Distributed File Service
SMB Administration
z/OS

Distributed File Service
SMB Administration
Tenth Edition (September 2009)

This is a major revision of SC24-5918-08.

This edition applies to Version 1 Release 11 of z/OS (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM® welcomes your comments. A form for readers’ comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation
MHVRCFS, Mail Station P181
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
  Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrfs@us.ibm.com

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 document
• Page number or topic related to your comment

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

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

Figures ................................................................. ix

Tables ................................................................. xi

About this document .................................................. xiii
Who should use this document .................................... xiii
How this document is organized .................................... xiii
Conventions used in this document ................................ xiii
Where to find more information ..................................... xiv
  Softcopy publications ............................................ xiv
  Internet sources .................................................... xiv
  Information updates on the web ................................... xiv
  The z/OS Basic Skills Information Center ....................... xv

Summary of Changes .................................................. xvii

Part 1. SMB support administration guide ............................. 1

Chapter 1. An overview of SMB support ................................. 3
  SMB support features ............................................. 3
  SMB processes ...................................................... 4
  Shared directories ................................................. 4
  Shared printers ..................................................... 4
  Command structure and help ...................................... 5
  Command shortcuts ............................................... 5
  Receiving help .................................................... 6

Chapter 2. Considerations for a new SMB release ....................... 7
  New release considerations ....................................... 7
  SMB configuration file considerations .......................... 9

Chapter 3. SMB post installation processing .......................... 11
  Installing and configuring SMB file and print server .......... 11
  Defining SMB administrators ..................................... 15
  Creating the default DFS configuration files ................... 15
    Steps for using dfs_cpf files program ......................... 15
  Defining RACF definitions for SMB .............................. 17
    Setting up a new SMB installation to run under a UID(0) TSO user .......... 17
    Setting up new SMB installation to run under a unique nonzero UID TSO user .......... 18
    Modifying an existing SMB installation to run with a unique nonzero UID .......... 19

Chapter 4. Managing SMB processes .................................... 21
  Who can start and stop DFS server daemons? .................... 22
  Starting DFS server daemons .................................... 23
    The MODIFY DFS operator command ............................ 23
    Using MODIFY DFS to start DFS server daemons ............... 24
    Order of starting DFS server daemons ........................ 24
  Stopping DFS ..................................................... 24
    Using MODIFY DFS to stop DFS server daemons ................ 25
  Viewing the status of DFS server daemons ....................... 25
  Starting DFS server daemons during IPL ........................ 26
  Daemon configuration file ....................................... 26
  How dfsctrl starts the DFS server daemons ..................... 27

© Copyright IBM Corp. 1999, 2009
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using the -nodfs option to start dfscntl</td>
<td>28</td>
</tr>
<tr>
<td>Changing environment variables</td>
<td>28</td>
</tr>
<tr>
<td>Changing mappings</td>
<td>28</td>
</tr>
<tr>
<td>Changing shared directories or shared printers</td>
<td>28</td>
</tr>
<tr>
<td>Changing the hfsattr or the rfstab</td>
<td>28</td>
</tr>
<tr>
<td>Changing the Infoprint Server DLL</td>
<td>29</td>
</tr>
<tr>
<td>Chapter 5. Networking considerations</td>
<td>31</td>
</tr>
<tr>
<td>Chapter 6. Mapping SMB user IDs to z/OS user IDs</td>
<td>33</td>
</tr>
<tr>
<td>Creating an smbidmap file</td>
<td>33</td>
</tr>
<tr>
<td>Setting the _IOE_SMB_IDMAP environment variable</td>
<td>34</td>
</tr>
<tr>
<td>Modifying and deleting identity mapping entries</td>
<td>34</td>
</tr>
<tr>
<td>Determining the z/OS user ID from the SMB user ID</td>
<td>34</td>
</tr>
<tr>
<td>How the SMB user ID is determined</td>
<td>35</td>
</tr>
<tr>
<td>Chapter 7. Sharing files</td>
<td>37</td>
</tr>
<tr>
<td>Sharing HFS files</td>
<td>37</td>
</tr>
<tr>
<td>Exporting and sharing HFS file systems</td>
<td>37</td>
</tr>
<tr>
<td>smbtab, dfstab, and devtab entries for HFS</td>
<td>40</td>
</tr>
<tr>
<td>Creating a shared directory for HFS</td>
<td>41</td>
</tr>
<tr>
<td>Removing a shared directory for HFS</td>
<td>42</td>
</tr>
<tr>
<td>Dynamic export for HFS</td>
<td>42</td>
</tr>
<tr>
<td>File data translation for HFS</td>
<td>46</td>
</tr>
<tr>
<td>Authorization for HFS</td>
<td>48</td>
</tr>
<tr>
<td>File data translation for RFS</td>
<td>48</td>
</tr>
<tr>
<td>Authorization for RFS</td>
<td>52</td>
</tr>
<tr>
<td>Free space for RFS</td>
<td>53</td>
</tr>
<tr>
<td>Logon considerations</td>
<td>53</td>
</tr>
<tr>
<td>Using Windows Terminal Server as a client to the z/OS SMB server</td>
<td>54</td>
</tr>
<tr>
<td>Using passthrough authentication</td>
<td>55</td>
</tr>
<tr>
<td>Adjusting the authentication level on Vista and XP</td>
<td>56</td>
</tr>
<tr>
<td>RACF DCE segments for SMB encrypted password support</td>
<td>57</td>
</tr>
<tr>
<td>Chapter 8. Sharing printers</td>
<td>59</td>
</tr>
<tr>
<td>Steps for creating a shared printer</td>
<td>59</td>
</tr>
<tr>
<td>Steps for removing a shared printer</td>
<td>59</td>
</tr>
<tr>
<td>Print data translation</td>
<td>60</td>
</tr>
<tr>
<td>Authorization</td>
<td>60</td>
</tr>
<tr>
<td>Chapter 9. Locating the SMB server</td>
<td>61</td>
</tr>
<tr>
<td>Set up the SMB server</td>
<td>61</td>
</tr>
<tr>
<td>Steps for Windows XP and Vista using DNS, WINS, or LMHOSTS file</td>
<td>61</td>
</tr>
<tr>
<td>Find the SMB server</td>
<td>63</td>
</tr>
<tr>
<td>Steps for using Network and Sharing Center on Windows Vista</td>
<td>63</td>
</tr>
<tr>
<td>Steps for using My Network Places on Windows XP</td>
<td>63</td>
</tr>
<tr>
<td>Steps for using search to find the server on Windows XP</td>
<td>63</td>
</tr>
<tr>
<td>Chapter 10. Accessing data</td>
<td>65</td>
</tr>
<tr>
<td>Using SMB server shared directories</td>
<td>65</td>
</tr>
<tr>
<td>Figures</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1. Example output of dfs_cpfiles</td>
<td>17</td>
</tr>
<tr>
<td>2. DFS server address space</td>
<td>21</td>
</tr>
<tr>
<td>3. DFSKERN in a separate address space</td>
<td>22</td>
</tr>
<tr>
<td>4. Daemon configuration file</td>
<td>27</td>
</tr>
<tr>
<td>5. Example of simplified file hierarchy on a z/OS system</td>
<td>38</td>
</tr>
<tr>
<td>6. Example of separate file systems that are mounted together</td>
<td>38</td>
</tr>
<tr>
<td>7. Example of three file systems</td>
<td>39</td>
</tr>
<tr>
<td>8. Example of where file systems are mounted</td>
<td>39</td>
</tr>
<tr>
<td>9. SMB server with a single user session on each communications session</td>
<td>54</td>
</tr>
<tr>
<td>10. SMB server with multiple, concurrent users on a single communications session</td>
<td>55</td>
</tr>
<tr>
<td>11. Linux mount and unmount commands issued from a Linux UID 0 user</td>
<td>66</td>
</tr>
<tr>
<td>12. SMB server components</td>
<td>119</td>
</tr>
<tr>
<td>13. Typical thread states</td>
<td>125</td>
</tr>
<tr>
<td>14. Possible abnormal thread state</td>
<td>125</td>
</tr>
<tr>
<td>15. Sample configuration file</td>
<td>165</td>
</tr>
</tbody>
</table>
Tables

1. IOEDFSCL and IOEPDCT daemon return codes ........................................ 21
2. IOEDFSKN and IOEDFSXP daemon return codes ........................................ 21
3. LAN Manager authentication settings and descriptions .......................... 56
4. Data set creation attributes ................................................................. 85
5. Processing attributes ........................................................................... 87
6. Site attributes ....................................................................................... 89
7. Environment variables in SMB ............................................................. 143
8. DFS customizable files ......................................................................... 171

© Copyright IBM Corp. 1999, 2009
About this document

The purpose of this document is to provide complete and detailed guidance and reference information. This information is used by system and network administrators working with the Server Message Block (SMB) support of the IBM z/OS Distributed File Service base element of z/OS.

Distributed File Service includes an SMB function that is based on the X/Open SMB Version 2 specification and the IETF RFCs on Netbios over IP (RFC1001 and RFC1002).

The Distributed File Service base element supports both Distributed Computing Environment (DCE) DFS™ file protocols and SMB file/print protocols. This document focuses on the SMB support of z/OS Distributed File Service. If you are using only SMB protocols, then you should use this document. If you are using DCE DFS protocols, see z/OS Distributed File Service DFS Administration. If you are using both protocols (DCE DFS and SMB), you may need to refer to both documents.

Who should use this document

This document is intended for users and network and system administrators who understand the basic concepts of data communications. A knowledge of Transmission Control Protocol/Internet Protocol (TCP/IP) communications, z/OS UNIX® operating system concepts, and Microsoft Windows (referred to as Windows throughout this document) operating system concepts can help you use this guide more effectively.

How this document is organized

The information in this document is divided into the following parts, each part divided into chapters.

Part 1, “SMB support administration guide,” on page 1 discusses guidance information. The chapters in that part begin with a short introduction and are followed by detailed information.


Conventions used in this document

This document uses the following typographic conventions

**Bold** words or characters represent system elements that you must enter into the system literally, such as commands.

*Italic* italicized 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 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 indicated that you can repeat the preceding item one or more times.

1. Server Message Block (SMB) is a protocol for remote file/print access used by Microsoft® Windows® clients. This protocol is also known as Common Internet File System (CIFS).
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 document 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 positioned for ease of readability.

A pound sign is used to indicate a command is entered from the shell, specifically where **root** authority is needed (**root** refers to a user with a **UID = 0**).

This document used 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”, with a left arrow.

**Entering commands**

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

### Where to find more information

Where necessary, this document references information in other documents. For complete titles and order numbers for all elements of **z/OS**, refer to **[z/OS Information Roadmap](#)**.

For information about installing Distributed File Service components, refer to **[z/OS Program Directory](#)**.

