clients. If this number is too low, it can appear that the response time of the File portion of the SMB server is slow, but actually the file related requests are being queued instead of immediately handled. See the SMBINT and SMBCOMM reports for more information on how the SMB requests are actually being received and handled by the SMB server.

SMB Service Threads

The DFS server has two thread pools to service incoming SMB requests. If all threads are busy then incoming requests are queued until a service thread becomes available. There is a primary pool that is used to service most SMB requests and a secondary pool to handle callbacks from clients and is used to ensure that client callbacks have available threads to process them. An administrator tunes the size of the primary pool.

Tuning Options: The following environment variables control the size of the DFS file server main and secondary thread pools for SMB. The storage requirements of an SMB processing thread is shown in the SMBCOMM report.

_IOE_SMB_MAIN_POOL

This is a whole number indicating the number of threads to assign to the DFS file server main thread pool. The default is 14. This is shown as pool number 0 in the SMBCOMM report.

_IOE_SMB_CALLBACK_POOL

This is a whole number indicating the number of threads to assign to the DFS file server secondary thread pool. The default is 2. This is shown as pool number 1 in the SMBCOMM report.

The default may not be good for all your workloads. If you have a large number of active clients you want to increase the size of the main thread pool. The following section provides a diagnostic aid that can help an administrator determine the optimal thread pool size.

Diagnostics: The following sample output is from the SMBCOMM report. This report indicates the incoming requests to the DFS/SMB file server and the amount of queuing that occurs.

```
SMB Asynchronous I/O Statistics
-----
Service Thread Stack Size=76K  Number of Pools=2
Pool:  0  Threads:  40  Pool:  1  Threads:  2

SRB Accepts :      0      SRB Requests :  387008
SRB Queued  :      5
Schedules   :      0      Accepts      :      0
Reads       :      0      Readvs     :      0
Recvs      :  386875      Recvfroms :    136
Writes     :      0      Writevs    :      0
Sends      :      0      Sendtos    :      0
Cancels    :      0      Cancelsockets :      0
```

The number of threads in the pool along with the size of each thread's stack is shown to allow for an approximate determination of the amount of storage required if a thread pool is adjusted in size. The SRB Accepts field indicates the number of new clients connecting since the statistics were reset (or since startup). The SRB Requests field is a raw count of client requests received by the server. The SRB Queued field indicates the number of requests that needed to be queued due to lack of service threads. The administrator should ensure that the percentage of requests queued is not more than 5% of the total requests since this represents increased response time to process SMB requests.

DCE Workloads

The most fundamental report available for determining the server DCE workload is the AFS4INT report. There are no server tuning options available that affects the rate or type of requests made to the server. However, larger client caches would reduce RPC rates to the server. The AFS4INT report shows response times and CPU times for each request along with the total number of file bytes sent or received by the server. These response times are *DFS only*, they do not include time spent by DCE RPC runtime sending or receiving data. DFS server response times are mainly affected by workload and disk I/O response times. For most OS/390 systems tested, overall response time ranged from approximately two to thirty milliseconds on average. The affect of any kind of server change, including a tuning adjustment or new hardware, is reflected in this overall report. Note that this command can also be used for problem determination. You could issue this command to determine if a DFS file server is receiving any RPC requests from clients or not. This is useful if there is a network or configuration problem.

The following sample output is from the AFS4INT report.

AFS4 Interface RPC Call Counts

RPC Call	Count	Average DFS Time (msecs)	
		Response Time	CPU Time
SAFS_Lookup	77055	2.093	0.593
SAFS_LookupRoot	1815	10.055	0.658
SAFS_FetchStatus	15382	0.905	0.378
SAFS_StoreStatus	21266	1.037	0.451
SAFS_FetchData	33742	20.536	0.700
SAFS_StoreData	23296	17.276	1.650
SAFS_CreateFile	7669	2.848	1.547
SAFS_RemoveFile	8234	7.044	2.020
SAFS_Rename	160	4.449	1.792
SAFS_Readdir	6396	9.796	0.667
SAFS_MakeDir	788	28.222	2.523
SAFS_RemoveDir	947	11.319	1.630
SAFS_Link	0	0.000	0.000
SAFS_Symlink	2	14.735	1.614
SAFS_FetchACL	0	0.000	0.000
SAFS_StoreACL	0	0.000	0.000
SAFS_GetToken	21023	1.541	0.296
SAFS_ReleaseTokens	4953	0.320	0.305
SAFS_SetContext	2037	23.513	0.455
SAFS_SetParams	0	0.000	0.000
SAFS_GetTime	36419	0.082	0.081
SAFS_MakeMountPoint	0	0.000	0.000
SAFS_BulkKeepAlive	0	0.000	0.000
SAFS_BulkFetchVV	0	0.000	0.000
SAFS_GetStatistics	7776	0.024	0.022
SAFS_BulkFetchStatus	0	0.000	0.000

Total RPC calls 268960
 Average DFS Response Time per RPC 5.901
 Average DFS CPU Time per RPC 0.656

Derived Fetch/Store Statistics:

Total Fetched Bytes 766433K
 Total Stored Bytes 955937K

IOEN00119I DFSKERN: SEND command - QUERY,AFS4INT completed successfully.

DCE Service Threads

There are no DFS reports available to show service thread or request queuing for DCE workloads because DCE RPC manages the service threads. It must ensure there are enough service threads to process incoming requests or the requests are delayed waiting for an available thread.

There are two thread pools used to process DCE RPC requests. The primary pool is the main pool and the secondary pool is used to prevent deadlock when clients are called back to reclaim tokens. The administrator sets the size of the main pool. The size of the thread pools are controlled by the following two startup parameters for DFSKERN. These are specified on the DFSKERN command line that is listed in the **ioepdcf** file.

-mainprocs

This is the size of the main thread pool. The default is 4.

-tokenprocs

This is the size of the secondary thread pool. The default is 2.

Each thread requires an amount of 48K storage unless overridden with LE runtime options.

If DCE RPC runtime detects that the settings are not appropriate (excessive queuing), new threads are created on a temporary basis (up to three times the setting of each pool) to handle the requests.

| However, thread creation and deletion is expensive so it is recommended that an administrator provides an ample setting for the thread pools.

| **Token Management**

| This section discusses token management.

| **Token Revokes:** Token management is important in any DFS product because it is used to guarantee client cache consistency for both SMB and DCE remote clients. DCE clients obtain tokens from the file server and return them as needed. Whenever the file server needs to reclaim a token from a client it issues a TokenRevoke RPC to reclaim it. The amount of token revokes needed and the response time of those revokes affect performance. For example, token revokes should be small relative to the total number of RPCs present in the AFS4INT report for a DCE workload.

| **Central Token Manager:** The DFS file server central token manager is a cache of tokens with a fixed maximum size. The maximum size is tailorable by the administrator. When setting this size the administrator should consider that every DCE client obtains at least one token for each fileset it references and keep that token for long periods of time. Because of this, systems with a large number of filesets may need to increase the size of the central token manager cache. When the central token manager runs out of tokens it performs garbage collection to reclaim a large amount of tokens from clients. This garbage collection is expensive and should be minimized during the garbage collection period. It is possible to ensure garbage collection does not happen at all.

| **Local User Glue:** Local user access to file systems exported by DFS (either to DCE or SMB clients) also obtain tokens to ensure remote client caches remain correct. The DFS glue code caches tokens in the UNIX System Services kernel address space. For all local user access to file systems exported by DFS, the DFS glue code gets control and ensures needed tokens are obtained before the request is processed by the file system. If the tokens are in the cache then no call to the central token manager is needed. Administrators can set the size of this cache. When a token is needed from the central token manager a task switch is required. There are a pool of threads in DFS file server address space reserved to process local user token requests and the administrator can set the size of this thread pool.

| **SMB Token Cache:** Ensuring proper file sharing semantics and client cache consistency, DFS uses an SMB token cache to obtain tokens for SMB requests from the central token manager. Windows (SMB) clients use a form of caching called opportunistic locking. These opportunistic lock requests are mapped to DFS token requests. The server caches SMB tokens from the central token manager to reduce pathlengths. Opportunistic locks may require a callback to the client. The frequency of these callbacks and their average round-trip response time are shown in the SMBTKC report. The administrator can control the size of this cache by using tuning options. Although the SMB token cache can be set larger than the central token manager cache, the SMB token cache never uses more tokens than are available in the central token manager cache. The central token manager does not allow the cache maximum to be exceeded.

| **Tuning Options:** The central token manager cache maximum size can be set by using the following tuning option:

| **`_IOE_TKM_MAX_TOKENS`**

| This is the maximum number of tokens that the central token manager can hold. The default is 10,240. The size of each token (in bytes) is shown in the TKM report.

| The following tuning options can be used to control the number of threads available to process local HFS user token requests. This is important if HFS file systems are updated concurrently from remote DFS clients and local HFS users.

| **_IOE_TKMGLUE_SERVER_THREADS**

| The number of threads available for processing requests from local HFS users to get tokens from the central token manager. The default is 5.

| **_IOE_TKCGLUE_CACHE_SIZE**

| The maximum number of files that the DFS glue code caches tokens for. The more files that are cached, there is not as much of a need to obtain tokens from the central token manager. The DFS glue provides this cache in the UNIX Systems Services kernel address space. The default value is 1000.

| The following tuning option can be used to control the size of the SMB token cache:

| **_IOE_SMB_TOKEN_FILE_MAX**

| The SMB token cache allows the administrator to set the number of files that can have tokens cached. The default is 4096. Although the maximum size is allowed to be exceeded (the central token manager really determines the maximum number of tokens and files with tokens), the SMB token cache tries to re-use its internal file structures before exceeding the maximum size. Setting the SMB token cache too small is permitted, though pathlengths are larger because of the execution of the file reuse logic trying to keep the number of files or tokens cached below the maximum. The storage for an internal file and token structure is shown in the SMBTKC report.

| **Diagnostics:** The following sample output is from the TKM report.

```
|
|           Token Management Statistics
|           -----
| Token Ceiling:      262144      Allocated Tokens:      7168
| Max Tokens:        17408      In Use:                111
| Min Tokens:         128      Free Tokens:        7057
| GC Invocations:      0      GC Failures:        0
| Tokens Obtained:    220035    Tokens Returned:    208916
| Tokens Revoked:     2237     Revokes Refused:     95
| Max Files:          18496     Files Allocated:    5120
| Min Files:           3082     Files Freed:        5107
| Files In Use:       3095     Files Recycled:    57964
| Max Filesets:      18496     Filesets in use:    43
|
| Token struct size: 80   File struct Size: 120   Fileset struct size: 168
|
|           TKN4 Interface RPC Call Counts
|
|           Average TKN RPC Response Time (msecs)
| RPC Call           Count      Response Time      CPU Time
| -----
| TKN_TokenRevoke     2094        16.221          1.045
| TKN_AsyncGrant        0          0.000          0.000
| TKN_Probe            0          0.000          0.000
| TKN_InitTokenState   211        32.989          1.100
|
| Total TKN RPC callbacks      2305
| Average Response Time per TKN RPC      17.756
| Average CPU Time per TKN RPC      1.050
|
| IOEN00119I DFSKERN: SEND command - QUERY,TKM completed successfully.
```

| The TKM report shows the maximum number of tokens allowed in the cache (Max Tokens) which is the administrator setting. The Allocated Tokens field indicates how many tokens are allocated and In Use indicates how many are in use by clients at the given time. Similarly, the Max Files, Allocated Files, and Files in Use indicate the file maximum, the number currently allocated in memory, and the number in use to hold client tokens. The most important fields in this report are GC Invocations and GC Failures. These numbers should be as close to 0 as possible. (0 is optimal). Garbage collection should not be invoked more than a few times per day or peek period. It is expensive, especially for DCE workloads, because it requires large RPC requests made to many clients at once to reclaim tokens. The bottom of the report shows RPC callbacks to reclaim tokens (TKN_TokenRevoke). These counts should be small,

relative to the number of DCE client RPC requests (found in AFS4INT report), not more than a few percent of the total number of DCE RPC requests.

The following sample output is from the SMBTKC report.

```

SMB Token Cache Statistics
-----

```

	Allocated	In-use	
Hosts	3	3	
Files	190	68	(Max: 4096, Struct size: 152)
Tokens	189	67	(Struct size: 72)
Open Holds	21	0	(Oplocked: 0)
Lock Holds	21	0	
Data Holds	55	1	

	Count		Count
Token Obtains:	239	Token Returns:	212
Fast Revokes:	217	Slow Revokes:	0
File Opens:	1779	Compatibility:	0
Oplocks:	1779	Exclusive Grants:	0
Batch Grants:	0	Pre-CIFS Grants:	1509
Level2 Grants:	0	Failed Opens:	0
File Locks:	0	Permitted Checks:	29702
Tkset Adds:	43277	Longterm Holds:	40


```

Total Callbacks to clients:      72      Response Waits:      72
Average Callback Response Time: 20.335 (msecs)

```

This SMBTKC report shows a number of statistics related to SMB token cache performance. The important fields are the Total Callbacks To Clients and Average Callback Response Time. The counts should not be much more than a few percent of the total number of SMBs that are shown on the SMBINT report. The Files allocated and in-use count should be examined. If these values are greater than the maximum then the SMB token cache is low on tokens. Server response times are increased slightly because of increased pathlength. If there are no DCE clients then the SMB token cache number of files can be set to the central token manager maximum files setting. With DCE clients it is more difficult but the central token manager is always the limiting factor.