Information concerning Distributed File Service-related messages can be found in **[z/OS Distributed File Messages and Codes](#)**.

### Softcopy publications

The **z/OS** Distributed File Service library is available on a CD-ROM, **z/OS Collection**, SK3T-4269. The CD-ROM online library collection is a set of documents for **z/OS** and related products that includes the IBM Library Reader™. This is a program that enables you to view the BookManager® files. This CD-ROM also contains the Portable Document Format (PDF) files. You can view or print these files with the Adobe Acrobat reader.

### Internet sources

The softcopy **z/OS** publications are also available for web-browsing and for viewing or printing PDFs using the following URL: [http://www.ibm.com/systems/z/os/zos/bkserv](http://www.ibm.com/systems/z/os/zos/bkserv)

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

### Information updates on the web

For the latest information updates that have been provided in PTF cover letters and Documentation APARs for **z/OS**, see the online document at: [http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/ZDOCAPAR](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/Shelves/ZDOCAPAR)

This document is updated weekly and lists documentation changes before they are incorporated into **z/OS** publications.
The z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a Web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS system programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

• Provide basic education and information about z/OS without charge
• Shorten the time it takes for people to become productive on the mainframe
• Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your Web browser to the following Web site, which is available to all users (no login required): http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp
Summary of Changes

Summary of Changes for SC24-5918-09
z/OS Version 1 Release 11

This document contains information previously presented in z/OS Distributed File Service SMB Administration, SC24-5918-07, which supports z/OS Version 1 Release 10 and subsequent releases.

New information

- New information about "The z/OS Basic Skills Information Center" on page xv.
- In V1R11, the SMB server cannot export zFS file systems when zFS is running sysplex-aware. For details, see "New release considerations" on page 7.
- You can now allow the SMB server to dynamically recognize client security permission changes by setting the _IOE_USE_PTHREAD_SECURITY environment variable to ON. See _IOE_USE_PTHREAD_SECURITY in Appendix A, "Environment variables in SMB," on page 143.
- SMB Security Administrators now need to permit the SMB Server userid to the BPX.DAEMON and BPX.SERVER FACILITY class. See z/OS Migration and Chapter 3, "SMB post installation processing," on page 11.
- SMB now supports Windows Vista Business and Windows Vista Enterprise. For server details, see "Set up the SMB server" on page 61. For client details, see Chapter 10, "Accessing data," on page 65. Note that for Windows Vista you might have to define your domain name, which was not necessary in previous releases.
- The modify subcommand now supports reset, print, and resize of the internal trace table. For details, see "modify dfs processes" on page 78, "TRACE command" on page 124, and Appendix A, "Environment variables in SMB," on page 143.
- SMB now supports NT LAN Manager version 2 (NTLMv2) authentication protocol. For details, see the default value for _IOE_SMB_PROTOCOL_LEVEL in Appendix A, "Environment variables in SMB," on page 143 and the topics "Using passthrough authentication" on page 55 and "Adjusting the authentication level on Vista and XP" on page 56.
- New information describing "TRACE command" on page 124 and "SMB hang detection" on page 124.

Changed information

- You now have the option of running SMB with a unique nonzero UID. See "Defining RACF definitions for SMB" on page 17.
- Additional information about SMB support of Windows Vista in found throughout this documentation.

You may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

This document 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 SC24-5918-08
z/OS Version 1 Release 10

This document contains information previously presented in z/OS Distributed File Service SMB Administration, SC24-5918-07, which supports z/OS Version 1 Release 9 and subsequent releases.
New information

- SMB now performs automatic syntax checking of /opt/dfslocal/home/dfsckenv. See "dfssyntax" on page 116.
- The DFSKERN level of dispatching priority. See "Installing and configuring SMB file and print server" on page 11.
- SMB supports hang detection. See Appendix A, "Environment variables in SMB," on page 143 for the following environment variables:
  - _IOE_HANG_DETECT_DUMP_LIMIT
  - _IOE_HANG_DETECT_INTERVAL
  - _IOE_HANG_DETECT_OENODE_TIMEOUT
  - _IOE_HANG_DETECT_THREAD_TIMEOUT
- New information about resolving "Insufficient space errors" on page 164.

Changed information


This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Summary of Changes
for SC24-5918-07
z/OS Version 1 Release 9

This document contains information previously presented in z/OS Distributed File Service SMB Administration, SC24-5918-06, which supports z/OS Version 1 Release 8 and subsequent releases.

New information

- Windows Terminal Server (WTS) on Windows 2003 and Windows 2003 domain controllers are now supported.
- A new appendix is added, Chapter 15, "Tuning and debugging guidelines," on page 119.

Changed information

- The default value for "_IOE_VNODE_CACHE_SIZE" on page 162 is changed.

Deleted information

- z/OS.e is not supported as of z/OS V1R9.
- The following are no longer supported:
  - WTS on Windows 2000
  - Windows 2000 clients
  - Windows 2000 domain controller

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.
Part 1. SMB support administration guide

This section covers guidance information for system administrators and Personal Computer (PC) users. Specifically, Chapter 1 is intended for PC users and system administrators, Chapters 2 through 8 are intended for system administrators, and Chapters 9 through 11 are for PC users.

- Chapter 1, “An overview of SMB support,” on page 3
- Chapter 2, “Considerations for a new SMB release,” on page 7
- Chapter 3, “SMB post installation processing,” on page 11
- Chapter 4, “Managing SMB processes,” on page 21
- Chapter 5, “Networking considerations,” on page 31
- Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33
- Chapter 7, “Sharing files,” on page 37
- Chapter 8, “Sharing printers,” on page 59
- Chapter 9, “Locating the SMB server,” on page 61
- Chapter 10, “Accessing data,” on page 65
Chapter 1. An overview of SMB support

The Distributed File Service Server Message Block (SMB)\(^2\) support provides a server that makes Hierarchical File System (HFS) files and data sets available to SMB clients. The data sets supported include sequential data sets (on DASD)\(^3\), partitioned data sets (PDS), partitioned data sets extended (PDSE) and Virtual Storage Access Method (VSAM) data sets. The data set support is typically referred to as Record File System (RFS) support. To access shared directory paths and shared printers, the SMB protocol is supported through the use of TCP/IP on z/OS. Personal Computer (PC) clients on the network use the file and print sharing functions that are included in their operating systems. Supported SMB clients include:

- Windows Vista
- Windows XP Professional
- Windows 2003
- Windows Terminal Server on Windows 2003
- SUSE LINUX with Samba (SUSE LINUX Enterprise Server 9 (s390) - Kernel 2.6.5-7.202.5-s390 with Samba 3.0.14a-0.4)
- Redhat Linux\(^\circ\) with Samba (Redhat Linux 2.4.20-43.9.legacy with Samba 3.0.20-2).

At the same time, these files can be shared with local z/OS UNIX applications and with DCE DFS clients.

**Note:** Throughout this documentation, references are made to HFS. Unless otherwise stated, HFS is a generic reference that includes HFS, zFS, TFS, and AUTOMNT file system data. If you are in a sysplex with Shared HFS, SMB support of zFS is limited to zFS compatibility mode file systems.

In addition, Windows SMB clients can make remote print requests to z/OS printers that are connected to the Infoprint Server for z/OS.

### SMB support features

PC users work with files on their computers. Files are used to store data, programs, and other information. Files are stored on a disk on the computer. There can be several disks on the computer. Each of these disks are referred to by a different drive letter (for example, A:, B:, C:, D:, etc.). PC Users can read or write files on different disks on their computer by using the appropriate drive letter in the file name (for example, D:\dir1\file1).

PC users also work with printers attached to their computers. When a print request is made, the PC user can choose which printer the print request should go to.

### Disclaimer

The z/OS SMB server supports basic file and print serving. It does not necessarily support all functions that a Windows file server supports. For example, the z/OS SMB server does not support Kerberos authentication or DFS. There might be other functions that are not supported.

Using SMB support, PC users can access files that are on a z/OS system. That is, PC users can access files that are not located on their computer. Remote files are visible to the PC user on one or more separate drive letters. PC users can “connect” an unused drive letter to a “shared resource” on a remote computer. This is sometimes called “mapping a network drive”. This capability is provided by software that

---

\(^2\) Server Message Block (SMB) is a protocol for remote file/print access used by Windows clients. This protocol is also known as Common Internet File System (CIFS).

\(^3\) Direct Access Storage Device
resides on the PC (the client), in combination with software that resides on the remote computer (the server). There must also be a TCP/IP network connection between the PC and the remote computer.

In addition, Windows PC users are able to use remote printers that are attached to a z/OS system. Remote printers are visible as additional printers available to the PC user. Remote printers are installed on PCs using existing commands or install utilities.

**SMB processes**

SMB support provides a server process that makes file data and printers available to PC users. Administrators can define shared directories and shared printers. It also handles PC requests to connect to the server process to satisfy file or print requests.

Another SMB process is the control process (also known as the DFS control task). It oversees the server process. When SMB support is started, it is really the DFS control task that is started. The DFS control task, in turn, starts the server process. If the server process ends abnormally for some reason, the DFS control task can automatically restart the server process.

**Shared directories**

A PC accesses files located on a z/OS system through one or more shared directories on the z/OS system. Distributed File Service administrators make files available to SMB clients by creating shared directories. A shared directory is given a share name and specifies a directory path name in a file system. Any directory can be shared with clients. To access shared directories from a PC, clients can map a network drive by choosing an available drive letter and mapping it to a computer name and a share name, or they can use Universal Naming Convention (UNC) mapping. (See Chapter 10, “Accessing data,” on page 65 for more information on UNC mapping.) The computer name is the name of the Distributed File Service SMB server and the share name is the name of the shared directory created on that server. After this is done, the remote files can be read or written as though they were local files.

**Shared printers**

Distributed File Service administrators make z/OS printers available to Windows PCs by creating shared printers. A shared printer is given a share name and specifies a z/OS Infoprint Server printer definition name. To access a remote printer from a Windows PC, clients can install a remote printer or they can use commands. After this is done, the remote printers can be used as though they were local printers.

**Command structure and help**

The SMB commands share a similar structure. The following example shows the basic format of an SMB command:

```
$ command [(-option1 | -option2 argument)]
```

The following example illustrates the elements of an SMB command:
The following list summarizes the elements of an SMB command:

**Command**
A command consists of the command name. This name directs the server process or program to perform a specific action. The command name always appears in bold font.

**Options**
Command options appear in bold font, are always preceded by a - (dash), and are often followed by arguments. In the previous example, -all and -share are options, and name is the argument. The { | } (braces separated by a vertical bar) indicate that you can enter only one of two possible options. An option and its arguments tell the server process or program which entities to manipulate when executing the command (for example, the name to assign to the shared directory or printer). In general, you should provide the options for a command in the order presented in the documentation.

**Arguments**
Arguments for options always appear in italic font.

**Optional Information**
Some commands can have optional, as well as required, options. Optional information is enclosed in [ ] (brackets). The -all and the -share with its name argument in the previous example are optional.

Enter each SMB command and its options and arguments on a single line followed by a carriage return at the end of the line. Use a space to separate each element (command name, options, and arguments) on the command line. Also use spaces to separate multiple arguments. Do not use a space to separate an option from its - (dash).

**Command shortcuts**
When supplying an argument (such as name in the previous example), you can omit the option (such as -share in the example) associated with the argument if:
- All arguments supplied with the command are entered in the order in which they appear in the command’s syntax. The syntax for each command is presented with its description in Part 2, “SMB support reference,” on page 75.
- Arguments are supplied for all options that precede the option to be omitted.
- All options that precede the option to be omitted accept only a single argument.
- No options, either those that accept an argument or those that do not, are supplied before the option to be omitted.
In the case where two options are presented in { | } (braces separated by a vertical line), the option associated with the first argument can be omitted if that argument is provided; however, the option associated with the second argument is required if that argument is provided.

If you must provide an option, you can abbreviate it to the shortest possible form that distinguishes it from other options of the command. For example, the -share option can typically be omitted or abbreviated to be -s.

The following example illustrates three acceptable ways to enter the same dfsshare command:

- Complete command:
  $ dfsshare -share name
- Abbreviated command name and abbreviated option:
  $ dfsshare -s name
- Abbreviated command name and omitted option:
  $ dfsshare name

**Receiving help**

You can receive help on the SMB commands by using the -help option:

$ command -help
Chapter 2. Considerations for a new SMB release

If you have an earlier release of the Distributed File Service installed on your system and you plan to install a new release of the Distributed File Service to use the SMB support, it is important to follow the instructions in [z/OS Migration] and review the following topics:

- "New release considerations"
- "SMB configuration file considerations" on page 9.

New release considerations

When migrating to a new release of Distributed File Service, it is not necessary to copy and save your customized files before installing. The files you can customize are listed in Appendix D, "Customizable files," on page 171.

Some of the customizable files listed might not be present on your system when installing a new release of Distributed File Service (unless you are reinstalling this version) because the files are new for this release. Most customizable files that do not exist are created as part of Distributed File Service post installation by the dfs_cpfies program from sample files provided in the /opt/dfsglobal/examples directory. There are several files that you need to create (if this is your first installation of Distributed File Service) in order to specify the HFS data and shared printers to be made available to PC clients. These include the smtab, dfstab, and the devtab files. Also, the smbimap file must be created to map PC user IDs to z/OS user IDs.

Review the following list before installing a new release of Distributed File Service:

1. Distributed File Service zFS considerations
   Beginning with z/OS V1R11, the SMB server cannot export zFS read/write file systems when zFS is running sysplex-aware on the same system where the SMB server is running or on the system that owns the zFS file system. If you want to export zFS file systems using the SMB server, ensure zFS is configured as non-symplex aware (specifying sysplex=off in the zFS IOEFSPRM configuration file).
   For additional information, see
   - The topics on What's new in zFS for V1R11 and Using the SMB server in [z/OS Distributed File Service zSeries File System Administration]
   - The topic on Distributed File Service migration actions in [z/OS Migration].

2. Beginning with z/OS V1R11, to provide greater security, the SMB server requires access to BPX.DAEMON and BPX.SERVER. For additional information, see [z/OS Migration].

3. DCE DFS considerations
   If you previously used DCE DFS, see “DFS Migration Considerations” in [z/OS Distributed File Service Customization]. It is recommended that you migrate the DFS in a DCE environment before completing any additional steps to migrate the SMB support to the new release.
   If you have already migrated DFS to the new release and you want to activate SMB support for the first time, you should follow the instructions for "Installing and configuring SMB file and print server on page 11.

4. Symbolic link considerations
   The Distributed File Service customizable files are in the /etc/dfs path. During installation, symbolic links are deleted and recreated to link to files in /etc/dfs.
   If you have replaced any Distributed File Service symbolic links, they are deleted during installation. You should save the file data before installing the Distributed File Service. These files are listed in Appendix D, "Customizable files," on page 171 and the topic on Directories and files.

5. Installing Distributed File Service
   If you have not already done so, install the current release of Distributed File Service using the instructions in [z/OS Program Directory].
6. Stopping the Distributed File Service server

If the Distributed File Service server is running on the target image, stop the server at this time using the operator command `stop dfs`. See Chapter 4, “Managing SMB processes,” on page 21 for more information.

7. RACF® database considerations

If you are installing into a target image that uses the same RACF database as the production image, the preexisting SMB configuration files that were included when the production `/etc` file system was copied by using the instructions in z/OS Program Directory do not need to be modified to change the RACF user IDs. You need to only review and update the target image copy of the configuration files with optional parameters new for this release later in the migration process. Socket option access control is provided with the SERVAUTH class resource profile EZB.SOCKOPT.sysname.tcpname.SO_BROADCAST. If this profile is defined, the SMB server userid must be permitted at least READ authority to this profile.

If the target image uses a different RACF database, the `smbidmap` file defined by the `__IOE_SMB_IDMAP` environment variable in the `/opt/dfslocal/home/dfskern/envar` file must be updated to define the correct user identifiers. See Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33 for more details. You might also need to review the `__IOE_MVS_DFSDFLT` environment variable in the `/opt/dfslocal/home/dfskern/envar` file. See “Logon considerations” on page 53 for more details.

8. Exported data considerations

If you are installing into a target image that can access the same exported file data as the production image, the preexisting `smbtab`, `dfstab`, and `devtab` files in the `/opt/dfslocal/var/dfs` directory can be used.

If the target image cannot access the same exported file data as the production image, or if you do not want to export the production data files during the testing of the new z/OS release on the target image, you must update these files to export and share only test data.

9. Printer considerations

If you are installing into a target image that can access the same printers as the production image, the preexisting `smbtab` file in the `/opt/dfslocal/var/dfs` directory can be used.

If the target image cannot access the same printers as the production image, or if you do not want to share the production printers during the testing of the new z/OS release on the target image, you must update this file to create different shared printers.

10. Running the `/opt/dfsglobal/scripts/dfs_cpfiles` program

Some of the customizable files listed in Appendix D, “Customizable files,” on page 171, might not exist in the `/etc/dfs` file system on the target image when installing a new release of the Distributed File Service (unless you are reinstalling this release). Customizable files that do not exist are created as part of post installation processing by the `dfs_cpfiles` program from the sample files in the `/opt/dfsglobal/examples` directory.

If you have not already run the `dfs_cpfiles` program following the instructions in z/OS Program Directory, you should run it now against the target image `/etc/dfs` to ensure the new release has all the new configuration files.

More information about the `dfs_cpfiles` program can be found in Chapter 3, “SMB post installation processing,” on page 11.

Note: The `dfs_cpfiles` program does not create all the files necessary for SMB support. See “SMB configuration file considerations” on page 9 for more information.

11. Updating existing configuration files and server startup parameters

Compare your target system customizable files with the example `ioepdct` and `envar` files in the `/opt/dfsglobal/examples` directory for this release to determine if any new parameters or variables for this release are applicable to your system.
12. **Authorized program considerations**

The list of authorized programs for the Distributed File Service is:

- IOEGRWAG
- IOENEWAG
- IOESALVG
- SMBPW.

See [z/OS Program Directory](#) for a description of the PARMLIB member updates required for the member IKJTSOxx.

13. **SMB file and print considerations**

For the optional SMB support, the `_IOE_PROTOCOL_SMB` environment variable in the `/opt/dfslocal/home/dfskern/envvar` file for the `dfskern` process must be updated to specify `_IOE_PROTOCOL_SMB=ON`.

Ensure that the `LIBPATH` environment variable in the `/opt/dfslocal/home/dfskern/envvar` file is updated. If SMB print serving support is used, specify the `LIBPATH` environment variable to identify the z/OS Infoprint Server library. If the SMB support for encrypted passwords is used, update the `LIBPATH` to include the path `/usr/lib` to find the OCSF library. For example, the envvar can specify `LIBPATH=/usr/lpp/Printsrv/lib:/usr/lib`.

The `smbtab`, `dfstab`, and `devtab` files must be created if they do not exist and updated to specify the HFS data to be made available to PC clients. The `smbtab` file must be created and updated to specify the shared printers to be made available to PC clients. These files are in the `/opt/dfslocal/var/dfs` directory. See Chapter 7, “Sharing files,” on page 37 and Chapter 8, “Sharing printers,” on page 59 for more information about these topics.

### SMB configuration file considerations

Certain SMB configuration files contain system specific information. These HFS files, if copied to another system, must be modified to contain or point to data on that system. In particular, the `smbtab`, `dfstab`, and `devtab` files point to the data that is to be made available to PC clients. The `smbtab` can also refer to shared printers that are to be made available to PC clients. The `smbidmap` file and the `_IOE_MVS_DFSDFLT` `dfskern` environment variable both contain z/OS user IDs that might need to be changed. Other `dfskern` environment variables that might need to be changed include:

- `_IOE_SMB_COMPUTER_NAME`
- `_IOE_SMB_DOMAIN_NAME`
- `_IOE_SMB_IDMAP`
- `_IOE_SMB_PRIMARY_WINS`
- `_IOE_SMB_SECONDARY_WINS`
- `_IOE_SMB_WINS_PROXY`.

The `dfs_cpfiles` program is run during Distributed File Service installation. The `dfs_cpfiles` program creates (not replaces) certain customizable files with default information if the file does not exist. The files that are copied are listed in Chapter 3, “SMB post installation processing,” on page 11.
Chapter 3. SMB post installation processing

The SMB file and print server function is part of the Distributed File Service base element of z/OS. Before using the SMB support, install the z/OS release, Distributed File Service, and the other base elements of z/OS using the appropriate release documentation. During the installation of the z/OS release, you must perform actions to activate the SMB support. If you are migrating from a prior release of Distributed File Service and you want to use the SMB function, review Chapter 2, “Considerations for a new SMB release,” on page 7. For supporting installation and migration information, see http://www.ibm.com/systems/z/os/zos/installation/.

Guidelines:

- If you are using the SMB file and print server support in the Distributed File Service only (and not the DCE DFS support), DCE does not need to be configured and started. DCE only needs to be installed. Ensure that the DCE target libraries (EUV.SEUVLINK and EUV.SEUVPAL) are in the LNKLIST set and the LPALST set. For more information about how to allocate and concatenate these data sets, see APAR OA08546.
- Beginning with z/OS V1R11, you have the option of running SMB with either UID(0) or a unique nonzero UID. See “Defining RACF definitions for SMB” on page 17.
- To use the SMB File/Print support, you might also need to install and configure the Open Cryptographic Service Facility (OCSF) or the Infoprint Server Feature of z/OS.
  - If you plan to use encrypted passwords and you want to use hardware encryption, you need to install and configure the Open Cryptographic Services Facility (OCSF) base component of the Cryptographic Services element. If you plan to use password encryption and you do not want to use hardware encryption, you do not need to install and configure OCSF and you should set the dfskern environment variable _IOE_SMB_OCSF=OFF.
  - If you plan to use the SMB print serving support, you need to install and configure the Infoprint Server feature.