Virtual Memory File Cache

The DFS/SMB file server has a cache that is used to contain file data to reduce disk I/Os and perform read-ahead and write-behind of file data. This cache is hereafter referred to as the virtual memory (VM) cache. The administrator can tailor the maximum number of files to cache and how much storage is used to contain the cache. This disk cache is like most disk caches. It has least recently used (LRU) algorithms used to keep to most recently used file data in memory.

This cache is used for all DFS/SMB server file systems: HFS, RFS, and LFS. Because of the nature of RFS, the cache is mainly used in a write-through mode and less write optimizations are made for RFS. The proper tuning of this caching system is essential, especially for SMB client workloads and for all LFS workloads, regardless of client type. Following are some guidelines for tuning this cache for the different types of workloads and file systems.

SMB Workloads: Windows (SMB) clients caches directory and file status information in some cases. It performs limited file data caching but the data is not cached as efficiently as DCE/DFS clients. Therefore, the VM cache is important to these workloads. As with all data caching systems, the base measure of performance is the file read hit ratio. This is the percentage of times that a read request is satisfied from the cache rather than requiring a disk I/O. This ratio for SMB workloads should be at least 80-90%. The VM cache provides diagnostics and tuning options that allows the administrator to determine the performance of the cache and alter the cache performance. The HFS file system also caches data in

| the address space, but the VM cache is used for HFS to reduce calls to HFS and hence reduce
| pathlength. A production DFS/SMB file server should be run with a larger VM cache for SMB workloads
| than the current 1M default.

| **DCE/DFS Workloads:** DCE/DFS clients aggressively cache data. You will see a very low read hit
| ratio for these types of workloads. However, that does not mean the cache is not important. Another
| function of the cache is to provide the ability to perform write-behind to disk to allow for faster file write
| performance. This requires cache buffers to store the data while it is pending a disk I/O. If the cache is
| too small relative to the amount of data being stored at a given time, then as cache buffers are recycled,
| synchronous I/O waits will need to be performed to free buffers for a new file write request. The VM
| cache diagnostics provides the administrator an indication if this is occurring. The VM cache needs to be
| larger if there is much concurrent activity from the local DFS/SMB client because that bypasses DCE/RPC
| and hits the server much faster with file write requests than remote clients. This write-behind optimization
| is especially important for LFS workloads.

| **Tuning Options:** The basic VM cache tuning options control how much storage is used for the cache
| and the maximum number of files allowed to be cached. The number of files can be set large because
| each in-memory structure that represents a file in the VM cache is not too large. The size of each file
| structure is shown in the VM report.

| **_IOE_VM_MAX_FILES**

| This is the maximum number of files to cache. The default is 4096.

| **_IOE_VM_CACHE_SIZE**

| This is the maximum size in bytes of the VM cache. The user can prefix the size with 'K' or 'M'
| to indicate kilobytes or megabytes, respectively. The default size is 1M. The following
| example sets the cache size to 256 megabytes.

| `_IOE_VM_CACHE_SIZE=256M`

| **Diagnostics:** The following sample output is from the VM report. This report shows the performance
| of the virtual memory caching system since last reset or since server startup (whichever is most recent).

```

|                                     Virtual Memory Caching System Statistics
|                                     -----
|
| External Requests:
| -----
| Reads          37871    Fsyncs          251    Opens          119818
| Writes         35362    Truncates       12723    Unmaps           3805
| Asy Reads      2882     Getattrs      283760    Schedules        7610
|
| File System Reads:
| -----
| Reads Faulted    28075    (Fault Ratio 74.13%)
| Writes Faulted    0      (Fault Ratio 0.00%)
| Read Waits       1829    (Wait Ratio 4.83%)
| Total Reads      30766
|
| File System Writes:
| -----
| Scheduled Writes    35362    Sync Waits          223
| Error Writes        0      Error Waits           0
| Page Reclaim Writes 0      Reclaim Waits        168
| Write Waits         3      (Wait Ratio 0.01%)
|
| File Management: (File struct size=128)
| -----
| Total Files      4096    Free          3447    Pending          0
| Unreferenced     649    Referenced      0      File Waits        0
| Lookups    119818    Hits      80213 (Hit Ratio 66.95)
| Misses      39605    Reuses        0 (Reuse Ratio 0.00)
|
| Page Management (Segment Size = 64K) (Page Size = 4K)
| -----
| Total Pages      8192    Free          10      Pending          0
| Unreferenced     8182    Referenced      0      Page Waits        0
| Steal Invocations 15640    Steals From Self    0
| Files Stolen From 40164    Steals From Others  40164
| Waits for Reclaim 17

```

| The following fields from the VM report are important:

| External Requests

| These are the number of requests presented to the VM cache system. The most significant fields are Reads and Writes which represent read and write requests made to the cache.

| Reads Faulted

| This field is the read miss ratio. It is the opposite of hit ratio. Obtaining the hit ratio you subtract the miss ratio from 100%. In this case, the hit ratio would be 100-74 = 26%.

| **Note:** The hit ratio is low, as expected, in this DCE client workload example.

| Reclaim Waits

| This is the number of times a thread had to be suspended when reclaiming cache pages for an in-progress I/O (an I/O for file data written in write-behind mode). This number should be low relative to the number of external requests. It represents a thread waiting on disk I/O. A higher number means slower general response time.

| File Waits

| This is the number of times a thread had to be suspended waiting to obtain an in-memory VM cache file structure. It means that all the VM cache files are in-use and represents an extreme shortage of VM cache file structures. The preferred value is 0.

| Total Pages

| This is the total number of pages allocated to the VM cache. A page is 4K in size. In this case there are 8192 pages, therefore, 32M.

Page Waits

This is the number of times a thread had to wait for a free page in the cache to become available. It represents cases where there is an extreme shortage of pages in the cache. This number should be low, preferably 0, and should not be more than 1% of the total external requests.

Waits for Reclaim

The VM cache allows only one thread at a time to reclaim pages from the cache for use for a new file read or write request. If another thread requires a page and it finds another thread is already running reclaim processing, it waits. The occurrence of this depends on how many concurrent threads are running at a time and how often reclaim needs to run. This count should be low (not more than a few percent of the total number of external requests).

In the above VM report, performance of the VM cache for a DCE client workload is very good based on the indicators. The administrator can use these indicators and set the cache size to obtain an optimal size of the cache for their environment.

If File Waits is high then the number of files should be increased by setting the **_IOE_VM_MAX_FILES** environment variable to a higher number. If the number of Waits For Reclaim or Page Waits is high then the maximum storage should be increased by setting the **_IOE_VM_CACHE_SIZE** environment variable to a higher number. Refer to Appendix C, "Environment Variables in DFS" on page 153 for more information on these environment variables.

HFS/RFS Caches

The HFS/RFS file systems share a cache used to store file status information such as owner ID, file change time, file permissions, directory contents, and directory name to vnode mappings. A vnode in DFS/SMB is much like a UNIX inode. It is the data structure used to represent a file system object such as a directory or file. The RFS file system uses MVS access methods to manipulate datasets while HFS is accessed through OS/390 UNIX System Services. This cache hides HFS/RFS implementations that do not use the same interfaces as the rest of DFS/SMB and provides pathlength reduction because it reduces calls substantially to MVS access methods or HFS. The performance of this cache is essential for RFS performance. Formal tests have shown it provides a throughput boost for HFS (up to 39% for some workloads) when this cache was originally added to OS/390 V1R3.0 Distributed File Service.

Status: File status is cached in an extension to the vnode. By caching this status calls are reduced to OS/390 UNIX System Services or OS/390 dataset access methods. By caching vnode handles the number of calls are reduced to OS/390 UNIX System Services to obtain and return handles. The size of this cache is controlled by the administrator.

Permissions: The HFS/RFS cache stores user permissions to file system objects. This is done by querying the permissions from SAF or OS/390 UNIX System Services. By saving the permissions, calls to OS/390 UNIX System Services or SAF are reduced.

Directory Contents: The directory cache stores directory contents in DFS/SMB format. Use of this cache reduces lookup and directory read calls made to OS/390 dataset access methods or HFS.

Name Lookup Cache: The name lookup cache saves directory entry name to associated vnode mappings reducing calls to obtain vnode handles.

Tuning Options: An important tuning option for the shared HFS/RFS cache is the size of the directory cache. A larger cache allows directories to be cached and results in less calls to the associated physical file system routines. However, for large production systems using HFS or RFS, a large vnode cache may be desirable.

_IOE_VNODE_CACHE_SIZE

This is the number of vnodes provided in the cache. The default is 4096. More vnodes are provided, if needed, resulting in less calls to the HFS or RFS physical file system to read file status, file permissions, and obtaining vnode handles. The size of a vnode structure is shown in the ADAPTER report.

_IOE_DIRECTORY_CACHE_SIZE

This is the number of 512 byte blocks that can be used to store directory contents and name to vnode mappings. The default is 2048 blocks, which is one megabyte of storage.

Diagnostics: The following sample output is from the ADAPTER report. The ADAPTER command can be used to determine the performance of the shared HFS/RFS cache and the performance of HFS.

```

Adapter Caching Statistics
-----
Vnode Cache Size = 6144 vnodes, structure size = 488 bytes
Vnode lookups = 1368121, hits = 1326732, ratio = 97%
Vnode invalidations = 41342
Directory Cache Size = 2048 (512 byte) blocks, Free = 729
Directory buffer refreshes = 0
Directory buffer reads = 5494594, hits = 5494594, ratio=100%
Get attributes calls = 2451296, hits = 2394971, ratio=98%
Name cache lookups = 5300598, hits = 5054509, avoided=246089, ratio=100%
Avoided set attributes calls = 1462
Access checks = 8145852, hits = 8063074, ratio=99%
Access cache invalidates = 82731

HFS Adapter Vnode Op Counts

Vnode Op          Count  Vnode Op          Count
-----
xoefs_hold         1158433  xoefs_readdir     143600
xoefs_rele         7651388  xoefs_create       41387
xoefs_inactive      0         xoefs_remove       41343
xoefs_getattr      2043702  xoefs_rename       8684
xoefs_setattr       316196   xoefs_mkdir        0
xoefs_access        519908   xoefs_rmdir        0
xoefs_lookup        5313507   xoefs_link         0
xoefs_getvolume     0         xoefs_symlink      0
xoefs_getlength     0         xoefs_readlink     0
xoefs_afsfid        8149223   xoefs_rdwr         0
xoefs_fid           0         xoefs_fsync        14362
xoefs_vmread        150510   xoefs_translate    0
xoefs_vmwrite       101034

Total HFS Vnode Ops  25653277

```

Average HFS Time (msecs)			
VFS Call	Count	Response Time	CPU Time
v_access	82778	1.443	0.021
v_lookup	0	0.000	0.000
v_get	0	0.000	0.000
v_getattr	56325	18.862	0.024
v_setattr	356120	7.694	0.065
v_create	41387	41.396	0.196
v_link	0	0.000	0.000
v_mkdir	0	0.000	0.000
v_rdwrr	265906	49.819	0.171
v_readdir	0	0.000	0.000
v_readlink	0	0.000	0.000
v_rel	41342	3.549	0.041
v_remove	41343	26.775	0.154
v_rename	8684	28.960	0.349
v_rmdir	0	0.000	0.000
v_rpn	0	0.000	0.000
v_export	0	0.000	0.000
v_symlink	0	0.000	0.000
v_fstatfs	10140	0.059	0.041

Total VFS calls 904025
 Average HFS Response Time per Call 22.553
 Average HFS CPU Time per Call 0.101

RFS Adapter Vnode Op Counts			
Vnode Op	Count	Vnode Op	Count
xrda_hold	0	xrda_readdir	40
xrda_rele	0	xrda_create	219
xrda_inactive	0	xrda_remove	197
xrda_getattr	17692	xrda_rename	10
xrda_setattr	1479	xrda_mkdir	20
xrda_access	21456	xrda_rmdir	20
xrda_lookup	7555	xrda_link	0
xrda_getvolume	0	xrda_symlink	0
xrda_getlength	0	xrda_readlink	0
xrda_afsfid	50687	xrda_rdwrr	0
xrda_fid	0	xrda_fsync	0
xrda_vmread	1591	xrda_translate	10059
xrda_vmwrr	8468		

Total RFS Vnode Ops 119493

RFS Adapter VFS Op Counts			
VFS Op	Count	VFS Op	Count
r_get	0	r_readdir	20
r_rel	222	r_create	219
r_getattr	57	r_remove	197
r_setattr	1259	r_rename	10
r_access	478	r_mkdir	20
r_lookup	5	r_rmdir	20
r_export	0	r_link	0
r_fstatfs	2	r_symlink	0
r_rpn	0	r_readlink	0
		r_rdwrr	10059

Total RFS Operations 12568

The following sections provide explanations of the ADAPTER report.

Adapter Caching Statistics: This is an important section of the report because it details performance of the caches.

Vnode Cache Size

This is the size of the vnode cache. It determines how many files or directories status and file permissions can be cached in the HFS/RFS cache. It also displays the size of an HFS/RFS vnode.

Vnode lookups

This is the number of searches for vnodes based on the inode (file number) and the hit ratio. The higher the hit ratio, the better.

Directory Cache Size

This is the size of the directory cache.