To use the SMB support, you must configure the support on the system using the following topics:

- “Installing and configuring SMB file and print server,” which includes configuration tasks such as:
  - Activating the SMB file/print serving function
  - Specifying optional DFS process options
  - Defining SMB users
  - Defining HFS and RFS data sets to export and directories to share for access by SMB clients
  - Defining printers to share for access by SMB clients
  - Updating SMB client PC machines running Windows or other operating systems that issue requests to file/print servers using the SMB protocol.
- “Creating the default DFS configuration files” on page 15
- “Defining SMB administrators” on page 15
- “Defining RACF definitions for SMB” on page 17

Note: During dfskern initialization, a small file (opt/dfslocal/var/dfs/rfsfile) is created (or updated) to keep track of RFS file identifiers. This file should not be changed or erased.

This chapter contains the information to assist and guide you in completing the installation and configuration of the Distributed File Service SMB file and print server support.

Installing and configuring SMB file and print server

Before you begin: You must ensure the user ID the SMB Server is running under has access to the BPX.DAEMON and BPX.SERVER RACF FACILITY class. If you are configuring SMB for the first time, or for a new release, see “Defining RACF definitions for SMB” on page 17.
To install, configure, and access the Distributed File Service server (dfskern) for SMB file and print server operation, perform the following steps:

1. Install and perform post-installation of the Distributed File Service using the applicable instructions in the z/OS Program Directory and ServerPac: Installing Your Order.

   The following list is a summary of the information in those documents:
   
   a. Ensure that the target and distribution libraries for the Distributed File Service are available.
   
   b. Run the prefix.SIOESAMP(EUVSMKID) job from UID 0 to create the symbolic links for DCE that are used by the Distributed File Service. This job reads the member prefix.SIOESAMP(EUVMKDIR) to delete and create the symbolic links.

   c. Run the prefix.SIOESAMP(IOEISMKID) job from UID 0 to create the symbolic links used by the Distributed File Service. This job reads the member prefix.SIOESAMP(IOEMKDIR) to delete and create the symbolic links.

   d. Ensure that the DDDEFS for the Distributed File Service are defined by running the prefix.SIOESAMP(IOEISDDD) job.

   e. Install the Load Library for the Distributed File Service. The Load Library (hlq.SIOELMOD) must be APF authorized and must be in link list.

   f. Install the samples (hlq.SIOESAMP).

   If you plan to use encrypted passwords (recommended) and optionally, you want to use OCSF and hardware encryption, you must ensure that the appropriate authorizations have been given to the DFS server user ID to use OCSF services. See the "Cryptographic Services OCSF Customization Considerations" section in the z/OS Program Directory and the "Configuring and Getting Started" section in the z/OS Open Cryptographic Services Facility Application Programming for information about this topic.

   If you are using ICSF, you might need to PERMIT the user ID DFS READ access to the profiles in the CSFSERV general resource class. See the z/OS Cryptographic Services ICSF Administrator's Guide for more information about the CSFSERV resource class.

   The SMB server process (DFSKERN) needs a relative dispatching priority set less than TCP/IP and UNIX System Services, but not too low. When the dispatch priority is too low, TCB and associated SRBs are not processed quickly enough, which can result in possible resource contention and hang conditions.

   The SMB server uses an Event Notification Facility (ENF) exit for event code 51 (contention). When this event occurs, an SRB is scheduled to queue a request to the SMB server. If the SMB server dispatching priority is too low, the requests can become backed up and the system can eventually run out of resources. The SMB server needs a dispatching priority that is high enough to permit these requests to be processed in a timely manner.

2. Stop the Distributed File Service server (dfskern), if it is already running, using the instructions in Chapter 4, "Managing SMB processes," on page 21.

3. Define administrators on the host system using the instructions in "Defining SMB administrators" on page 15.

4. Create the default DFS configuration files using the /opt/dfslocal/global/scripts/dfs_cpfiles shell script, if they were not created during the installation process.

   These configuration files, required by SMB file and print server, are typically created before the Distributed File Service installation is verified by the /opt/dfslocal/global/scripts/dfs_cpfiles shell script, as indicated in the z/OS Program Directory. dfs_cpfiles is described in further detail, See "Creating the default DFS configuration files" on page 15.

5. Modify the /opt/dfslocal/home/dfskern/envar file to activate SMB file and print servers by setting the environment variable (envar) _IOE_PROTOCOL_SMB=ON. Assuming that you do not want to also use DCE DFS file serving, you should also set the environment variable _IOE_PROTOCOL_RPC=OFF in the file /opt/dfslocal/home/dfskern/envar.
If you are using OCSF, ensure that the `/opt/dfslocal/home/dfskern/envar` file has a LIBPATH that adds the directory that contains the OCSF DLLs. Be sure that the directory added is the directory indicated in the `z/OS Open Cryptographic Services Facility Application Programming`

If you are using the print capability of the SMB file and print server, ensure that the Infoprint Server is installed and customized using the applicable instructions in the `z/OS Program Directory`. In addition, ensure that the `/opt/dfslocal/home/dfskern/envar` file has a LIBPATH entry that adds the directory that contains the Infoprint Server DLLs. Be sure that the directory added is the directory indicated in the "Infoprint Server Customization Considerations" section of the `z/OS Program Directory`.

For example, a LIBPATH that specifies both the OCSF DLL directory and the Infoprint Server DLL directory might be `LIBPATH=/usr/lib:/usr/lpp/Printsrv/lib`.

There is a relationship between number of threads specified for the SMB server and the maximum number of threads that z/OS UNIX permits in a process. The following DFSKERN envars have an effect on the number of threads created for the SMB server:

- `_IOE_RFS_WORKER_THREADS`
- `_IOE_SMB_CALLBACK_POOL`
- `_IOE_SMB_MAIN_POOL`
- `_IOE_TKMGLUE_SERVER_THREADS`

There are also a number of dynamically created DFSKERN threads (approximately 25). The total of the DFSKERN threads must be less than the z/OS UNIX MAXTHREADS specification in the BXPXPRMxx. If this is not the case, DFSKERN can abend during thread creation. The number of z/OS UNIX MAXTHREADS can be increased using the `SETOMVS MAXTHREADS=nn` operator command. The number of z/OS UNIX MAXTHREADS can be displayed using the D OMVS,O operator command. See `z/OS MVS System Commands` for additional information about these operator commands.

6. Because the SMB file and print server runs as an APF-authorized server, you must ensure that any DLLs that are used by the SMB file and print server are APF-authorized. This can be accomplished by using the OMVS `extattr +a` command. If you are using the Infoprint Server or OCSF, see the “Infoprint Server Customization Considerations” section in the `z/OS Program Directory` and the “Cryptographic Services OCSF Customization Considerations” section in the `z/OS Open Cryptographic Services Facility Application Programming` for information about the location of the DLLs and setting the APF-authorized extended attribute. The DFS load library is called hlq.SIOELMOD.

7. SMB clients must be able to find the server on the network in order to use the shares that the SMB server makes available. If you are using Windows, you should ensure that your computer name (specified in the `_IOE_SMB_COMPUTER_NAME` environment variable in the `/opt/dfslocal/home/dfskern/envar`) file is the same as your TCP/IP host name. See Chapter 5, “Networking considerations,” on page 31.

8. SMB communicates over several TCP/IP ports. Check your TCP/IP profile data set and verify that there are no reserves for ports 137, 138, 139, and 445. See `z/OS Communications Server: IP Configuration Reference` for information about TCP/IP configuration and reserving ports.

9. Define SMB users by modifying the `smbidmap` file identified by the `_IOE_SMB_IDMAP` environment variable in `dfskern`. Map SMB users to z/OS users on the host system using the instructions in Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33.

In addition, z/OS users that do not use DCE, should put the following line in their HFS `.profile` file in their home directory:

```
export _EUV_AUTOLOG=NO
```

Alternatively, if no z/OS users are using DCE, the above line can be placed in `etc/profile`. It is then set for all z/OS UNIX users.

10. Determine whether you intend to use passthrough authentication. See “Using passthrough authentication” on page 55 for information about passthrough authentication. Users in the domain will be authenticated using the Windows Domain Controller. Users that are not in the domain and that fail the domain authentication will additionally attempt local authentication (at the SMB server). This local
authentication will use clear or encrypted passwords based on what the Domain Controller chose
(most likely encrypted passwords) independent of the \_IOE\_SMB\_CLEAR\_PW environment variable.
In the case of encrypted passwords, those users that get authenticated locally will need to store their
SMB password in their RACF DCE segment.

11. Determine whether you intend to use password encryption. See the \_IOE\_SMB\_CLEAR\_PW
environment variable on page 154 and "Logon considerations" on page 53 for information about
password encryption. Before you enable password encryption, your PC users must store their SMB
password into their RACF DCE segment. Otherwise, they are not able to logon except possibly as a
guest user.

12. Determine whether you intend to permit guest users. Guest users are PC users that have (limited)
access to files and printers on the SMB server without identifying themselves. Guest users are
permitted when the \_IOE\_MVS\_DFSDFLT environment variable in the dfskern process is set to a
valid z/OS user ID. Guest users can access any data or files that z/OS user ID can access. If guest
users are permitted, users that specify an incorrect password or no password become the guest user ID.
It is better to deny guest users until you are certain you need this capability and that it meets your
security guidelines.

13. Determine whether you intend to use the dynamic export capability. It is controlled by the
\_IOE\_DYNAMIC\_EXPORT environment variable of dfskern. The default is OFF meaning that
dynamic export is not enabled. Dynamic export permits the SMB server to support file systems
mounted by using the z/OS Automount Facility. See z/OS UNIX System Services Planning for
information about the automount facility. Dynamic export also permits the SMB server to dynamically
"discover" mounted file systems without the need to provide dfstab and devtab entries for the file
systems. See "Dynamic export for HFS" on page 42 for information about using the dynamic export
capability of the SMB server.