Directory buffer reads

This is the number of directory buffer read requests and the percentage of time the directory buffer was in storage.

Get attributes calls

This is the number of calls made to obtain file status information and the percentage of time valid attributes were found in the cache.

Name cache lookups

This is the number of calls made to find the vnode associated with a given file name. The percentage of times the vnode was found without a call to the HFS or RFS physical file system is also shown.

Access checks

This is the number of permission checks made and the percentage of times the permissions for a user were found cached.

The hit ratios of the various caches should be approximately 80% or more. If the hit ratios are low then the size of the vnode or directory caches could be increased.

HFS Adapter Vnode Op Counts: This section of the report shows the number of each type of call made by the DCE or SMB protocol servers to the HFS adapter. Many of these calls do not result in an HFS call due to the presence of the shared cache.

Average HFS Time: This section shows the average response time and CPU time in milliseconds for each HFS physical file system call. The administrator determines the portion of the overall DFS/SMB response time that is represented by HFS calls, if desired. Changes to disk configurations would affect the performance of these response times because disk I/O wait time would be the predominant factor in these response times. Large HFS call times relative to the average DFS server response time could indicate DASD I/O bottlenecks.

RFS Adapter Vnode Op Counts: This is the number of calls made to the the DCE or SMB protocol servers to the RFS adapter. Many of these calls do not result in an RFS file system call due to the presence of the shared cache.

RFS Adapter VFS Op Counts: This is the number of calls made to the DFS/SMB RFS physical file system. Response times are not given for these calls because the DFS/SMB product provides the support for RFS. The RFS report shows the OS/390 access method response times.

HFS

In addition to the shared HFS/RFS cache, the bulk of the performance and tuning issues are a function of the HFS file system. The appropriate HFS publications should be used to tune the HFS file system.

LFS (Episode)

The performance of Episode, like any file system, is dependent on I/O response times. Episode attempts to perform as much asynchronous I/O as possible to speed up response times. It uses logging techniques to provide fast and reliable recovery and reduces file system metadata I/O substantially.

Aggregate Log File: Every Episode aggregate has a log file that is fixed in size. This size is set by the NEWAGGR program. The default is 1% of the total disk size. This may be too small for small disks with many concurrent transactions. The size of the log file can only be set at aggregate format time by NEWAGGR. If the log file size needs to be changed the aggregate has to be reformatted.

Aggregate Block Size: In order to obtain optimal aggregate block size performance, a default block size of 8K is recommended. It should be used when formatting the aggregate with the NEWAGGR program.

Metadata Cache: The Episode file system has a cache for file system metadata, which includes directory contents and the data of small files (smaller than the aggregate block size). The setting of this cache size is important to Episode performance because Episode references the file system metadata frequently. Synchronous reads of metadata increases I/O rates to disk and server response times.

Tuning Options The setting of the VM cache size is important for LFS performance. The administrator should set the VM cache tuning options in conjunction with the LFS tuning options.

_IOE_EPI_CACHE_SIZE

This sets the size, in bytes, of the LFS buffer cache used to store file system metadata and small file contents. You can append a 'K' or 'M' to the size to indicate kilobytes or megabytes, respectively. The default is 1M.

_IOE_LFS_IO_SEEK

This allows the administrator to enable a seek scheduling algorithm for LFS aggregates where I/O is optimized by disk location. This changes the order from a first-come first-serve basis to one based on traditional elevator algorithms for non-priority I/O (I/O that is not being waited on by a thread). The possible values are ON or OFF. The default is ON. The following example sets the size of the LFS buffer cached to 10 megabytes.

```
_IOE_EPI_CACHE_SIZE=10M
```

Diagnostics: The following sample output is from the LFS report. This report shows the performance of the LFS file system, the I/O rates to disk, and the average I/O wait time.

```

                                LFS Vnode Op Counts
Vnode Op          Count  Vnode Op          Count
-----
efs_hold           0     efs_readdir      486909
efs_rele           0     efs_create       140363
efs_inactive       0     efs_remove       140580
efs_getattr        5435375  efs_rename       29422
efs_setattr        594838  efs_mkdir        0
efs_access         1122105  efs_rmdir        0
efs_lookup         5357251  efs_link         0
efs_getvolume      0       efs_symlink      0
efs_getlength      0       efs_readlink     0
efs_afsfid         10295933  efs_rdwrr       0
efs_fid            0       efs_fsync        48720
efs_vmread         46430  efs_waitIO       210106
efs_vmwrite        239265  efs_cancelIO     71672

Total LFS Vnode Ops    24218969
```

```

|                                     LFS Caching Statistics
|
| Buffers      (K bytes)  Requests      Hits      Ratio  Updates
| -----
|          4096      32768      8975442      8419314  93.8%   9552884
|
|                                     LFS Directory Cache Statistics
|
| Dir Blocks (K bytes)  Requests      Hits      Ratio  Deletes
| -----
|          128      1024      10138937      10121974  99.8%      0
|
|                                     I/O Summary By Type
| -----
|
| Count      Waits      Cancels      Merges      Type
| -----
|          1474      1598      0      0  File System Metadata
|         136416      51562      0      2931  Log File
|         361717      105865      310      4  User File Data
|
|                                     I/O Summary By Circumstance
| -----
|
| Count      Waits      Cancels      Merges      Circumstance
| -----
|          47502      47502      0      0  LFS cache read
|           213      213      0      0  VM cache direct read
|            0      0      0      0  Log file read
|            0      0      0      0  LFS cache async delete write
|            0      0      0      0  LFS cache async write
|         104355      10307      0      1  LFS cache lazy write
|            0      0      0      0  LFS cache sync delete write
|            0      0      0      0  LFS cache sync write
|        209648      47843      310      3  VM cache direct write
|            0      0      0      0  LFS cache file sync write
|           725      723      0      0  LFS sync daemon write
|            0      0      0      0  LFS aggregate unexport write
|            0      0      0      0  LFS buffer block reclaim write
|            0      0      0      0  LFS buffer allocation write
|            0      0      0      0  LFS fileset quiesce write
|           749      875      0      0  LFS buffer log file full write
|        136415      51562      0      2931  Log file write
|            0      0      0      0  LFS buffer shutdown write
|
|                                     LFS I/O by Currently Attached Aggregate
|
| Minor
| Device MV  Mode      Reads      K bytes      Writes      K bytes  Dataset Name
| -----
|      1  N  R/W      0      0      0      0  DFS.EPISODE.AGGR001.LDS00001
|     401  N  R/W      5854      46888      54963      1334992  IOEDFSKN.LFS.SET41
|     402  N  R/W      3958      31944      37204      887544  IOEDFSKN.LFS.SET42
|     301  N  R/W      5745      45960      53338      1304256  IOEDFSKN.LFS.SET31
|     302  N  R/W      3873      31040      36571      881288  IOEDFSKN.LFS.SET32
|     201  N  R/W      6005      48040      56573      1370048  IOEDFSKN.LFS.SET21
|     202  N  R/W      4218      33744      39778      950968  IOEDFSKN.LFS.SET22
|     100  N  R/W      0      0      0      0  IOEDFSKN.LFS.TEST3
|     101  N  R/W      5697      46080      53302      1296728  IOEDFSKN.LFS.SET11
|     102  N  R/W      3949      31984      37275      892400  IOEDFSKN.LFS.SET12
|     501  N  R/W      4206      33928      39886      955864  IOEDFSKN.LFS.SET51
|     502  N  R/W      4209      33840      39762      951864  IOEDFSKN.LFS.SET52
| -----
|      12      47714      383448      448652      10825952  *TOTALS*
|
| Total number of waits for I/O:      159025
| Average I/O wait time:      6.861 (msecs)

```

| The following sections provide explanations of the LFS report.

| **LFS Vnode Op Counts:** This is the count of each type of LFS vnode operation. It shows the request rate from the DCE protocol exporter.

| **LFS Caching Statistics:** This section of the report shows the performance of the LFS buffer cache. Because LFS allows aggregates to be blocked at different sizes (4K, 8K, etc..), the cache may grow or shrink in size depending on how many buffers for disks that are blocked larger than 8K in memory. If all aggregates are blocked at 8K then the size of the cache is fixed in size. The Buffers field indicates the number of buffers in memory and the (K bytes) column indicates total storage.

| The Hits and Ratio columns are the base indicators of performance of the cache. The Ratio should show a hit ratio of 90% or more. LFS reads from the metadata cache frequently, so even if the cache is small the hit ratio is high (100% is preferred). Because LFS hits the cache so frequently per operation, a better indication of LFS I/O performance is provided in "LFS I/O by Currently Attached Aggregate"

| **LFS Directory Cache Statistics:** The LFS directory cache is a layer above the LFS buffer cache that is used to simplify coding of directory management. This is done because each Episode aggregate could be blocked at a different size, but Episode directories are always logically blocked at 8K bytes. This cache is not tailorable by the end user and it is expected that you would not need to change the size of this cache. It is fixed in size at 1M. The contents of this cache are obtained from the LFS buffer cache, when needed.

| **I/O Summary:** The I/O summary sections show I/O by type and circumstance (why the I/O is being performed).

| **LFS I/O by Currently Attached Aggregate:** This report shows the read and write I/O rates to every LFS aggregate, the minor device number, R/W indicator, name of the first dataset for the aggregate, and read and write counts. The MV column indicates if the aggregate spans more than one dataset.

| The read and write counts show I/O rates to disk. This can be used to determine the affect of a change, such as modifying one of the file server cache sizes. It is also used to determine if there is an I/O imbalance among the disks.

| The number of I/O waits and average I/O wait time is also displayed. This should not be confused with DASD response time from reports such as RMF. LFS attempts to perform as much I/O as it can asynchronously, so LFS waits on fewer I/Os than it issues. The I/O wait count is incremented each time a thread or unit of work needs to be suspended waiting for I/O completion. It is the bottom line performance measurement of LFS. Any change to the disk I/O configuration along with cache sizes affects this number. The administrator can use this report to determine the average number of I/Os per RPC and the average I/O wait time per RPC. This should help an administrator balance I/O and determine if a disk or set of disks is a bottleneck.

| RFS

| The RFS file system maps POSIX based file I/O and maps them to OS/390 dataset access method calls. It uses OS/390 access methods to store file status information into OS/390 catalogs and retrieves the information later. The important factors regarding RFS performance is the number of access method and OS/390 dynamic allocation calls made and their corresponding response times. DFS/SMB attempts to perform I/O efficiently to and from the corresponding datasets. It tries to read/write a large number of blocks when performing I/O and attempts to overlap processing with I/O as much as possible. However, much of RFS is gated by the access method performance and disk I/O performance and is therefore the most important measure of performance. Additionally, because of the nature of OS/390 datasets, certain file access patterns such as random file access and file update patterns of Windows NT machines, RFS may be forced to perform small I/Os to the datasets or may perform synchronous I/O.

Tuning Options: The adapter cache tuning options and VM cache tuning options are important for RFS performance because the larger those caches are the smaller the number of access method calls RFS needs to make. The following RFS tuning options are important:

_IOE_RFS_WORKER_THREADS

This sets the number of worker threads required to process RFS dataset open/close requests. OS/390 has a restriction where a dataset must be closed by the same task that opened the dataset. Because the DCE or SMB protocol exporters manage the threads to process incoming requests there is no control available to RFS to ensure a close request is processed by the same thread that opened the dataset. DFS/SMB handles this restriction by making available a pool of threads used exclusively for dataset open/close processing. The default size of this pool is one thread.

bufhigh Many RFS processing options are specified in an RFS attributes file stored in HFS. An important option is the size of the cache used to store BSAM dataset blocks. The VM cache caches the file contents in POSIX byte stream format. BSAM record datasets require additional control information imbedded in the disk blocks, therefore I/O to the datasets must use disk blocks properly formatted for the dataset. RFS provides a buffer cache used to hold these blocks in raw dataset format. A sufficient size for it should be provided. This size should be set to the average number of bytes being read/written at a moment in time by the server. The default is 2M.

blksize Another RFS processing option is the *blksize* dataset creation parameter. This can be specified on the user command line and/or the RFS attributes file. This option is used for creation of BSAM files. Records should either be large, or should be grouped into large block sizes for efficient I/O performance. OS/390 allows applications to specify a block size of 0 which means OS/390 will determine optimal block size, this is the recommended setting by DFS/SMB for BSAM datasets (sequential datasets and PDS or PDS/E members).

recordsize

This sets the size of the records for VSAM datasets. OS/390 DFS will read/write up to 64K of data at a time to a VSAM dataset. However, I/O is more efficient with larger records.

space Another RFS processing option allows the administrator or user to specify the primary and secondary extent sizes of a dataset. Dataset creation and initial write takes longer if secondary extents are needed. It is best to use only a primary extent if possible, or at least make the secondary extents large enough so contents of the dataset are not scattered.