14. Define shared directories if the SMB file and print server is run on the host system to export file data
sets for access by PC clients by updating the smbtab, dfstab, and devtab files and optionally, for
RFS, by specifying an rfstab file in the /opt/dfslocal/var/dfs directory. Define file systems and
filesets using the applicable instructions in Chapter 7, "Sharing files," on page 37. For RFS, the DFS
server user ID (typically DFS) must have RACF ALTER authority to the data sets that are made
available to PC users. Alternatively, you can give the DFS server user ID the OPERATIONS attribute.
If you specify a single level prefix in the devtab, you must use the OPERATIONS attribute because
you cannot create a data set profile that covers a single level prefix. (The OPERATIONS attribute can
be limited so that the DFS server user ID does not have authority to all required data sets. See z/OS
Security Server RACF Security Administrator's Guide for information about the OPERATIONS
attribute).

15. Define shared printers if the SMB file and print server is run on the host system to export Infoprint
Server printers for access by PC clients. Define the print shares by updating the file

16. SMB server performance can be significantly enhanced using the LE HEAPPOOLS(ON) parameter.
See "ioepdcl" on page 100 on how to specify HEAPPOOLS for the SMB server. See z/OS Language
Environment Programming Guide for information about HEAPPOOLS.

17. Start the Distributed File Service server (dfskern) by following the applicable instructions in
Chapter 4, "Managing SMB processes," on page 21.

18. Configure PC client workstations to access the SMB file and print server using the instructions in
Chapter 9, "Locating the SMB server," on page 61.

**Rule**

If you modify the RACF FSSEC class to activate or deactivate ACL checking, the SMB server must
be restarted. The SMB server caches permissions and does not get notified of changes to the
FSSEC class.
**Defining SMB administrators**

The SMB Administrator must have **root** authority on the z/OS system. This means a z/OS user ID with a **UID=0** or superuser authority assigned through the BPX.SUPERUSER resource in the RACF FACILITY or the equivalent for another security-based product. This is required in order to issue certain commands and to define shared directories and shared printers. For additional details on assigning superuser authority, see the topic on Using the BPX.SUPERUSER resource in the FACILITY class in *z/OS UNIX System Services Planning*.

There are two types of users who can start or stop the Distributed File Service server address space and the processes controlled by DFSCNTL, (see Chapter 4, “Managing SMB processes,” on page 21):

- A user with z/OS operator privileges.
- A user who has update privilege to the **DFSKERN.START.REQUEST** profile in the RACF FACILITY class. This profile is created during the installation of DFS. For more information about this, see *z/OS Program Directory*.

If the SMB Administrator does not use DCE facilities, the following environment variable should be specified in the SMB Administrator’s OMVS environment (for example, in the SMB Administrator’s $HOME/.profile file):

```
export _EUV_AUTOLOG=NO
```

**Creating the default DFS configuration files**

The `/opt/dfsglobal/scripts/dfs_cpfiles` program is a 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 creates files that do not exist. It does not replace an existing file to preserve any installation configuration data from a previous release.

 `/opt/dfslocal` is a symbolic link to `/etc/dfs`. Therefore all the files created by the `dfs_cpfiles` program are actually created in the `/etc` file system. See *z/OS Distributed File Service Customization* for more information about the symbolic links defined to identify the configuration files.

**Steps for using dfs_cpfiles program**

**Before you begin:** You need to have all the configuration files in the `/opt/dfslocal` subdirectories in the `/etc` file system even though some configuration files can be created in other directories and identified by `envar` specifications.

Perform the following steps to invoke `dfs_cpfiles`:

1. Log in as **root** on the local system (**UID=0** or superuser authority).
2. Enter the following command in the shell environment to invoke the `dfs_cpfiles` program:

   ```
   /opt/dfsglobal/scripts/dfs_cpfiles
   ```

   **Note:** An existing DFS configuration file is not replaced in order to ensure that any existing user customized data is not overlaid.

**Result:** `dfs_cpfiles` creates customizable files for all aspects of the Distributed File Service. A subset of these files are applicable to the SMB file and print server. The customizable files applicable to the SMB file and print server are:

- `/opt/dfslocal/etc/ioepdcf`
- `/opt/dfslocal/var/dfs/devtab`
- `/opt/dfslocal/var/dfs/dfstab`
- `/opt/dfslocal/var/dfs/rfstab`
- `/opt/dfslocal/var/dfs/smbtab`
- `/opt/dfslocal/var/dfs/hfsattr`
The `smbtab`, `dfstab`, and `devtab` files in the `/opt/dfslocal/var/dfs` directory are created by `dfs_cpfiles`. They must be updated to define shared directories. See Chapter 7, “Sharing files,” on page 37 for more information.

- The `smbtab` file must be updated to define shared printers. See Chapter 8, “Sharing printers,” on page 59 for more information.
- The `smbidmap` file must be updated to map PC user IDs to z/OS user IDs. See Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33 for more information.
- Optionally, the `hfsattr` and `rfstab` files can be updated to first define HFS data translation by file name extension attributes and next by RFS attributes for RFS files. See Chapter 7, “Sharing files,” on page 37 for more information.

The other customizable files are used for DFS client and server support in a distributed computing environment.

**Recommendation:** The files exist as created by the `dfs_cpfiles` program, otherwise it can be ignored if only the SMB file and print server is used.

Figure 1 on page 17 displays the possible output after using `dfs_cpfiles` to create the customizable configuration files for the Distributed File Service.
Figure 1 shows the **dfs_cpfiles** messages when the 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

**Tips:**

- If you are migrating to this release of the Distributed File Service from an earlier release and **dfs_cpfiles** created a new set of customizable files, you might need to add your customization data to the newly created files.

- If you are migrating to this release of z/OS from an earlier release and new customizable configuration files were not created by **dfs_cpfiles**, you might want to update preexisting customizable files with new customization options available with this release of Distributed File Service. See Chapter 2, “Considerations for a new SMB release,” on page 7 for more information about what is new in this release.

- The SMB user identity mapping file is identified by the env var `_IOE_SMB_IDMAP`. It is recommended that the `/opt/dfslocal/home/dfskern/smbidmap` file created by the **dfs_cpfiles** program should be used by the installation for user identity mapping.

When all the configuration files required by SMB file/print processing have been created and updated, you can proceed to the next step of configuration. See **z/OS Distributed File Service Customization**.

---

### Defining RACF definitions for SMB

You must define SMB clients and servers to the z/OS Security Server RACF (or equivalent security product) so they have the correct authorization to initialize and perform required system services. Access to z/OS file data is done using the authorization of the user accessing the data or the default authorization defined by your installation.

This section covers the following topics:

- [“Setting up a new SMB installation to run under a UID(0) TSO user”](#) (This is the most common set up for SMB.)
- [“Setting up new SMB installation to run under a unique nonzero UID TSO user” on page 18](#)
- [“Modifying an existing SMB installation to run with a unique nonzero UID” on page 19](#)

#### Setting up a new SMB installation to run under a UID(0) TSO user

Perform the following steps if you are setting up SMB for the first time and you want the TSO user for SMB to have UID(0) authority:

1. Define DFSGRP as a group. For example:
   ```
   ADDGROUP DFSGRP SUPGROUP(SYS1) OMVS(GID(2))
   ```
2. Create a RACF-defined UID(0) TSO user under which the SMB server will run. For example:
   ```
   ADDUSER DFS OMVS(HOME('/opt/dfslocal/home/dfscntl') UID(0)) DFLTGRP(DFSGRP) AUTHORITY(CREATE) UACC(NONE)
   ```
3. Define DFS, DFSCM, and DFSKERN as started tasks. For example:
4. Add a security facility for DFSKERN to the security program and permit the userid that runs DFS, SMB, or both server functions. For RACF, use the following commands:

   RDEFINE FACILITY DFSKERN.START.REQUEST UACC(NONE)
   PERMIT DFSKERN.START.REQUEST CLASS(FACILITY) ID(DFS) ACCESS(UPDATE)
   SETROPTS CLASSACT(FACILITY)

5. If BPX.SERVER and BPX.DAEMON FACILITY classes are not defined, define them in RACF. For example:

   RDEFINE FACILITY BPX.SERVER UACC(NONE)
   RDEFINE FACILITY BPX.DAEMON UACC(NONE)

6. Permit the TSO user to the BPX.SERVER and BPX.DAEMON RACF FACILITY class profiles. For example:

   PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(READ) ID(DFS)
   PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(DFS)
   SETROPTS RACLIST(FACILITY) REFRESH

7. If your system is set up for RACF program control protection, allow the SMB daemon control task (IOEPDCT) to run using your installation’s procedures.

**Setting up new SMB installation to run under a unique nonzero UID TSO user**

Perform the following steps if you are setting up SMB for the first time and have a requirement that the TSO user that SMB runs under is a unique nonzero UID:

1. Define DFSGRP as a group. For example:

   ADDGROUP DFSGRP SUPGROUP(SYS1) OMVS(GID(2))

2. Create a RACF-defined TSO user under which the SMB server will run. **(Restriction: Do not use DFS as the TSO user for a unique nonzero UID setup because DFS is already in use for zFS.)** This procedure uses NEWID as the example new TSO user ID:

   ADDUSER NEWID OMVS(HOME('/opt/dfslocal/home/dfscntl') UID(unique UID)) DFLTGRP(DFSGRP) AUTHORITY(CREATE) UACC(NONE)

3. If it is not already defined, define the BPX.SUPERUSER RACF FACILITY class and permit the NEWID. For example:

   SETROPTS CLASSACT(FACILITY)
   RDEFINE FACILITY BPX.SUPERUSER UACC(NONE)
   PERMIT BPX.SUPERUSER CLASS(FACILITY) ACCESS(READ) ID(NEWID)

4. Define DFS, DFSCM, and DFSKERN as started tasks. For example:

   RDEFINE STARTED DFS.** STDATA(USER(NEWID))
   RDEFINE STARTED DFSCM.** STDATA(USER(NEWID))
   RDEFINE STARTED DFSKERN.** STDATA(USER(NEWID))
   SETROPTS RACLIST(STARTED) REFRESH

5. Add a security facility for DFSKERN to the security program and permit the TSO user that runs DFS, SMB, or both server functions. For RACF, use the following commands:

   RDEFINE FACILITY DFSKERN.START.REQUEST UACC(NONE)
   PERMIT DFSKERN.START.REQUEST CLASS(FACILITY) ID(NEWID) ACCESS(UPDATE)

6. If not defined, define the BPX.SERVER and BPX.DAEMON FACILITY classes in RACF. For example:

   RDEFINE FACILITY BPX.SERVER UACC(NONE)
   RDEFINE FACILITY BPX.DAEMON UACC(NONE)

7. Permit the userid to the BPX.SERVER and BPX.DAEMON RACF FACILITY class profiles. For example:

   PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(READ) ID(NEWID)
   PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(NEWID)
   SETROPTS RACLIST(FACILITY) REFRESH
8. If your system is set up for RACF program control protection, allow the SMB daemon control task (IOEPDCT) to run using your installation's procedures.