Diagnostics: The following sample output is from the RFS report.

```

|                               FILBLK Management Statistics
|                               -----
| Active NAMEBLKs = 108
| Active FILBLKs = 66
| FILBLK inactivations = 0
| FILBLK invalidations = 0
|
|                               Logical I/O Statistics
|                               -----
| Logical Cache High = 4194304 (4096K) (4M)
| Logical Cache Used = 0 (0K) (0M)
| Cache Window Per File = 16
| Logical Read Requests = 7752
| Read-for-size Requests = 0
| Read-for-offset Requests = 5
| Out-of-sequence Reads = 0
| Logical Write Requests = 12404
| Out-of-sequence Writes = 0
| Writes With Holes = 0
| Logical File Syncs = 48
| Logical File Sync Closes = 19
```

RFS Open/Close Thread Statistics

```

Number of threads: 5 (stack size = 16K)
Open requests:      809   Queued:      0 (0.0%)
Close requests:     797   Queued:     24 (3.0%)
AsyClose requests:   12   Queued:      1 (8.3%)

```

Physical I/O Statistics

```

Buffer Cache High = 2097152 (2048K) (2M)
Buffer Cache Used = 0 (0K) (0M)
Buffer Cache Steal Percent = 20%
Buffer Cache Trims = 249
Buffer Steals = 1108
Buffer Closes = 12

```

Dynamic Allocation Calls = 289

Dynamic Unallocation Calls = 267

Call Counts By Access Method

Access Method	OPENS	CLOSEs	GETs READs	PUTs WRITEs	CHECKs	WAITs	*Total*
BPAM	40	40	0	15	30	15	140
BSAM	263	263	509	704	1276	1213	4228
QSAM	20	20	20	0	0	0	60
VSAM	526	526	640	4062	903	0	6657
Total	849	849	1169	4781	2209	1228	11085

Average Time Per Call By Access Method

Access Method	OPENS	CLOSEs	GETs READs	PUTs WRITEs	CHECKs	WAITs	*Avg*
BPAM	4.901	7.276	0.000	0.163	0.015	2.708	3.790
BSAM	10.202	23.315	0.200	0.186	0.120	4.688	3.521
QSAM	11.428	2.705	0.008	0.000	0.000	0.000	4.714
VSAM	44.727	42.300	1.894	0.891	0.694	0.000	7.696
Avg	31.371	112.676	1.124	0.785	0.353	4.664	8.630

Average CPU Time Per Call By Access Method

Access Method	OPENS	CLOSEs	GETs READs	PUTs WRITEs	CHECKs	WAITs	*Avg*
BPAM	1.785	1.621	0.000	0.100	0.015	0.031	0.991
BSAM	2.097	1.705	0.112	0.104	0.055	0.028	0.292
QSAM	2.488	1.206	0.008	0.000	0.000	0.000	1.234
VSAM	13.534	9.203	1.628	0.495	0.581	0.000	2.334
Avg	9.177	26.340	0.940	0.436	0.269	0.028	1.532

The important fields in the RFS report are the access method counts, response times, and the open/close service threads statistics.

RFS Open/Close Thread Statistics: The open/close statistics show the Number of threads available to process open/close requests, the number of requests, and the number of requests that needed to be queued (because all of the service threads were busy). If a large portion of open/close requests need to be queued then response times increase because of open/close queue wait time when reading/writing files or directories. The size of each thread's program stack is shown to allow an approximation of the required storage if the number of service threads is changed.

Physical I/O Statistics: The Buffer Cache Steals and the Buffer Cache Trims indicates cases where storage needed to be stolen from other files in the cache allowing an I/O to be performed for a given file. A high steal rate relative to the total number of RFS xrdm_vmread and xrdm_vmwrite calls (from the ADAPTER report) might indicate that the RFS buffer cache could be too small and the **bufhigh** RFS tuning option might need adjustment.

| **Access Method Statistics:** RFS performance is determined by the response time of the OS/390 access methods. DFS tries to read/write to files in large (64K) amounts but certain client access patterns might prevent that. An administrator can determine from this report and the reports that detail external client requests the relative amount of access method calls per DCE/RPC or SMB and determine what portion of the response time the access methods use.

| Locking and Serialization

| The DFS file server and client provides information on DFS internal lock performance.

| **I/O Wait Time Affects Lock Wait Time:** In certain cases the file systems require a lock to be held over an I/O. I/O waits are long relative to processor speed. The slower the disk response time the longer the I/O waits. Therefore, disk I/O wait time affects not only those threads waiting on the I/O, but also other threads if the I/O waiter is holding a lock. Improving I/O response time often improves lock wait time and then overall file server performance.

| **File Op Queuing:** There are certain events that the DFS file server and client monitor. One event is file operation queuing for SMB clients. The SMB protocol exporter queues file write operations and truncation operations on an internal file handle. This is done to ensure client requests are presented to the physical file system in order. This queuing does not introduce additional serialization since the physical file system locks the file in write mode when performing a write operation. However, file operation queue waits are part of the overall server response time. Hence, the locking and serialization diagnostics provide an indication of file operation queuing. This queue time is directly related to physical file system performance related to disk response times and the efficiency of physical file system file write algorithms.

| **Diagnostics:** The following sample output is from the LOCK report.

```
|
|                               Locking Statistics
|
| Lock Obtains:      587193954   Fastpath:  577430213   (98.3%)
| Lock Releases:    658325735   Fastpath:  649381697   (98.6%)
| Lock Upgrades:         0      Fastpath:         0   (0.0%)
| Lock Downgrades:  140400      Fastpath:  140400   (100.0%)
| Non-block attempts: 533426    Success:   529193   (99.2%)
| Work units run:   71131781    Sync:     70996757   (99.8%)
|
| Untimed sleeps:    16766   Timed Sleeps:    4769   Wakeups:    443450
|
| Total waits for locks:                                8900506
| Percent of lock obtains/upgrades requiring wait: 1.5%
| Average lock wait time:          0.552 (msecs)
|
| Total monitored sleeps:                                7048
| Average monitored sleep time:      5.347 (msecs)
```

Top 15 Most Highly Contended Locks			
Thread Wait	Async Disp.	Pct.	Description
-----	-----	-----	-----
4977280	0	55.1%	LFS vnode main lock
1959305	0	21.7%	LFS main transaction lock
1272775	0	14.1%	LFS main equivalence class lock
254672	0	2.8%	SMB time of day services lock
154204	79264	2.6%	SMB AS queue lock
81346	0	0.9%	LFS log map lock
34953	33162	0.8%	SMB TKC global file LRU and free list lock
42055	0	0.5%	LFS volume handle lock
24416	0	0.3%	LFS buffer lock
18548	0	0.2%	VM cache all file lock
0	17195	0.2%	OSI Global queue of threads waiting for locks
14504	0	0.2%	LFS allocation handle lock
12283	0	0.1%	LFS vnode lock
11319	0	0.1%	SMB CS handle table lock
8014	0	0.1%	LFS file zero lock

Top 10 Most Common Thread Sleeps		
Thread Wait	Pct.	Description
-----	-----	-----
7048	100.0%	SMB CS File Operation Queue Wait
0	0.0%	VM Page Wait
0	0.0%	VM File Wait
0	0.0%	VM Page Reclaim Wait
0	0.0%	SMB TKC Tkset Revoke Wait for Tkset Holds
0	0.0%	SMB TKC Tkset Wait For Pending Revoke
0	0.0%	SMB TKC Get Token Wait
0	0.0%	SMB TKC Open-in-progress Wait
0	0.0%	SMB TKC Callback Response Wait
0	0.0%	SMB CS Oplock Break Read-Raw Wait

The important fields from the LOCK report are the Total waits for locks, Average lock wait time, Total monitored sleeps, Average monitored sleep time. These fields indicate average amount of time a thread processing a RPC or SMB request must wait on a lock or for access to a shared resource. In the previous example the SMB CS File Operation Queue Wait is not too large and therefore the disk performance and the file system performance were good. File operation queue waits occur more frequently and the average wait time is longer when a disk is overloaded and occurs infrequently when the disk is not a bottleneck.

Storage Usage

The DFS/SMB file server and client provides a command that reports DFS storage usage of the runtime heap. Storage DFS/SMB allocates directly from base OS/390 storage subpools. Therefore all DFS allocated storage is shown with the exception of storage allocated for thread stacks. There is no way for DFS/SMB to track that information, though there are LE runtime options that shows you that information.

Diagnostics: The following sample output is from the STORAGE report.

```

|                               OSI Storage Statistics
|                               -----
|
| Current allocated storage size: 49515778 (48355K) (47M)
| Number of storage allocations: 722977
| Number of storage frees: 721684
|
|                               MVS Obtained Storage Statistics
|                               -----
|
| Current allocated storage size above 16M line: 303387036 (296276K) (289M)
| Number of storage allocations above 16M line: 0
| Number of storage frees above 16M line: 0
|
| Current allocated storage size below 16M line: 1092 (1K) (0M)
| Number of storage allocations below 16M line: 0
| Number of storage frees below 16M line: 0
|
|                               TCB Owned Storage
|                               -----
| Lock storage allocations: 157
| Lock storage allocated: 11304
| Non-lock storage allocations: 804
| Non-lock storage allocated: 2317976

```

| The OSI Storage Statistics indicates current DFS/SMB program use of heap storage. MVS and TCB owned storage statistics indicate the number of bytes allocated directly from OS/390 storage subpools.

| DFS Client

| DFS client performance is tied to its cache performance and the performance of DCE RPC.

| Workload and Threading

| The DFS client has a number of caches to reduce RPC rates to file servers. DFS/SMB clients are called on the same task (TCB) as the OMVS user, and then runs in cross memory mode. The DFS/SMB client attempts to process the request fully on the user's TCB, but in some cases, when an RPC is needed, the request needs to be switched to a DFS/SMB client task to perform the RPC. There is a pool of service threads that can be controlled by the administrator that are used when a task switch is necessary to complete the processing of a request. Task switches are expensive, especially given the request rate to a production DFS/SMB client. The more often the data resides in a DFS client cache the more likely a task switch is not needed. Sufficient service threads need to be made available to process requests that could not be satisfied on the user's TCB to ensure that requests are not waiting in the service thread request queue (which results in larger response times).

| **Tuning Options:** The administrator can set the size of the DFS/SMB client service thread pool with the following environment variable:

| _IOE_CM_REQUEST_THREADS

| This is the number of service threads available to process requests that could not be fully processed on the user's task. The default number is 10.

| **Diagnostics:** The following sample output is from the CMSERV report. The CMSERV report provides details on the overall client workload, the client's average request response time, and the performance of the service thread pool.

```

|                                     DFS/MVS Client External Calls
|                                     -----
| Operation              Count      Fastpath    % Fast    Avg Time
| -----
| cm_opens                1010        992        98%        0.030
| cm_closes               1010        719        71%        12.655
| cm_reads                4061       4061       100%        0.144
| cm_writes                851        849       100%        0.412
| cm_ioctls                 5          5       100%        0.010
| cm_getattnrs            1660       1660       100%        0.019
| cm_setattnrs              0          0          0%        0.000
| cm_accesses              46          46       100%        0.015
| cm_lookups              5033       4459        89%        1.695
| cm_creates               193          0          0%       18.193
| cm_removes               212          0          0%       35.365
| cm_links                  5          0          0%        5.904
| cm_renames               51          0          0%        8.545
| cm_mkdirs                35          0          0%       53.043
| cm_rmdir                 35          0          0%       13.081
| cm_readdir               154        104        68%        1.700
| cm_symlinks              20          0          0%       13.665
| cm_readlinks             25          0          0%        0.213
| cm_fsyncs                 0          0          0%        0.000
| cm_truncs                 0          0          0%        0.000
| cm_lockctls              0          0          0%        0.000
| cm_inactives             247        247       100%        0.006
| cm_recoverys              0          0          0%        0.000
| -----
| *TOTALS*                14653       13142        90%        2.501
|
| Number of service threads: 5 (stacksize=48K)
| Requests: 1517    Queued: 4 (0.3%)

```

| This example shows that most requests were satisfied directly on the user's task with no task switch necessary (Fastpath and %Fast columns show how often a task switch is avoided).

| The Number of service threads indicates the number of defined threads and the amount of storage required for a client service thread. The Requests and Queued fields indicate how many requests from UNIX Systems Services needed to be switched to a service thread task and how often a request needed to be queued when all service threads were busy. The percentage of requests queued should be small because queue wait time directly increases end user response times.

| Client Caches

| The DFS/SMB client has a number of caches used to significantly reduce RPC rates to servers. Specifically, the client caches file status information (such as size, permissions), user ACL permissions (authority a user has to a file system object), directory name to local vnode mappings (mapping names to file system vnodes), directory contents, file contents, RPC binding handles, and DFS tokens. The sizes of most of these caches can be controlled by the administrator. The sizes of some are dependent on the settings of others.

| **File Caching:** Server file contents are cached locally in client files that are either stored in memory only (memory cache clients) or are stored in a special purpose LFS aggregate (disk cache). The size of each file will at most equal the client chunk size. This controls the granularity of each file and the amount of file data that the client transmits or receives to remote servers in a single RPC. Note that a single server file is often split and stored in multiple client files in the client disk cache. The administrator can specify the size of the cache either by simply specifying the total size of the cache (the client divides total size by chunk size to obtain number of files) or can specify the number of files that can be cached (the client multiplies the number of files by the chunk size to get total bytes).

| The DFS client saves two other files (either in memory or on the special purpose LFS aggregate). The first contains a mapping of each client file to the remote server file and offset within the remote server file.

| The other file contains cached fileset information. The mapping of local client files to remote server files is called the **CachelItems** file and this file is essentially an array of mappings. The client saves in memory (if a disk cache is used) a number of these mappings for faster access. These mappings are called a *dcache*.

| **Status Caching:** The client caches the status, user file permissions and DFS tokens for files to reduce RPC rates to servers. The number of files whose status can be cached can be set by the administrator. The size of a status cache entry is shown in the CMCACHE report.

| **Directory Caching:** The contents of directories are cached if the raw directory size is 8K or less. The administrator can control how much memory is used to cache directory contents.

| **Name Lookup Cache:** The lookup operation is the most common operation performed by applications or by the kernel. The client caches a mapping of parent directory/object name/vnode mappings that allows a lookup to be performed without the need of an RPC. Furthermore, the client caches negative lookups. That is, the client also remembers names that do not exist in directories because at times the same name is looked up repeatedly. These actions significantly reduce RPC rates. The administrator can control the size of the lookup cache. The size of each name mapping entry is shown in the CMCACHE report.

| **Tuning Options:** The following IOEDFSD startup parameters control the sizes of the various caches.

| **-blocks** Number of 1K blocks in the file cache.

| **-files** Number of files to cache. This is the number of files used for disk caching. It is not relevant for memory cache clients.

| **-stat** This sets the size of the status cache. The default is 300.

| **-memcache**
| This parameter indicates if a memory cache is to be used instead of a disk cache. The default is disk cache.

| **-dcache** The number of disk cache file descriptors stored in memory. As mentioned above, the client stores server file contents in local files on a special purpose LFS aggregate for disk cache. The cache items file lists each client file, what server file, and what portion of that file the local client file stores. By caching the descriptors in memory, the cache items file needs to be read less often. It is not recommended that this option be set by an administrator.

| **-chunksize**
| This controls the size of each client file used to store remote server files and how much data can be read or written in a single RPC. A large chunk size is preferred. The default is 64K for disk cache and 8K for memory cache.

| **-namecachesize**
| This is the size of the name lookup cache. The default is 256.

| The following environment variable controls client caching:

| **_IOE_CM_DIRCACHE_SIZE**
| This controls the amount of storage the DFS client uses to store directory contents. The default is 1M.

| **Diagnostics:** The following sample output is from the CMCACHE report. It shows the performance of the client caches.

```

|                                     DFS Client Cache Performance
|                                     -----
| Status Cache Statistics
| -----
| Total Entries      300      Entry Size 376 bytes.
| Total Searches     4226     Hits      3964 (Ratio 93.80%)
| Total Flushes      0
|
| Data Cache Statistics
| -----
| Total Files        512      Chunk Size 64K bytes.
| Total Entries      512
| Total Searches     5588     Hits      5150 (Ratio 92.16%)
| Total Recycles     0
| Total Bytes Read From Cache      0 (KB)
| Total Bytes Written To Cache    0 (KB)
|
| Directory Name Lookup Statistics
| -----
| Total Entries      256      Entry Size 80 bytes.
| Total Searches     5589     Hits      4457 (Ratio 79.75%)
|
| Token Management Statistics
| -----
| Status Tokens Acquired      265      Released      212      Revoked      0
| Data Tokens Acquired      265      Released      212      Revoked      0
|
| Connection Statistics
| -----
| Binding Handles Created      19      Waited      538
|
| Directory Cache Statistics
| -----
| Total Entries      28      Entry Size 8192 bytes.
| Total Reads      114      Hits      64 (Ratio 56.14%)
| Reads for directories too large to cache: 0

```

| The following sections provide explanations of the CMCACHE report.

| **Status Cache Statistics:** This section shows the number of status cache entries in the cache (Total Entries) and the status cache hit ratio (Hits/Ratio fields). The hit ratio should be 80% or higher. A miss requires one or more DCE/RPCs to be issued to obtain the status.

| **Data Cache Statistics:** This section shows the number of data cache files (Total Files) and the size of each "file" (given by Chunk Size). For memory cache, each file is in memory and the total cache storage equals the Total Files times the Chunk Size. For disk cache clients, not all cache files are stored in memory, but a certain number of cache descriptors, which indicate where the files are on disk are in memory. This is the Total Entries fields. The important statistic is the Total Searches and the Hits/Ratio fields. A high cache hit ratio is recommended because a miss requires an RPC.

| **Directory Name Lookup Statistics:** This section indicates the number of directory lookups and the percent of time those lookups were found in the name lookup cache. The Total Entries is the size of the cache and the Total Searches and Hits/Ratio fields indicate the percentage of time the name was found in the cache. A high hit ratio is recommended because a miss requires one or more DCE/RPCs made to the file server.

| **Directory Cache Statistics:** This section shows the current cache size and indicates the number of directory buffer reads and the percent of time the directory data was cached at the client, (Total Reads and Hits/Ratio, respectively). A high hit ratio is recommended because a miss requires a DCE/RPC. In this case, the directories that are larger than 8K are not cached by the client and always result in remote RPC calls. If your installation has many directories larger than 8K your hit ratio is lower and increasing directory cache size does not help. The field Reads for directories too large to cache indicates how often this occurs. An acceptable directory cache hit ratio would be at least 60% or higher.

RPC I/O Performance

DFS makes DCE RPC calls whenever it cannot satisfy a request locally. Performance of the DFS client is often directly related to the performance of the DFS client cache and the response times of the DCE RPC requests (which includes remote server response time). If the client can satisfy a request locally, without any RPC, then the client response time is fast (less than 1 millisecond). However, when an RPC is required, response time is much larger.

Local Server: If the server resides on the same system as the client then the client bypasses DCE to make the RPC call and makes the call directly using a local UNIX socket. This eliminates any wire transmission and significantly reduces pathlengths. Therefore, if the server is local, the response time is often dominated by the server response time itself.

Background Threads: When performing file read/write requests, the client uses a pool of background I/O threads to perform read-ahead and write-behind. By processing large file read/write requests on separate threads in parallel with user application processing, large file read/write performance is improved substantially. If the pool of threads is busy then a read/write request will either be queued waiting for a background thread to become free, which adds queue wait time to the response time, or will be performed by the user application task synchronously. The administrator can set the size of the background I/O thread pool and monitors performance of the thread pool.

Tuning Options: The only tuning option is the size of the background I/O thread pool which is specified in the DFS client startup parameters.

-mainprocs

This sets the number of background I/O threads. The default is 5.

Diagnostics: The following sample out is from the AFS4INT report. This report shows performance of the background thread pool, the count and average response times of RPCs made to the local server on the same system as the client, and the count and average response times of RPCs made to remote servers.

| AFS4 Interface RPC Call Counts

Average RPC Time (msecs)		
RPC Call	Count	Response Time
-----	-----	-----
AFS_Lookup	1	15.447
AFS_LookupRoot	2	168.591
AFS_FetchStatus	1	14.589
AFS_StoreStatus	1	39.415
AFS_FetchData	1	14.023
AFS_StoreData	0	0.000
AFS_CreateFile	0	0.000
AFS_RemoveFile	0	0.000
AFS_Rename	0	0.000
AFS_Readdir	1	47.987
AFS_MakeDir	0	0.000
AFS_RemoveDir	0	0.000
AFS_Link	0	0.000
AFS_Symlink	0	0.000
AFS_FetchACL	0	0.000
AFS_StoreACL	0	0.000
AFS_GetToken	1	14.714
AFS_ReleaseTokens	0	0.000
AFS_SetContext	20	40.866
AFS_SetParams	1	12.683
AFS_GetTime	0	0.000
AFS_MakeMountPoint	0	0.000
AFS_BulkKeepAlive	0	0.000
AFS_BulkFetchVV	0	0.000
AFS_GetStatistics	0	0.000
AFS_BulkFetchStatus	0	0.000

| Total RPC calls 29
 | Average Response Time per RPC 45.288

| Derived Fetch/Store Statistics:

| Total Fetched Bytes 119K
 | Total Stored Bytes 0K

| AFS4 Interface LOCAL RPC Call Counts

Average Local RPC Time (msecs)		
RPC Call	Count	Response Time
-----	-----	-----
L.AFS_Lookup	566	5.725
L.AFS_LookupRoot	0	0.000
L.AFS_FetchStatus	1	0.827
L.AFS_StoreStatus	0	0.000
L.AFS_FetchData	7	35.541
L.AFS_StoreData	545	48.077
L.AFS_CreateFile	193	16.510
L.AFS_RemoveFile	212	34.112
L.AFS_Rename	51	7.959
L.AFS_Readdir	49	3.341
L.AFS_MakeDir	35	52.045
L.AFS_RemoveDir	35	12.435
L.AFS_Link	5	3.351
L.AFS_Symlink	20	12.995
L.AFS_FetchACL	0	0.000
L.AFS_StoreACL	0	0.000
L.AFS_GetToken	6	10.953
L.AFS_ReleaseTokens	8	1.377
L.AFS_BulkFetchStatus	0	0.000

| Total Local RPC calls 1733
 | Average Response Time per LOCAL RPC 24.980

```

|           Background Thread Statistics
|           -----
| Number of threads: 16 (stack size=32K)
| Number of background requests 493
| Number of background requests queued 24 (4.8%)
| Number of background requests failed 5
| Number of background requests waited 0

```

| The following describes the sections of the AFS4INT report.

| **AFS4 Interface RPC Call Counts:** This reports shows the average response time of remote procedure calls. Two tables are shown, one for remote server calls and one for calls made to a file server that reside on the same physical machine as the client (if one exists). Normally, local RPCs are much faster than remote RPCs since no wire transmission is made. The response time of a remote procedure call is not controllable by the client. Network performance and file server performance determines that. This information is shown merely so an administrator can determine what affect RPC response time is having on local user response time. The administrator can only set larger caches in order to reduce RPC calls made by using DFS tuning options. Faster network hardware improves RPC response times and network bottlenecks would be reflected in these numbers.

| **Background Thread Statistics:** The Number of background requests and Number of background requests queued indicate how many times a file read/write request was sent to a background I/O thread for processing and how often the request had to be queued. It is recommended, in this case, to have a low amount of queueing because background queue waits contribute to increased end user response time and RPC requests are themselves waiting for RPC completion (which are long term waits).

| Disk Caching Considerations

| All of the information described above is relevant for clients that use disk caching rather than memory caching. However, there are additional considerations when using a disk cache. File data is stored on a utility LFS aggregate, then the performance of LFS is very relevant to the performance of the client disk cache. Also, because file data is stored in the virtual memory cache associated with LFS, the performance of the VM cache is important to disk cache performance. The description of LFS and VM cache performance in the **Server Tuning Guidelines** section is relevant for clients with disk cache. The local user client access hits the VM cache and LFS much harder than a server will because there are no RPC delays between requests. This makes performance of these components much more critical.

| The performance of the VM cache is critical for client disk cache performance. The administrator should set the size of the VM cache to ensure a recommended hit ratio of at least 80%. If this is accomplished this allows performance of disk cache clients to approach memory cache clients but with much less memory used. Additionally, the LFS utility aggregate are hit heavily. It is recommended to place this aggregate on a disk that is otherwise unused or has a low amount of activity not related to the LFS utility aggregate. Finally, the size of the LFS metadata cache should be set large enough to ensure at least a 90% hit ratio.

| Locking

| The DFS client performs most processing on a UNIX System Services user task. This task runs in synchronous cross memory mode. This means that standard POSIX locking cannot be used and the processing code used for UNIX System Services tasks uses a different lock package than the rest of the DFS product. Appended to the LOCK report that is shown for the server is a report indicating lock waits and average lock wait time for this code.

| The following sample output is from the LOCK report.

```

|           DFS XMEM Lock Statistics
|           -----
| TREAD lock waits:      0   avg. time    0.000 (msecs)
| TWRITE lock waits:    0   avg. time    0.000 (msecs)
| TLOCK lock waits:     0   avg. time    0.000 (msecs)
| Sleeps:               17   avg. time   10.961 (msecs)
| -----
| Total waits           17   avg. time   10.961 (msecs)
| Wakeups:              0   Wait Failures: 0

```

Some Final Considerations

Production sized distributed file system performance is affected by many factors. The network and disk hardware are a large portion of end user response time. A bottleneck in either the network or the disk systems will reduce end user response time. Tuning DFS properly places less of a strain on both the network and disk systems.

Eliminating a bottleneck might uncover other bottlenecks. For example, assume that a particular file system environment has a severe network bottleneck. If the network bottleneck is fixed you might actually notice that DFS response times increase. The reason is because the faster the RPCs or SMBs arrive to the server the longer it takes the server to process them. When the network is very slow, the server often has little work to do and therefore has threads available to process work. It is also less likely that a disk I/O wait will occur. When work is arriving quickly to a server (particularly from the same client) there is a good chance of lock contention and CPU waits. But, there is a greater chance of disk I/O waits and the disk I/O wait time increasing. This is because the more work a server is processing, the more likely that when a thread needs to wait on an I/O to disk there are other disk I/Os ahead of it on the device queue. This makes response times longer because not only is the thread waiting for the disk, but it is waiting for other I/Os to the disk to complete first. This is seen in a DCE/RPC workload. If the client and server reside on the same machine, RPCs are bypassed. If you write large files from a DCE/DFS client to a DFS server on the same machine, from the end user's point of view, response time is much better than writing large files from another client located on a different physical machine. If you view the DFS statistics for each run you will see that DFS response times are greater for the local user case. The elimination of the network overhead makes requests arrive at the DFS server sooner and when a disk I/O wait occurs it is likely other I/Os are ahead of it in the queue and lock contention occurs more frequently.