**Modifying an existing SMB installation to run with a unique nonzero UID**

Perform the following steps if you are modifying an existing SMB setup that runs under a TSO user with UID(0) authority to run under a TSO user without UID(0) authority. In this procedure, you create a new unique nonzero UID user ID that SMB will run under, establish the user ID's relationship with the existing RACF defined group, and give the new user ID the appropriate RACF authorizations.

**Restriction:** Do not use DFS as the TSO user for a unique nonzero UID setup because DFS is already in use for zFS.

1. Create a RACF-defined TSO user under which the SMB server will run. This procedure uses NEWID as the example new TSO user ID:

   ADDUSER NEWID OMVS(HOME('/opt/dfslocal/home/dfscntl') UID(unique UID)) DFLTGRP(DFSGRP) AUTHORITY(CREATE) UACC(NONE)

2. Create BPX.SUPERUSER facility class and permit the NEWID access:

   RDEFINE FACILITY BPX.SUPERUSER UACC(NONE)
   PERMIT BPX.SUPERUSER CLASS(FACILITY) ACCESS(READ) ID(NEWID)

3. Give UPDATE access to the DFSKERN.START.REQUEST FACILITY class for the NEWID:

   PERMIT DFSKERN.START.REQUEST CLASS(FACILITY) ID(NEWID) ACCESS(UPDATE)

4. Update the current generic started task definition for the unique nonzero UID user, NEWID, that the SMB server will run under by issuing the following commands:

   RALTER STARTED DFS.** STDATA(USER(NEWID))
   RALTER STARTED DFSKERN.** STDATA(USER(NEWID))
   RALTER STARTED DFSCM.** STDATA(USER(NEWID))
   SETR RACLIST(STARTED) REFRESH

5. Give NEWID READ access to the RACF BPX.SERVER and BPX.DAEMON FACILITY class:

   RDEFINE FACILITY BPX.SERVER UACC(NONE)
   RDEFINE FACILITY BPX.DAEMON UACC(NONE)
   PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(READ) ID(NEWID)
   PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(NEWID)
   SETROPTS RACLIST(FACILITY) REFRESH

6. In some cases, you might have to modify the dfscntl and dfskern environment variable files to ensure they have the correct permissions (that is, file permissions set to 644). The files originally have owner, group, and other permissions set to 644 (owner read/write, group read, other read), but over time, the file permission can change. If the nonzero UID created in the preceding steps is not able to read the dfscntl and dfskern files, you can receive a RACF ICH408I message (or unauthorized request message from equivalent security products). Check the following files to ensure the file permissions are set correctly:

   • /opt/dfslocal/home/dfscntl/envar
   • /opt/dfslocal/home/dfskern/envar

   For example, the correct permissions display in this format:

   -rw-r--r-- 1 DFSID DFSGRP 267 Dec 3 08:56 envar

   If not, change the permissions using the chmod command:

   chmod 644 envar
Chapter 4. Managing SMB processes

The SMB daemons run as separate processes within the DFS address space as shown in Figure 2.

Figure 2. DFS server address space

If the SMB server daemons are not started as instructed, they fail with the following return codes. The IOEDFSCL and IOEPDCT daemons are shown in Table 1.

Table 1. IOEDFSCL and IOEPDCT daemon return codes

<table>
<thead>
<tr>
<th>Return code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Module is not APF authorized</td>
</tr>
<tr>
<td>8</td>
<td>Module was NOT started independently through JCL</td>
</tr>
<tr>
<td>12</td>
<td>Module is not running as a started task, but instead as EXEC PGM=xxxxxxxx started from JCL</td>
</tr>
</tbody>
</table>

The IOEDFSKN and IOEDFSXP daemons are shown in Table 2.

Table 2. IOEDFSKN and IOEDFSXP daemon return codes

<table>
<thead>
<tr>
<th>Return code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Module is not APF authorized</td>
</tr>
<tr>
<td>8</td>
<td>Module WAS started independently though JCL</td>
</tr>
</tbody>
</table>

Figure 2 shows dfsctl (the DFS Control Task) acting as the parent process to all the DFS server daemons. The DFS server daemons (dfskern and export) run as child processes of dfsctl. The export daemon communicates with dfskern to make the specified file systems available on the network and then stops. The dfskern daemon remains active to handle incoming file and print requests. dfskern can be run in the DFS Server address space or in its own address space. This is controlled by the _IOE_DAEMONS_IN_AS environment variable, which is set in the /opt/dfslocal/home/dfsctl/envar file.

If the _IOE_DAEMONS_IN_AS environment variable is not specified, dfskern runs in the DFS Server address space.

If it is specified as _IOE_DAEMONS_IN_AS=DFSKERN, dfskern runs in its own address space (called the dfskern address space) as shown in Figure 3 on page 22.
Running `dfskern` in its own address space might reduce contention for resources and provide better failure recovery. IBM recommends that `dfskern` be run in its own address space. MODIFY commands are unchanged whether `dfskern` runs in its own address space or not. Ensure that the `dfskern` JCL is available and the `daemonct` envar file is in the `daemonct` home directory (`/opt/dfslocal/home/daemonct`) if you want to run `dfskern` in its own address space. Then, if you want to change where `dfskern` runs, add or remove the `dfscntl` environment variable (in the `/opt/dfslocal/home/dfscntl/envar` file), `_IOE_DAEMONS_IN_AS=DFSKERN`, stop DFS if it is running, and then restart DFS.

![Diagram of DFSKERN Address Space and DFS Server Address Space](image)

After the DFS server address space is configured, both daemons (`dfskern` and `export`) are started automatically when the DFS server address space is started.

All requests to DFS are directed to `dfscntl`, that performs the requested action. The DFS server daemons can be started and stopped using an operator command. In starting and stopping the daemons, `dfscntl` uses a Daemon Configuration File (that is, the `ioepdcf` file) that contains information about the daemons that were previously configured on the host. The Daemon Configuration File contains runtime options, startup parameters, and restart information for its subprocesses. For details on the Daemon Configuration File, see “Daemon configuration file” on page 26.

Besides starting and stopping the DFS server daemons, `dfscntl` can also detect a daemon that has prematurely stopped and tries to restart it automatically. The algorithm used by `dfscntl` in starting and restarting the DFS server daemons is summarized in “How `dfscntl` starts the DFS server daemons” on page 27.

The following are the PDS member names for the DFS processes started by `dfscntl`:

- **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 `dfscntl` and executes the load library entry, `IOEDFSXP`. The `export` process has no alias.

**Who can start and stop DFS server daemons?**

There are two types of users who can start and stop the DFS server daemons in DFS:

- A user with z/OS operator privileges.
A user who has update privilege to the `DFSKERN.START.REQUEST` profile in the RACF FACILITY class. This profile is created during the installation of DFS. For information about this, see [z/OS Program Directory](#).

### Starting DFS server daemons

DFS server daemons are started in any of the following ways:

- During the system IPL
- Using the `MODIFY DFS` operator command
- Using the `START` operator command.

Based upon the control files for `dfscntl` set up by the DFS administrator, all of the DFS server daemons can be automatically started when the DFS started task is started.

The DFS daemons can all run in one address space (except for possibly `dfskern`) which is a started task (default name DFS). A user with z/OS operator privileges can start and stop the DFS started task. The DFS server address space can be started automatically during system IPL. To start the DFS server address space, use the z/OS `START` command. For example,

```plaintext
start dfs
```

Ideally, the daemons run continuously in the background and do not need to be started or stopped again. However, you might start the DFS server daemons manually in certain situations, for example, if a daemon ends abnormally. You can use the `MODIFY` operator command to manually start or stop the DFS server daemons.

Each of these alternatives is discussed briefly in the following sections. For complete details, see [Chapter 12, "z/OS system commands," on page 77.](#)

### The MODIFY DFS operator command

The DFS server daemons (processes) can be started or stopped using the `MODIFY DFS` operator command. Using `MODIFY DFS`, you can also view the status of the DFS server daemons. Following is the syntax of the `MODIFY DFS` command:

```plaintext
MODIFY DFS,command daemon[,options]
```

**where:**

- **DFS** is the name of the DFS server address space.
- **command** is the action that is to be performed on the SMB server daemon or daemons. It can have any of the following values:
  - `start` Starts the DFS server daemon or daemons.
  - `stop` Stops the DFS server daemon or daemons.
  - `query` Displays the state of the DFS server daemon or daemons.
  - `send` Sends requests to the DFS server daemon or daemons.
- **daemon** is the name of the DFS server daemon for which the action is being requested. It can have any of the following values:
  - `dfskern` DFS kernel program (includes the SMB file and print server).
  - `export` Export program to make file systems available for exporting and shares available to PC users.
  - `unexport` Unexport program to unexport file systems and delete shares.
  - `all` All the DFS server daemons (that is, `dfskern` and `export`).

---

Chapter 4. Managing SMB processes 23
options Values that are passed to the daemons.

Using MODIFY DFS to start DFS server daemons
With the MODIFY DFS operator command, you have the option of starting an individual daemon or starting all the daemons using a single command.

For example, to start the dfskern daemon, enter the following:
modify dfs,start dfskern

To start all the daemons, enter the following:
modify dfs,start all

Note: Do not use the MODIFY command to start the DFS server daemons while the DFS server address space is still initializing. During initialization, DFS attempts to start all the DFS server daemons that have been configured on the z/OS host. If you issue the MODIFY command while DFS is initializing, the DFS server daemons can start out of order or stopped erroneously. This can lead to unexpected errors during initialization and cause DFS to end abnormally.

It is recommended that you wait until DFS has issued a log message indicating that DFS server initialization has completed before using the MODIFY commands. For complete details on MODIFY, see Chapter 12, “z/OS system commands,” on page 77.

Order of starting DFS server daemons
When DFS server daemons are started manually, the successful startup of some daemons depend on the availability of the services provided by other daemons. This implies that the DFS server daemons must be started in a particular order.

Following is the sequence by which DFS server daemons should be started.

Note: This is only applicable if you need to start any of the DFS server daemons individually. If the DFS server daemons are started collectively, (for example, using the start all option of the MODIFY DFS command) DFS ensures that the correct starting sequence is followed.

1. dfskern
2. export

For example, to successfully start the export daemon, the dfskern daemon must already be up and running.

If you are using the DFS server's SMB capability to do printing, the Infoprint Server should be started before the dfskern server daemon. Otherwise, you need to issue the dfsshare command to share the printers defined in smbtab.

Stopping DFS
DFS server daemons are stopped in any of the following ways:
- Using the STOP operator command.
- Using the MODIFY DFS operator command

To stop the DFS server address space, use the STOP operator command to ensure the normal shutdown of the address space.

To stop the DFS server address space and all DFS server daemons, enter the following z/OS command:
stop dfs
To stop the DFS server daemons, but not the DFS server 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.