Therefore, eliminating one bottleneck might uncover another. The faster the network, the more important DFS server tuning becomes.

Appendix E. Notices

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

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

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

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

Trademarks

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

AIX/6000	BookManager	IBM
IBMLink	Library Reader	OpenEdition
OS/2	OS/390	RACF
MVS/ESA		

Microsoft is a registered trademark of the Microsoft Corporation.

| UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Bibliography

This section lists and provides a brief description of each publication in the OS/390 Distributed File Service library. Also listed are publications from the OS/390 DCE library that may be useful.

OS/390 Distributed File Service Publications

This section lists and provides a brief description of each publication in the OS/390 Distributed File Service (DFS) library.

Administration

- *OS/390 Distributed File Service DFS Configuring and Getting Started*, SC28-1722
- *OS/390 Distributed File Service DFS Administration Guide and Reference*, SC28-1720

This book helps system and network administrators configure the OS/390 Distributed File Service.

This book introduces the DFS concepts to system and network administrators and provides an in-depth understanding of DFS, its uses and benefits. This book also provides reference information for the commands and files used by system and network administrators to work with DFS.

- *OS/390 Distributed File Service SMB Administration Guide and Reference*, SC24-5882

This book provides guidance and reference information for system and network administrators to use when they work with the Server Message Block (SMB) support of the IBM OS/390 Distributed File Service base element of OS/390. SMB is a protocol for remote file/print access used by Windows.

Reference

- *OS/390 Distributed File Service Messages and Codes*, SC28-1724

This book provides detailed explanations and recovery actions for the messages, status codes, and exception codes issued by the OS/390 DFS.

OS/390 DCE Publications

This section lists and provides a brief description of each publication in the OS/390 DCE library.

Overview

- *OS/390 DCE Introduction*, GC28-1581
- This book introduces OS/390 DCE. Whether you are a system manager, technical planner, OS/390 system programmer, or application programmer, it will help you understand DCE, and evaluate the uses and benefits of including OS/390 DCE as part of your information processing environment.

Planning

- *OS/390 DCE Planning*, SC28-1582
- This book helps you plan for the organization and installation of OS/390 DCE. It discusses the benefits of distributed computing in general, and describes how to develop plans for a distributed system in an OS/390 DCE environment.

Administration

- *OS/390 DCE Configuring and Getting Started*, SC28-1583

This book helps system and network administrators configure OS/390 DCE.

- *OS/390 DCE Administration Guide*, SC28-1584

This book helps system and network administrators understand OS/390 DCE, and tells how to administer it from the batch, TSO, and shell environments.

- *OS/390 DCE Command Reference*, SC28-1585

This book provides reference information for the commands that system and network administrators use to work with OS/390 DCE.

- *OS/390 DCE User's Guide*, SC28-1586

This book describes how to use OS/390 DCE to work with your user account, use the directory service, work with namespaces, and change access to objects that you own.

Application Development

- *OS/390 DCE Application Development Guide: Introduction and Style*, SC28-1587

This book assists you in designing, writing, compiling, linking, and running distributed applications in OS/390 DCE.

- *OS/390 DCE Application Development Guide: Core Components*, SC28-1588

This book assists programmers in developing applications using application facilities, threads, remote procedure calls, distributed time service, and security service.

- *OS/390 DCE Application Development Guide: Directory Services*, SC28-1589

This book describes the OS/390 DCE directory service and assists programmers in developing applications for the cell directory service and the global directory service.

- *OS/390 DCE Application Development Reference*, SC28-1590

This book explains the DCE Application Program Interfaces (APIs) that you can use to write distributed applications on OS/390 DCE.

Reference

- *OS/390 DCE Messages and Codes*, SC28-1591

This book provides detailed explanations and recovery actions for the messages, status codes, and exception codes issued by OS/390 DCE.

SecureWay Security Server Publications

This section lists and provides a brief description of books in the SecureWay Security Server library that may be needed for the DCE SecureWay Security Server and for RACF interoperability.

- *OS/390 Security Server (DCE) Overview*, GC28-1938

This book describes the DCE security server and provides a road map for DCE security server information in the OS/390 DCE library.

- *OS/390 SecureWay Security Server RACF Security Administrator's Guide*, SC28-1915.

This book explains RACF concepts and describes how to plan for and implement RACF.

Index

Special Characters

- blocks 175
- files 175
- mainprocs 175
- memcache 175
- namecachsize 175
- stat 175
- _IOE_CM_DIRCACHE_SIZE 176
- _IOE_CM_REQUEST_THREADS 175
- _IOE_EPI_CACHE_SIZE 176
- _IOE_MVS_IDMAP environment variable 96
- _IOE_MVS_IDMAP_SAF environment variable 98
- _IOE_RFS_WORKER_THREADS 175
- _IOE_SMB_MAIN_POOL 175
- _IOE_SMB_MAXXMT 180
- _IOE_SMB_OPLOCKS 180
- _IOE_SMB_PROTOCOL_LEVEL 180
- _IOE_SMB_RAW 179
- _IOE_VM_CACHE_SIZE 176
- .profile file 77

A

- access control 5
- accessing
 - Identity Mapping files 97
- accounts
 - DCE
 - creating for DFS server administrator 84
 - creating for user 83
 - using DCE dcecp to create 83
- acct-admin group 5
- ADAPTERS report 177
- adding
 - DFS administrative principals to DCE registry 48
 - members to admin.bos 89
 - principals and groups to administrative lists 89
 - replication site DCD LFS fileset 112
 - replication site for root.dfs 109
- address space 131
- admin.bak file 86, 91
- admin.bos file 86, 89, 90
- admin.fl file 86, 90
- admin.ft file 86, 91
- admin.up file 86, 91
- administrative domains 6
- administrative lists
 - adding groups, members or principals to 89
 - creating 87
 - definition 6
 - types of 86
 - viewing members 89

administrator

- creating DCE account for 84

administrator accounts

- creating DCE accounts for 83
- security subsystem 92

AFS4INT report 176, 177

aggregates

- DCE local file system 103
- HFS 101
- RFS 102

alias

- creating for backup data set high-level qualifier 71
- default value 71
- where specified 71

ALL report 177, 178

anonymous user 3, 98

- unauthenticated requests mapped to 3

application programming interfaces 1

authorization groups

- creating 5

authorized programs 26, 27

B

background threads 205

Backup Database Server

- completing configuration 68
- configuring 58
- deconfiguring 120
- example creating, starting, stopping 137
- using DFSCONF 58

Backup Tape Coordinator

- completing configuration 69
- environment variables 70

bak commands 16

bakserver

- See Backup Database Server

Basic OverSeer Server 15, 44

- completing configuration

bibliography 211

bkserver 15

bkserver process

- See Backup Database Server

books, list of DCE and related 211

bos commands 16

BosConfig file 119

bosserver 15

- See also Basic OverSeer Server

bufhigh 176

BUTC 15

- See also Backup Tape Coordinator

C

CacheInfo file 21, 132

caching 175

- considerations 207
- directory 203
- file 202
- status 203
- tuning options
 - blocks 175
 - files 175
 - memcache 175
 - namecachesize 175
 - stat 175
 - _IOE_CM_DIRCACHE_SIZE 176
 - _IOE_EPI_CACHE_SIZE 176
 - _IOE_VM_CACHE_SIZE 176
- bufhigh 176

CDS (Cell Directory Server)

See Cell Directory Server (CDS)

cell

- filepace 5
- planning 5

Cell Directory Server (CDS) 69

changing

- DCE host name 128
- Identity Mapping entries 97

client caches 202

cm commands 16

CMCACHE report 177

CMSESV report 177

colony address space 9

commands

- bak 16
- bos 15, 16
- cm 16
- dfsexport 15, 16
- fts 16
- growaggr 16
- modify 105, 135
- newaggr 16
- QUERY 176
- RESET 178
- START 134
- STOP 133, 134
- udebug 16

completing

- Backup Database machine configuration 68
- bakserver configuration 51
- bakserver deconfiguration 121
- boserver
- butc configuration 52
- configuration
 - Backup Tape Coordinator 69
 - bakserver 68
 - BUTC 69
 - DFS Client 72

completing *(continued)*

configuration *(continued)*

- DFS File Server 61
- flserver 61
- ftserver 63
- rpserver 65
- upclient 65
- upserver 66

deconfiguration

- Backup Database Server 121
- Fileset Database Server 122
- DFS Client configuration 51
- DFS post installation processing 48
- flserver configuration 51
- ftserver configuration 51
- repserver configuration 51
- upclient configuration 51
- upserver configuration 51

configuring

- Backup Database server 58
- data set considerations 20
- DFS 47
- DFS client (DFSCM) 49, 73
- DFS client with memory caching 74
- DFS considerations 43
- DFS File Server 58
- DFS Servers 57
- Fileset Database server 59
- Fileset Location Servers 44
- overview xv, 47
- using the Server Configuration Program (DFSCONF)
 - example log file 141
 - steps, for Backup Database server 58
 - steps, for Fileset Database server 59

configuring DFS client panel 60

configuring DFS servers panel 57

considerations

- anonymous user 98
- before deconfiguring DFS 117
- DCE cell 19
 - DFS administrator 85
 - DFS server principal 85
- DCE cell configuration 19
- DFS client (DFSCM) 9, 45
- DFS configuration 43
- DFS customizable and configuration data set 20
- DFS server 13, 43
- DFSCM environment variable 77
- DFSCM user .profile file 77
- host system 19
- migration 20
- re-installation 20
- setting up filesets 6
- test DCE cell 19
- tuning 174
- upserver deconfiguration 118

creating

- additional DFS administrators 52
- admin.bos file 89
- aggregates 101
- alias in OS/390 master catalog 71
- data set 113
- DCE accounts
 - for DFS server administrator 84
 - for user 83
 - steps 83
- DCE LFS aggregate 52
- DCE local file system aggregates 103
- devtab file for root.dfs 107
- DFS administrative authorization lists 50
- DFS administrator 87
- DFS File Server FLDB entry 63
- DFS mount point for DCE LFS fileset 112
- DFS mount point for HFS fileset 114
- DFS mount point for RFS fileset 116
- dfstab file for root.dfs 107
- fileset for DCE LFS aggregate 111
- FLDB entry DFS File Server 51
- FLDB entry for HFS aggregate 113
- FLDB entry for RFS aggregate 115
- general use HFS aggregate 53
- general use RFS aggregate 53
- Identity Mapping file 95
- identity mapping input file (dfsimap) 93
- mount points for filesets 109
- MVS mount point for HFS aggregate 114
- RACF DCEUUIDS APPLDATA OS/390 ID entries 98
- RACF DCEUUIDS profile 97
- read/write mount point for root.dfs 108
- root.dfs fileset 108
- symbolic links 147
- VSAM data sets for DCE LFS aggregate 107
- VSAM data sets for DCE LFS aggregate for root.dfs 110
- VSAM tape backup management file 71

cron process 14

customizable files 21, 149

customizing

- DFS envar files 53
- DFS server envar files 80
- DFSCM envar file 81
- environment variables (envar) files 80
- ioepdcf file 52, 78

D

daemon configuration file 13

daemons

- BUTC 15
- for DFS client 9
- starting using modify command 135

daemons (continued)

- starting using START command 134
- stopping using modify command 135
- stopping using STOP command 134
- who can start or stop 133

data cache statistics 204

data normalization 178

data set

- creating 113

DCE

- accounts
 - for DFS server administrator 84
 - user, creating 83
- cell
 - DFS administrator considerations 85
 - DFS server principal considerations 85
- cell name 48
- configuration considerations 19
- considerations 19
- environment variables 153
- host name 48
- login 48
- user IDs
 - for MVS 92

DCE compatibility with MVS

- HFS 3

DCE local file system 103

aggregate

- creating fileset 111
- creating VSAM data sets 110
- exporting 111
- updating devtab file 111
- updating dfstab file 111

fileset

- adding replications site 112
- creating DFS mount point 112
- initiating schedule replication 112
- setting replication type 112

DCE login panel 55, 56

DCE User IDs, mapping to OS/390 user IDs 93

DCE_START_SOCKET_NAME 154

deconfiguring

- bakserver 120
- considerations before 117
- DFS client 75, 126
- DFS client (DFSCM) 119
- DFS component 119
- DFS File Server 123
- DFS servers 119, 124
- flserver 121
- ftserver 123
- overview xv, 117
- upserver considerations 118

deconfiguring DFS Client panel 126

deconfiguring servers panel 124

default mapping 94

defining

DFS administrators and users 83
root.dfs fileset for DCE cell 52

deleting

Identity Mapping entries 97

determining

DCE cell name and DCE host name 48
DCE login information 48

devtab file 101

DFS administration utilities 16

DFS administrative list files

admin.bak 86
admin.bos 86
admin.fl 86
admin.ft 86
admin.up 86

DFS administrator 83

creating 87

DFS client (DFSCM) 9

completing configuration 72
configuring 60, 73
considerations 45
deconfiguring 119, 126
disk caching 75
memory caching 74
minimum configuration 74
starting 52
system parmlib member BPXPRMxx entry 72
using DFSCONF 59