### Using MODIFY DFS to stop DFS server daemons

With the **MODIFY DFS** operator command, you have the option of stopping an individual daemon or stopping all the daemons that are configured on the host using a single command.

For example, to stop the **dfskern** daemon, enter:

```
modify dfs,stop dfskern
```

To stop all DFS server daemons on the host, enter:

```
modify dfs,stop all
```

### Viewing the status of DFS server daemons

You can query the status of the DFS server daemons using the **query** option of the **MODIFY** system command. You do not need the special privileges of a DFS administrator or an operator to use the **QUERY** option.

For example, to query the status of the **dfskern** daemon, enter the following:

```
modify dfs,query dfskern
```

A message about the status of the daemon is written on the system log. This message also contains the **process ID** of the daemon.

The status of the daemon can be any of the following:

- **READY**
  
  Indicates that the daemon is running, has been initialized, and is ready to receive and process incoming requests.

- **ACTIVE**
  
  Indicates that a manual process is running. When an active process stops, it is never considered an error and it is never restarted automatically.

- **INITIALIZING**
  
  Indicates that the daemon has been started, but is not yet ready to receive and process incoming requests.

- **STOPPING**
  
  Indicates that a request to stop the daemon has been received and that the daemon is in the process of stopping.

- **DOWN**
  
  Indicates that the daemon is not active.

- **UNKNOWN**
  
  Indicates that the status of the daemon cannot be determined. This can occur if the daemon was started, but no response was received by the system indicating a change in its status.

  **Note:** You can issue a command to stop a daemon if it is in the **UNKNOWN** state.

The status of DFS daemons controlled by **dfscntl** can be queried by z/OS operator commands. For example:

```
modify dfs,query all
modify dfs,query dfskern
```

Following is an example of a query command to **dfscntl**. The output is sent to the z/OS console:

```
modify dfs,query dfskern
```
Starting DFS server daemons during IPL

Because the DFS server daemons are contained in the DFS server address space (except for possibly dfskern), these daemons are started during the initialization of DFS. This gives you the ability to configure the host to automatically start the DFS server address space during the system IPL.

dfscntl uses the Daemon Configuration File to determine which daemons can be started, and the parameters to pass to the daemon load module when starting the daemon. The DFS server address space can be started automatically during system IPL. To start the DFS server address space, use the z/OS START command. For example:

\texttt{start dfs}

Daemon configuration file

The Daemon Configuration File is used by dfscntl to obtain necessary information when starting the DFS server daemons. The Daemon Configuration File contains the following information:

- The name of the process to be started.
- A parameter that specifies actions to be taken when the process is started. It can have the following values:
  - \texttt{Y}: Start the process during initialization. If the process ends abnormally, then restart it automatically.
  - \texttt{N}: Do not start the process automatically or manually.
  - \texttt{I}: Start the process during initialization. The process can be started manually. If the process ends, it does not restart.
  - \texttt{M}: Can be started manually.
- Parameters that are passed to the load module when a daemon is started, called the argument list (including LE/370 runtime options).
- The Minimum Restart Interval. dfscntl attempts to restart a daemon that ends abnormally only if the daemon was running for at least this time interval. If a daemon ends during this time interval, it is not be restarted.
- The Time-out Period, which is the maximum time interval that dfscntl waits for the daemon to complete its initialization after it has been started. When this time interval elapses, and dfscntl has not received confirmation from the daemon that initialization has completed, the status of the daemon is set to unknown.

The path name to the Daemon Configuration file is /opt/dfslocal/etc/ioepdcf. Figure 4 on page 27 shows the typical contents of the Daemon Configuration file.
In the ARG (argument list) field of this example, ENVAR indicates the Language Environment® (LE) environment variables used, in this case, the _EUV_HOME environment variable is specified to identify the daemon’s home directory. (The home directory contains the daemon’s envar file. See “envar” on page 97 for more information.) Anything after the “/” character are program parameters. The “>” character is the redirection character which indicates that the output is redirected to the DD name that follows.

Important Note to Users:
Under typical circumstances, you do not need to edit the Daemon Configuration File. Although there might be certain situations when you modify the Daemon Configuration File, it is recommended that it is under the supervision of an IBM service representative.

The Daemon Configuration File can only be modified when the DFS server address space is not running.

The Daemon Configuration File is optional. When it is omitted, the defaults are as listed in Figure 4 except that BOSERVER CONFIGURED=M. When using only the SMB capability of the Distributed File Service (and not the DCE DFS capability), there is no need to start or reference the BOSERVER or the BUTC0n servers.

How dfscntl starts the DFS server daemons
When a request arrives to start DFS server daemons, dfscntl looks at the Daemon Configuration File to see if the particular daemon or daemons are configured on the host. If the daemon is configured and is not running, dfscntl starts it and waits for the daemon to initialize successfully.

In case of the abnormal ending of any DFS server daemon, dfscntl tries to restart the daemon. dfscntl attempts to restart the daemon only if the daemon was running for at least the duration of the Minimum Restart Interval, as specified in the Daemon Configuration File. If the daemon ended within this time interval, it is not restarted.

DFSKERN CONFIGURED=Y LMD=DFSKERN ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/dfskern')/-admingroup subsys/dce/dfs-admin>DD:DFSKERN2>&1" RESTART=300 TIMEOUT=300
EXPORT CONFIGURED=I LMD=IOEDFSXP ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/dfsexport')/-all -verbose >DD:EXPORT 2>&1" RESTART=300 TIMEOUT=300
UNEXPORT CONFIGURED=M LMD=IOEDFSXP ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/dfsexport')/-detach -all -ver >DD:UNEXPORT2>&1" RESTART=300 TIMEOUT=300
BOSERVER CONFIGURED=N LMD=BOSERVER ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/boserver')/ >DD:BOSERVER 2>&1" RESTART=300 TIMEOUT=300
BUTC01 CONFIGURED=M LMD=BUTC01 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc01')/ -tcid >DD:BUTC01 2>&1" RESTART=300 TIMEOUT=300
BUTC02 CONFIGURED=M LMD=BUTC02 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc02')/ -tcid 1 >DD:BUTC02 2>&1" RESTART=300 TIMEOUT=300
BUTC03 CONFIGURED=M LMD=BUTC03 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc03')/ -tcid 2 >DD:BUTC03 2>&1" RESTART=300 TIMEOUT=300
BUTC04 CONFIGURED=M LMD=BUTC04 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc04')/ -tcid 3 >DD:BUTC04 2>&1" RESTART=300 TIMEOUT=300
BUTC05 CONFIGURED=M LMD=BUTC05 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc05')/ -tcid 4 >DD:BUTC05 2>&1" RESTART=300 TIMEOUT=300
BUTC06 CONFIGURED=M LMD=BUTC06 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc06')/ -tcid 5 >DD:BUTC06 2>&1" RESTART=300 TIMEOUT=300
BUTC07 CONFIGURED=M LMD=BUTC07 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc07')/ -tcid 6 >DD:BUTC07 2>&1" RESTART=300 TIMEOUT=300
BUTC08 CONFIGURED=M LMD=BUTC08 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc08')/ -tcid 7 >DD:BUTC08 2>&1" RESTART=300 TIMEOUT=300

Figure 4. Daemon configuration file

In the ARG (argument list) field of this example, ENVAR indicates the Language Environment® (LE) environment variables used, in this case, the _EUV_HOME environment variable is specified to identify the daemon’s home directory. (The home directory contains the daemon’s envar file. See “envar” on page 97 for more information.) Anything after the “/” character are program parameters. The “>” character is the redirection character which indicates that the output is redirected to the DD name that follows.
When dfscnt1 restarts an abnormally terminated daemon, it does not correct the problem that caused the daemon to end unexpectedly. Thus, depending on the cause of the abnormal ending, the daemon can fail again after restarting because of the same error condition.

Using the -nodfs option to start dfscnt1
You can start the DFS server address space without starting the configured DFS server daemons by using the -nodfs option of the START operator command, for example:

```
start dfs,parm='-nodfs'
```

Changing environment variables
Each SMB server process uses environment variables to control how it behaves. When any processes environment variables are changed in the envar file in the home directory of the process, the process must be restarted to make them take effect.

Changing mappings
The dfskern process optionally supports a mapping between SMB user IDs and z/OS user IDs. The smbidmap file is located by means of the _IOE_SMB_IDMAP environment variable of dfskern. If the smbidmap file is updated (to add, change, or delete mappings), you must issue the modify dfs,send dfskern,reload,smbmap operator command to make them take effect. If you change the location of the smbidmap file, then the _IOE_SMB_IDMAP environment variable must be updated and the dfskern process must be restarted. It is recommended that the smbidmap file is located in the /opt/dfs/local/var/dfs directory so that it is contained with the other customizable files.

Changing shared directories or shared printers
Shared directories and shared printers are defined in the smbtab file. If the smbtab file is changed to add or delete a share, the dfsshare command must be issued to make it take effect. Use the -share parameter of the dfsshare command to add a new share defined in smbtab. Use the -detach parameter of the dfsshare command to delete a share. The dfsshare -detach for the share must be issued before removing the share from the smbtab.

If the shared directory is contained in a file system that has not been exported before, or if there are additional file systems (below the file system containing the shared directory) that you want available to PC users, then you should add those file systems to the dfstab and the devtab and then issue the dfsexport command before issuing the dfsshare command.

If, however, you are using the dynamic export capability of the SMB server, dfstab and devtab entries are not required for file systems that are mounted below the file system containing the shared directory. See "Dynamic export for HFS" on page 42 for information about the dynamic export capability.

Shared printers do not need entries in the dfstab nor the devtab and do not require the dfsexport command.

Changing the hfsattr or the rfstab
The hfsattr contains directives that map a file name extension (suffix) to an indication whether the data should be translated from ASCII to EBCDIC and vice versa. Its location is specified in _IOE_HFS_ATTRIBUTES_FILE environment variable of the dfskern process. The rfstab contains creation (and other) attributes for RFS files. Its location is globally specified in the _IOE_RFS_ATTRIBUTES_FILE environment variable of the dfskern process. It can also be specified on an RFS file system basis in the devtab attrfile parameter. When the hfsattr or the global rfstab file is modified, the dfskern process must be restarted to make it take effect. If an RFS file system's rfstab is
modified, the file system must be re-exported. That is, if it is already exported, it must be unexported and then exported. This is accomplished with the \texttt{dfsexport} command.

\section*{Changing the Infoprint Server DLL}

If a new release of the Infoprint Server is installed or it is enabled, it can be activated for the SMB server by the \texttt{modify dfs,send dfskern,reload,print} system command. You also need to issue the \texttt{dfsshare} command to share the printers defined in \texttt{smbtab}. 
Chapter 5. Networking considerations

In order to use the shares that the SMB server makes available, PC clients must be able to find the server on the network. The communications mechanism used is TCP/IP, and the methods of server discovery follow:

- The Domain Name Service (DNS)
- The Windows Internet Naming Service (WINS)
- Clients on the same workgroup and same subnet as the server
- The LMHOSTS file.