DFS client (DFSCM) memory caching 74

DFS Client Machine (DFSCM)

using OS/390 UNIX to start 131
who can start and stop 131

DFS client processes 9

ioedfsd process 10

DFS Control Task (dfscntl) 14, 43, 78, 134

DFS directories

in /ect/dfs 148
in /usr/lpp/dfs/global 147

DFS File Server

completing configuration 61
configuring 58
deconfiguring 123

DFS files

in /etc/dfs 148
in /usr/lpp/dfs/global 147

DFS server tuning 179

DFS Servers

access to, granting administrator 89
administrator
 creating DCE account for 84
allowing foreign user access 97
configuring 57
deconfiguring 119, 124
starting
 initializing 78

DFS Servers (*continued*)

starting (*continued*)
 starting 134
stopping using modify command 135
stopping using STOP command 134

DFS user 83

dfs-admin group 5

dfs-bak-servers group 5

dfs-fs-servers group 5

dfs_cpfiles program 21, 41, 149

DFSC 72

DFSCM (DFS client)

See DFS client (DFSCM)

dfscntl

See DFS Control Task (dfscntl)

DFSCONF

bakserver 120
configuration log file 54
environment variables 54
flserver 121
ftserver 123
login panel 55
main menu 56, 57
using 53
 configure bakserver 58
 configure DFS Client 59
 configure DFS File Server 58
 configure flserver 59
 deconfigure DFS Client 126

DFSCONF main menu 57

dfsconf.log file 54

dfscpfil 41

dfsexport command 16

dfskern

process 102
separate address space 20

dfstab file 103

diagnostics

ADAPTER 191
AFS4INT 205
CMCACHE 203
CMSERV 201
LFS 194
LOCK 199
RFS 197
SMBCOMM 183
SMBINT 180
SMBPRT 181
SMBTKC 187
STORAGE 200
TKM 186
VM 188

directories

in /opt/dfsglobal 147
in /usr/dfslocal 148

directory

caching 203

directory cache statistics 204

directory name lookup statistics 204

disabling

NoAuth file 88

disk caching 74

Distributed File Service (DFS)

address space 13

administration programs, defined 16

administration utilities, defined 16

administrative lists 6

administrator 83

application programming interfaces 1

benefits 1

commands 16

creating aggregates 101

creating DCE local file system aggregates 103

customizable considerations 20

domains 6

environment variables 154

environment variables, listing of 153

exporting data 101

exporting file system data 103

features 1

introduction 1

machine roles 6

processes 13

restarting 52

supporting year 2000 20

user 83

domains 6, 85

E

enabling

DFSCM to use DCE single sign-on 77

NoAuth file 88

envvar file 21

environment variables

_EUV_AUTOLOG 153

_EUV_ECHO_STDIN 153

_EUV_ENVAR_FILE 153

_EUV_FTRACE 153

_EUV_HOME 73, 153

_EUV_SEC_KRB5CCNAME_FILE 153

_EUV_SVC_DBG_MSG_LOGGING 154

_EUV_SVC_MSG_LEVEL 53, 54, 154

_EUV_SVC_MSG_LOGGING 53, 54, 154

_IOE_BACKUP_TAPE_CAPACITY 155

_IOE_BOSERVER_ENVAR 155

_IOE_BOSERVER_OUTPUTDD 155

_IOE_BOSERVER_PGMNAME 155

_IOE_BUTC_DISKUNIT 71, 155

_IOE_BUTC_DUMP_HLQ 71, 156

_IOE_BUTC_KSDS 71, 156

environment variables (continued)

_IOE_BUTC_TAPEUNIT 71, 156

_IOE_CFG_CELL_ID 54, 156

_IOE_CFG_INFORM_LEVEL 54, 156

_IOE_CFG_LOG_FILE 54, 157

_IOE_CM_ATTRIBUTES_FILE 157

_IOE_CM_ATTRIBUTES_FILESETS 157

_IOE_CM_DIRCACHE_SIZE 158, 176, 203

_IOE_CM_LOGIN_PROCESSES 158

_IOE_CM_PARMS 158

_IOE_CM_REQUEST_THREADS 158, 175

_IOE_CM_REQUESTS_THREADS 201

_IOE_DAEMONS_IN_AS 159

_IOE_DFS_MODIFY_PATH 159

_IOE_DIRECTORY_CACHE_SIZE 159, 191

_IOE_EPI_CACH_SIZE 194

_IOE_EPI_CACHE_SIZE 159, 176

_IOE_HFS_ATTRIBUTES_FILE 102, 160

_IOE_HFS_TRANSLATION 102, 161

_IOE_LFS_IO_SEEK 194

_IOE_LFS_SYNC_INTERVAL 162

_IOE_MVS_DFSDFLT 162

_IOE_MVS_IDMAP 96, 162

_IOE_MVS_IDMAP_SAF 98, 162

_IOE_MVS_SERVER 162

_IOE_PROTOCOL_RPC 163

_IOE_PROTOCOL_SMB 163

_IOE_RFS_ALLOC_TIMEOUT 163

_IOE_RFS_ATTRIBUTES_FILE 163

_IOE_RFS_STATUS_REFRESH_TIME 164

_IOE_RFS_TRANSLATION 164

_IOE_RFS_WORKER_THREADS 164, 175, 197

_IOE_SERVER_ENV_NAME 164

_IOE_SMB_BROWSE_INTERVAL 165

_IOE_SMB_CALLBACK_POOL 165, 183

_IOE_SMB_CLEAR_PW 165

_IOE_SMB_COMPUTER_NAME 165

_IOE_SMB_DESCRIPTION 166

_IOE_SMB_DIR_PERMS 166

_IOE_SMB_DOMAIN_NAME 166

_IOE_SMB_FILE_PERMS 166

_IOE_SMB_IDLE_TIMEOUT 167

_IOE_SMB_IDMAP 167

_IOE_SMB_MAIN_POOL 167, 175, 182

_IOE_SMB_MAXXMT 167, 180

_IOE_SMB_NT_SMBs 168, 180

_IOE_SMB_OPLOCK_TIMEOUT 168

_IOE_SMB_OPLOCKS 168, 180

_IOE_SMB_PRIMARY_WINS 168

_IOE_SMB_PROTOCOL_LEVEL 169, 180

_IOE_SMB_RAW 169, 179

_IOE_SMB_SCOPE 169

_IOE_SMB_SECONDARY_WINS 169

_IOE_SMB_TOKEN_FILE_MAX 170, 186

_IOE_SMB_WINS_PROXY 170

_IOE_TKGLUE_CACHE_SIZE 186

environment variables *(continued)*

- _IOE_TKM_MAX_TOKENS 170, 185
- _IOE_TKMGLUE_CACHE_SIZE 170
- _IOE_TKMGLUE_SERVER_THREADS 170, 186
- _IOE_VM_CACHE_SIZE 171, 176, 188
- _IOE_VM_MAX_FILES 171, 188
- _IOE_VNODE_CACHE_SIZE 171, 191
- BIND_PE_SITE 154
- customizing 80
- DCE 153
- DCE_START_SOCKET_NAME 154
- DFS 154
- DFSCONF 54
- example 55
- for Backup Tape Coordinator 70
- KRB5CCNAME 154
- LIBPATH 154
- listing of 153
- NLSPATH 54, 154
- setting DFSCONF 55
- SVC_CDS_DBG 154
- SVC_DTS_DBG 154
- SVC_PLT_DBG 154
- SVC_RPC_DBG 154
- SVC_SEC_DBG 154
- syntax 153
- TZ 154

examples

- creating, starting, stopping
 - bakserver 137
 - flserver 137
 - ftserver 137
 - repserver 138
 - upclient 139
 - upserver 138
- devtab file 113, 115
- devtab file entry 101
- dfs_cpfiles 42
- dfstab entry 105
- dfstab file entry 111
- dfstab file entry for root.dfs 108
- envvar file DFSCONF program 55
- identity mapping input file (dfsimap) 94
- ioepdcf file 79
- MAPID program JCL 95
- mount HFS fileset 114
- updating
 - admin.bak file 91
 - admin.bos file 90
 - admin.fl file 90
 - admin.ft file 91
 - admin.up file 91
- using translation control parameter 102

export process 14

exporting

- DCE LFS aggregate 52, 111

exporting *(continued)*

- DCE local file system data 105
- files system data using DFS 103
- general use HFS aggregate 53
- general use RFS aggregate 53
- HFS aggregate 114
- non-local file system data 104
- RFS aggregate 116
- root.dfs aggregate 108

F

file

- caching 202

files

- .profile 77
- admin.bak 86
- admin.bos 86, 89
- admin.fl 86
- admin.ft 86
- admin.up 86
- BosConfig 119
- CacheInfo 132
- customizable 149
- devtab 101
- DFS administrative list 86
- DFS Server Configuration Program log 141
- DFS tape backup management file 71
- DFSCM configuration 131, 132
- dfsconf.log 54
- dfstab 103
- envvar 49
- Identity Mapping 95
- in /opt/dfsglobal 147
- ioepdcf 50, 52, 78

Fileset Database Server 44

Fileset Location Database (FLDB) Server

- completing configuration 61
- example creating, starting, stopping 137
- using DFSCONF 59

Fileset Location Database Server 44

Fileset Location Server 44

Fileset Server

- completing configuration 63
- configuring 59
- deconfiguring 121
- example creating, starting, stopping 137

filesets

- HFS 83
- limitations 6
- names limited to 102 characters 6
- naming 6
- recommendations for setting up 6
- RFS 83
- setting up 6

filespace 5
FILESYSTYPE 72
flserver 15
See also Fileset Location Database (FLDB) Server
fts command 16, 104
ftserver 15
See also Fileset Server

G

global planning 5
groups
 acct-admin 5
 adding to administrative lists 89
 created during configuration 5
 dfs-admin 5
 dfs-bak-servers 5
 dfs-fs-servers 5
growaggr command 16
guidelines
 performance 173
 tuning 173

H

HFS (Hierarchical File System) 3, 101
See also Hierarchical File System (HFS)
Hierarchical File System (HFS) 3, 92, 101
 aggregate
 creating 53
 creating MVS mount point 114
 exporting 53, 114
 steps to create and export 113
 updating devtab file 113
 updating dfstab file 114
 fileset
 creating DFS mount point 114
 example 114
 inconsistent token state, avoiding 3
 limitations with DFS 3
 mapping user IDs 92
host name
 changing 128
host system
 considerations 19
 removing DFS 127

I

I/O balancing 176
Identity Mapping file
 accessing 97
 creating 95
 creating the input file (dfsimap) 93
 default mapping to a DFS server 94
 deleting entries in 97
 modifying entries in 97

Identity Mapping file (*continued*)
 security 97
 specific mapping to a DFS server 94
identity mapping input file (dfsimap), creating 93
initiating
 release replication of root.dfs 109
 schedule replication DCE LFS fileset 112
installation 41
 creating symbolic links 147
 DCE on host system 48
 DFS on host system 48
 new release of DFS 21
 post 41
IOEBKSRV load module 15
IOEBOSRV load module 15
IOEBUTC load module 15
ioecmini process 9, 131
IOEDFSCL load module 14
ioedfsd process 9, 10, 131
IOEDFSKN load module 14
IOEDFSXP load module 14
IOEFLSRV load module 15
IOEFTSRV load module 15
ioelogin process 9, 131
IOEP0001 13
IOEP0002 9
IOEP0003 13
ioepdcf (daemon configuration) file
 configuration types specified in 78
 considerations for existing installations 78
 considerations for new installations 78
 customizing 52, 78
 example of 79
 processes specified in 78
ioepdcf file 14, 21, 50, 52, 78
IOEPDCT 14
IOERPSRV load module 15
IOEUPCLN load module 15
IOEUPSRV load module 15

J

JCL (job control language)
 See Job Control Language (JCL)
Job Control Language (JCL) 103

K

KRB5CCNAME 154

L

LFS report 177, 178
LIBPATH 154
linear data sets 103

load module

- IOEBKSRV 15
- IOEBOSRV 15
- IOEBUTC 15
- IOEDFSCL 14
- IOEDFSKN 14
- IOEDFSXP 14
- IOEFLSRV 15
- IOEFTSRV 15
- IOERPSRV 15
- IOEUPCLN 15
- IOEUPSRV 15

local server 205

locking 207

LOCKING report 177, 178

log files

- example of configuration program 141

M

mapping

- creating identity mapping file 95
- DCE user IDs to MVS user IDs 92
- default mapping to a DFS server 94
- definition 83
- identity mapping input file (dfsimap), creating 93
- specific mapping to a DFS server 94

memory caching 74

migration

- to new DFS release 20

minimum DFS client (DFSCM) configuration 74

modify system command 14, 105, 135

- using to start processes 135

modifying

- DFS envvar files 49
- Identity Mapping entries 97
- ioepdcf file 50

N

name lookup cache 203

newaggr command 16

NLSPATH 154

NoAuth file

- disabling 88
- enabling 88

O

OS/390 DFS create default DFS configuration files 41

OS/390 master catalog 71

OS/390 User IDs, mapping to DCE user IDs 92

overview

- configuration xv, 47
- deconfiguration xv, 117
- DFS installation and configuration 41

P

panels

- configuring DFS client panel 60
- configuring DFS servers 57
- DCE login 55, 56
- deconfiguring DFS server (DFSCM) 126
- deconfiguring servers 124
- DFSCONF main menu 57

parameter library (parmlib) 74

partitioned data set (PDS)

- member names for dfscntl processes 14

PDS (partitioned data set)

- See partitioned data set (PDS)

performance

- guidelines 173
- RPC I/O 205

planning

- for access control 5
- global 5
- questions to consider 5

post installation 41

post installation processing

- completing DFS 48

principals

- adding to administrative lists 89

processes

- bksrvr 15
- bosrvr 15
- bosrvr-controlled 136, 139
- cron 14
- DFS client 9
- dfskern 102
- export 14
- flsrvr 15
- ftsrvr 15
- ioecmini 9, 131
- ioedfsd 9, 131
- ioelgin 9, 131
- rpserver 15
- salvage 16
- simple 14
- unexport 14
- upclnt 15
- upsvr 15

proclib member

- IOEP0002 9
- IOPE0001 13
- IOPE0003 13

programs

- authorized 26, 27
- dfs_cpfiles 41, 42, 149
- scout 16

Q

QUERY command 176

R

RACF 20, 92

RACF DCEUIDs

APPLDATA OS/390 ID entries 98
profile 97

re-installation considerations 20

Record File System (RFS) 3, 92, 102

aggregate
creating 53
creating and exporting 115
creating FLDB entry 113, 115
exporting 53, 116
updating devtab file 115
updating dfstab file 116

fileset
creating DFS mount point 116
mapping user IDs 92

registering

users with MVS security subsystem 92

Registry Editor

using to create DCE account 83

Remote Procedure Calls (RPC) 2, 45

removing

DFS Client from host system 127
DFS from host system 127

Replication Server

completing configuration 65
example creating, starting, stopping 138

reports

DFS client
AFS4INT 177
ALL 178
CMCACHE 177
CMSERV 177
LFS 178
LOCKING 178
STORAGE 178
VM 177

DFS/SMB server
ADAPTERS 177
AFS4INT 176
ALL 177
LFS 177
LOCKING 177
RFS 177
SMBCOMM 177
SMBINT 176
SMBPRT 176
SMBTKC 177
STORAGE 177
TKM 177
VM 177

repserver

See Replication Server

RESET command 178

restarting

DFS 52

RFS (Record File System) 3, 102

See also Record File System (RFS)

RFS report 177

root fileset (root.dfs)

See also root.dfs (root fileset)
location of 6

root.dfs (root fileset) 6, 106, 109

creating 108
creating devtab file 107
creating dfstab file 107
creating VSAM data sets for DFS LFS
aggregate 110
example dfstab file entry 108
read/write mount point 108
release replication 109
replication site 109
replication type 109

RPC (Remote Procedure Call)

See Remote Procedure Calls (RPC)

RPC I/O

performance 205

rpserver 15

rpserver process

See Replication Server

S

salvage process 16

scout program 16

security 5

security subsystem 92

registering users with 92

separate address space

dfskern 20

Server Message Block (SMB)

See SMB (Server Message Block)

servers

See DFS Servers

setting

_IOE_MVS_IDMAP_SAF environment variable 96
Backup Tape Coordinator environment variables 70
DFSCONF administrator TSO user ID 55
DFSCONF command 49
DFSCONF environment variables 55
OS/390 shell flag 49
replication type DCE LFS fileset 112
replication type for root.dfs 109
root.dfs fileset 106
user IDs for HFS and RFS access 52

simple process 14

- single sign-on** 77
- SIOELMOD data set** 14
- SMB (Server Message Block)**
- SMBCOMM report** 177
- SMBINT report** 176, 180
- SMBPRT report** 176, 181
- SMBTKC report** 177
- specific mapping** 94
- START system command** 14, 134
- starting**
 - boserver 50
 - boserver-controlled processes 136
 - DFS client 52
 - DFS server 50
 - using modify command 135
 - using START command 134
 - who can start daemons 133
 - DFSCM client 131
 - order for DFS daemons 134
- statistics**
 - background thread 207
 - data cache 204
 - directory cache 204
 - directory name lookup 204
 - status cache 204
- status cache statistics** 204
- steps**
 - before proceeding with DFS configuration 48
 - configuration
 - Backup Database server 58
 - bakserver 68
 - boserver 61
 - BUTC 69
 - DFS 48
 - DFS Client 72
 - DFS File Server 61
 - Fileset Database server 59
 - flserver 61
 - ftserver 63
 - rpserver 65
 - upclient 65
 - upserver 66
 - create and export DCE LFS aggregate 110
 - creating and exporting HFS aggregate 113
 - creating and exporting RFS aggregate 115
 - creating mount points for filesets 109
 - DCE host name change 128
 - DCE host name change while DFS configured 128
 - deconfiguration
 - bakserver 120
 - DFS client (DFSCM) 119
 - DFS File Server 123
 - DFS servers 119
 - flserver 121
 - ftserver 123
 - define a root.dfs fileset 106

- steps (continued)**
 - exporting data 101
 - installing new release of DFS 21
 - post installation 41
 - re-establishing bakserver object UUID 72
 - reconfiguring when changing DCE host name 128
 - removing DFS from host system 127
- STOP system command** 49, 133, 134
- stopping**
 - boserver-controlled processes 139
 - DFS 48
 - DFS client 48
 - DFS server
 - using modify command 135
 - using STOP command 134
 - who can stop daemons 133
 - DFSCM client 131
- STORAGE report** 177, 178
- support**
 - year 2000 20
- SVC_CDS_DBG** 154
- SVC_DTS_DBG** 154
- SVC_PLT_DBG** 154
- SVC_RPC_DBG** 154
- SVC_SEC_DBG** 154
- symbolic links**
 - listing 147
- syntax**
 - environment variables 153
- system control machines** 85

T

- TapeConfig** 70
- threading** 175
 - tuning options
- TKM report** 177
- tuning**
 - considerations 174
 - DFS server 179
 - guidelines 173
 - threading 175
- tuning options** 179, 203, 205
- TZ** 154

U

- udebug command** 16
- unexport process** 14
- unsupported services and commands** xvi
- upclient** 15
 - See also* Update Client
- Update Client**
 - completing configuration 65
 - example creating, starting, stopping 139

Update Server

- completing configuration 66
- example creating, starting, stopping 138

updating

- devtab file for DCE LFS aggregate 111
- devtab file for HFS aggregate 113
- devtab file for RFS aggregate 115
- dfstab file for DCE LFS aggregate 111
- dfstab file for HFS aggregate 114
- dfstab file for RFS aggregate 116

files

- admin.bak 91
- admin.bos 90
- admin.fl 90
- admin.ft 91
- admin.up 91

upserver 15

See also Update Server

user

- creating DCE account for 83
- registering with MVS security subsystem 92

user ID

- mapping DCE to OS/390 93

using

- DCE dcecp to create DCE account 83
- DFS commands in TSO 3
- DFSCONF
 - configure bakserver 58
 - configure DFS Client 59
 - configure DFS File Server 58
 - configure flserver 59
- DFSCONF configure DFS servers 50
- DFSCONF to configure DFS client 49
- HFS 3
- modify command 135
- NoAuth file 89
- Server Configuration Program (DFSCONF)
 - for Backup Database server 58
 - for Fileset Database server 59
- START command 134
- STOP command 134

V

validating

- DCE configured for DFS on host system 49

variables, environment

- _EUV_AUTOLOG 153
- _EUV_ECHO_STDIN 153
- _EUV_ENVAR_FILE 153
- _EUV_FTRACE 153
- _EUV_HOME 73, 153
- _EUV_SEC_KRB5CCNAME_FILE 153
- _EUV_SVC_DBG_MSG_LOGGING 154
- _EUV_SVC_MSG_LEVEL 53, 54, 154
- _EUV_SVC_MSG_LOGGING 53, 54, 154

variables, environment (continued)

- _IOE_BACKUP_TAPE_CAPACITY 155
- _IOE_BOSERVER_ENVAR 155
- _IOE_BOSERVER_OUTPUTDD 155
- _IOE_BOSERVER_PGMNAME 155
- _IOE_BUTC_DISKUNIT 71, 155
- _IOE_BUTC_DUMP_HLQ 71, 156
- _IOE_BUTC_KSDS 71, 156
- _IOE_BUTC_TAPEUNIT 71, 156
- _IOE_CFG_CELL_ID 54, 156
- _IOE_CFG_INFORM_LEVEL 54, 156
- _IOE_CFG_LOG_FILE 54, 157
- _IOE_CM_ATTRIBUTES_FILE 157
- _IOE_CM_ATTRIBUTES_FILESETS 157
- _IOE_CM_DIRCACHE_SIZE 158, 176, 203
- _IOE_CM_LOGIN_PROCESSES 158
- _IOE_CM_PARS 158
- _IOE_CM_REQUEST_THREADS 158, 175
- _IOE_CM_REQUESTS_THREADS 201
- _IOE_DAEMONS_IN_AS 159
- _IOE_DFS_MODIFY_PATH 159
- _IOE_DIRECTORY_CACHE_SIZE 159, 191
- _IOE_EPI_CACH_SIZE 194
- _IOE_EPI_CACHE_SIZE 159, 176
- _IOE_HFS_ATTRIBUTES_FILE 102, 160
- _IOE_HFS_TRANSLATION 102, 161
- _IOE_LFS_IO_SEEK 194
- _IOE_LFS_SYNC_INTERVAL 162
- _IOE_MVS_DFSDFLT 162
- _IOE_MVS_IDMAP 96, 162
- _IOE_MVS_IDMAP_SAF 98, 162
- _IOE_MVS_SERVER 162
- _IOE_PROTOCOL_RPC 163
- _IOE_PROTOCOL_SMB 163
- _IOE_RFS_ALLOC_TIMEOUT 163
- _IOE_RFS_ATTRIBUTES_FILE 163
- _IOE_RFS_STATUS_REFRESH_TIME 164
- _IOE_RFS_TRANSLATION 164
- _IOE_RFS_WORKER_THREADS 164, 175, 197
- _IOE_SERVER_ENV_NAME 164
- _IOE_SMB_BROWSE_INTERVAL 165
- _IOE_SMB_CALLBACK_POOL 165, 183
- _IOE_SMB_CLEAR_PW 165
- _IOE_SMB_COMPUTER_NAME 165
- _IOE_SMB_DESCRIPTION 166
- _IOE_SMB_DIR_PERMS 166
- _IOE_SMB_DOMAIN_NAME 166
- _IOE_SMB_FILE_PERMS 166
- _IOE_SMB_IDLE_TIMEOUT 167
- _IOE_SMB_IDMAP 167
- _IOE_SMB_MAIN_POOL 167, 175, 182
- _IOE_SMB_MAXXT 167, 180
- _IOE_SMB_NT_SMBS 168, 180
- _IOE_SMB_OPLOCK_TIMEOUT 168
- _IOE_SMB_OPLOCKS 168, 180
- _IOE_SMB_PRIMARY_WINS 168

variables, environment *(continued)*

- _IOE_SMB_PROTOCOL_LEVEL 169, 180
- _IOE_SMB_RAW 169, 179
- _IOE_SMB_SCOPE 169
- _IOE_SMB_SECONDARY_WINS 169
- _IOE_SMB_TOKEN_FILE_MAX 170, 186
- _IOE_SMB_WINS_PROXY 170
- _IOE_TKGLUE_CACHE_SIZE 186
- _IOE_TKM_MAX_TOKENS 170, 185
- _IOE_TKMGLUE_CACHE_SIZE 170
- _IOE_TKMGLUE_SERVER_THREADS 170, 186
- _IOE_VM_CACHE_SIZE 171, 176, 188
- _IOE_VM_MAX_FILES 171, 188
- _IOE_VNODE_CACHE_SIZE 171, 191
- BIND_PE_SITE 154
- customizing 80
- DCE 153
- DCE_START_SOCKET_NAME 154
- DFS 154
- DFSCONF 54
- example 55
- for Backup Tape Coordinator 70
- KRB5CCNAME 154
- LIBPATH 154
- listing of 153
- NLSPATH 54, 154
- setting DFSCONF 55
- SVC_CDS_DBG 154
- SVC_DTS_DBG 154
- SVC_PLT_DBG 154
- SVC_RPC_DBG 154
- SVC_SEC_DBG 154
- syntax 153
- TZ 154

verifying

- server entry for DFS server 107

VM report 177

VSAM tape backup management file 71

W

- workloads 179

Y

- year 2000 20

Readers' Comments

OS/390®
Distributed File Service DFS
Configuring and Getting Started
Publication No. SC28-1722-06

You may use this form to report errors, to suggest improvements, or to express your opinion on the appearance, organization, or completeness of this book.

Date: _____

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

Note

Report system problems to your IBM representative or the IBM branch office serving you.
U.S. customers can order publications by calling the IBM Software Manufacturing Solutions at
1-800-879-2755.

In addition to using this postage-paid form, you may send your comments by:

FAX	1-607-752-2327	Internet	pubrcf@vnet.ibm.com
IBMLink	GDLVME(PUBRCF)		

Would you like a reply? ☐ **YES** ☐ **NO** If yes, please tell us the type of response you prefer.

☐ Electronic address: _____

☐ FAX number: _____

☐ Mail: (Please fill in your name and address below.)

Name

Address

Company or Organization

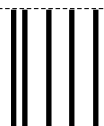
Phone No.



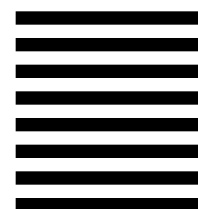
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department G60
International Business Machines Corporation
Information Development
1701 North Street
ENDICOTT NY 13760-5553



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5647-A01



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC28-1722-06





OS/390

DFS Configuring and Getting Started