The order in which a server name is resolved to a TCP/IP address may vary depending on the client software and the service pack level of the Windows client software.

TCP/IP networks can use the Domain Name Service (DNS) to map server system names to IP addresses. In a DNS network, an entry tells clients in the network how to map the name of the server to its proper TCP/IP address.

If you want PC clients to access the SMB server by using DNS, then you must ensure that the hostname and IP address are added to the DNS database. Using DNS is generally the easiest way for clients to access the SMB server on a distributed network. In this case, you should ensure that the (TCP/IP) hostname and the (SMB) computer name are the same. (This is the default if you do not specify a computer name for the SMB server by using the __IOE_SMB_COMPUTER_NAME environment variable of \system.) Also, if you have Windows XP, you should ensure that the SMB computer name is the same as the TCP/IP hostname.

The supported Windows SMB clients make SMB calls directly over TCP or through NETBIOS over TCP/IP. When connecting directly over TCP, the client makes calls using a TCP/IP connection to server port 445. When using NETBIOS over TCP/IP, the client makes calls to ports 137, 138 and 139. When connecting to the server, these clients will attempt to connect to ports 445 and 139 to establish a session. The __IOE_SMB_TRANSPORTS environment variable specifies the enable modes as NETBIOS (port 139), DIRECT (port 445), or BOTH (ports 445 and 139). The default is NETBIOS. Many SMB clients have configuration options to enable NETBIOS over TCP/IP, SMB directly over TCP/IP, or both. The server responds on the enabled ports; the client software can choose to attempt one protocol prior to the other or both in parallel.

Microsoft Windows servers can provide the Windows Internet Naming Service (WINS) which allows clients to map a computer name to the computer’s actual TCP/IP address. If you use a WINS server in your network, you can configure the SMB server to announce itself to the WINS server. Then you can configure PC clients to connect to the SMB server by using the WINS server. The SMB server announces itself to the primary WINS server (identified by the __IOE_SMB_PRIMARY_WINS environment variable of \system). If the primary WINS server cannot be contacted, the SMB server announces itself to the secondary WINS server (identified by the __IOE_SMB_SECONDARY_WINS environment variable of \system). The SMB server does not, itself, act as a WINS server. It can, however, act as a WINS proxy. That is, it can accept WINS requests from PC clients and forward them to a WINS server if the __IOE_SMB_WINS_PROXY environment variable of \system is specified as ON. The PC clients would have the IP address of the SMB server specified as the WINS Server IP address.

In the __IOE_SMB_DOMAIN_NAME environment variable of \system, you can specify the name of the domain or workgroup that the SMB server should be a member of. This can be the name of an existing domain or workgroup in your LAN environment. If possible, put your SMB server in the same domain or workgroup as your client PCs.

- An SMB server that is in the same workgroup and the same subnet as the PC clients appear in the network directory without any additional configuration on the server or those PC clients. An SMB server that is on the same subnet as a Primary Domain Controller that is acting as a WINS server appears in the
Network directory of PCs that contain the WINS server IP address. The SMB server announces itself to
the Browser by using subnet broadcast on User Datagram Protocol (UDP) port 138. It does this at every
Browser announcement interval (specified by the _IOE_SMB_BROWSE_INTERVAL environment variable
of dfskern). Those PC clients can also find the SMB server by using subnet broadcast to UDP port 137.
The SMB server responds to this broadcast. PC clients that want to use the My Network Places or
Network and Sharing Center function should have the NetBEUI protocol installed.

PC client operating systems can provide static configuration files that can map server system names to
TCP/IP addresses. These files are typically more difficult to manage than a solution that involves more
centralized control (for example, a DNS or WINS server). This is because the network administrator must
configure each PC client individually. Static configuration files are very useful, however, in large, distributed
networks. In this environment, clients and servers exist in different subnets (network segments) and
possibly different workgroups (domains). Static configuration files help clients locate servers.

**Note:** To enable the My Network Places or Network and Sharing Center directory function, there must be
at least one Windows XP client, Windows Vista client, or Windows 2003 domain controller on the
same subnet as the SMB server. The Windows XP, Vista, or Windows 2003 domain controller must
be acting as a Domain Master Browser (as it usually does). Refer to [SMB server does not show up in My Network Places](https://www.ibm.com/support/knowledgecenter/en/SSSH72_1.13.0/com.ibm.zos.v1r11.doc/smbadmn/smbadm_smbserverdoesnotshowup.html) on page 166 for more information on My Network Places.

Windows clients provide the LMHOSTS file that can map server computer names to IP addresses.
LMHOSTS contains IP addresses and server computer names for which to map those addresses. You can
use these files to map the IP address of the SMB server for clients. This allows clients to find the SMB
server in a large, distributed network environment.

You can find more information about LMHOSTS files in the sample LMHOSTS file that is provided with
your Windows operating system. Additional information is available in your PC operating system
documentation.
Chapter 6. Mapping SMB user IDs to z/OS user IDs

The local security subsystem (for example, RACF) determines if the client is authorized to access HFS or RFS directories and files. Because the local security subsystem does not recognize SMB user IDs, the SMB user ID that comes to the SMB server must be mapped to a local user ID. This mapping is defined in the `smbidmap` file, which is read during `dfskern` initialization or as a result of the `modify dfs,send dfskern,reload,smbmap` operator command. The `smbidmap` file is an HFS file, a sequential data set or a member of a PDS, and its location is specified in the `_IOE_SMB_IDMAP` environment variable of `dfskern`. It contains SMB user IDs and their corresponding z/OS user IDs. This information is used by `dfskern` to map SMB user IDs to z/OS user IDs. The following procedure can be used to map SMB user IDs to z/OS user IDs:

1. Create an `smbidmap` file
2. Set the `_IOE_SMB_IDMAP` environment variable
3. Stop and restart the `dfskern` process.

Each of these procedures are described in the following sections.

**Note:** If `_IOE_SMB_IDMAP` environment variable already has the name of the `smbidmap` file and the `smbidmap` is just being updated, the `modify dfs,send dfskern,reload,smbmap` command can be used to activate the updated mappings.

**Restriction:** To limit the number of calls to the security manager, the default behavior of the SMB server is to cache user credential information, including the Security Server RACF group information, when it is started. The cached information is not refreshed. To ensure the SMB server recognizes user credentials, you must stop and restart the SMB server. To change the default behavior to dynamically recognize changes to user credential information, set the `_IOE_USE_PTHREAD_SECURITY` environment variable to ON.

**Creating an smbidmap file**

The `smbidmap` file is a text file that the administrator creates and maintains. It must be an HFS file. Any editor available on z/OS UNIX can be used (for example: oedit, vi). The `smbidmap` file contains one or more mapping declarations and has the following general format:

```
SMB-user-ID1
z/OS-user-ID1

SMB-user-ID2
z/OS-user-ID2
...
```

Each entry has two elements: SMB user ID and z/OS user ID. A blank line is optional between entries (but is recommended for readability). The following list explains each element in an SMB user ID mapping entry:

- **SMB-user-ID** or **Domain/SMB-user-ID** or **Workgroup/SMB-user-ID** or **Domain/***
  - Specifies the client's SMB identity. This can either be a simple SMB user ID (when the domain of the SMB user ID is unimportant) or a fully qualified name (for clients within and outside the domain/workgroup) or a domain name with an asterisk (for all clients in a particular domain). The SMB user ID can be up to 20 characters in length. A Domain (Workgroup) name can be up to 15 characters in length.
  - `SMB-user-ID` is assumed to be in any domain.
  - `Domain/SMB-user-ID` is assumed to be in the specified domain.
  - `Workgroup/SMB-user-ID` is assumed to be in the specified workgroup.

**Note:** When a PC user is in a workgroup, the PC client sends the computer name as the domain name (not the workgroup name).
**z/OS-user-ID**

Specifies the z/OS user ID of the client. All potential SMB clients must have z/OS user IDs on the system where the SMB server is running.

**Note:** This field is case sensitive. The case of the z/OS user ID is important when using the &USERID keyword of the smbtab file. When a PC user does a net use to a shared directory that has &USERID in the directory path name field of the smbtab entry, the z/OS user ID is taken from the smbmap entry for that PC user and is used to reference (with the exact case specified for the z/OS user ID) the directory. When this directory reference causes automount to occur, a directory with this exact case is created by automount. Make sure you specify the z/OS user ID with the correct case so that the directory created by automount has the correct name. (For example, it might need to match the user ID in the definition of the user’s home directory in the RACF OMVS segment of the user’s profile.) The z/OS user ID will be folded to upper case when used to logon to z/OS.

For information about other entries in smbmap, see [“smbmap” on page 104](#).

**Note:** The SMB user ID will be used (as received over the wire) for the z/OS user ID for an &USERID value specified in the smbtab. In this case, the PC user ID as specified on the Windows logon (or on the net use command) will be used by automount to create a directory. This is case sensitive in that the directory will be created with the exact case as received from the PC. Some clients fold the PC user ID to upper case before sending it to the SMB server. In this case, if the directory created by automount needs to be lower case or mixed case, the administrator needs to add an entry to the smbmap file to map the upper case PC user ID to the correct case z/OS user ID.

Each SMB user can only have one mapping to a z/OS user. Conversely, you can map different SMB users to the same z/OS user.

### Setting the _IOE_SMB_IDMAP environment variable

The _IOE_SMB_IDMAP environment variable must be set to the name of the smbmap file used by the SMB 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 example, if the HFS path name of the smbmap file is /opt/dfslocal/home/dfskern/smbmap, this variable is set by the following entry in the envar file:

```plaintext
_IOE_SMB_IDMAP=/opt/dfslocal/home/dfskern/smbmap
```

If the smbmap file is contained in data set HLQ.DFSKERN, member name SMBIDMAP, this variable is set by the following entry in the envar file:

```plaintext
_IOE_SMB_IDMAP="/HLQ.DFSKERN(SMBIDMAP)"
```

### Modifying and deleting identity mapping entries

Edit the smbmap file to modify or delete mapping entries. For these changes to take effect, you must either restart the SMB server or reload the smbmap file by using the modify dfs,send dfskern,reload,smbmap command. See Chapter 12, “z/OS system commands,” on page 77 for more information about the modify command.

### Determining the z/OS user ID from the SMB user ID

The following decisions are made for how dfskern determines the z/OS user ID from the SMB user ID:

- The smbmap table is read during SMB server initialization or when you enter the modify dfs,send dfskern,reload,smbmap command.
- When an SMB comes to the SMB server,
If an SMB user ID and domain are in the SMB, then search for a match in `smbidmap` in a case insensitive manner.

If no match, use just the SMB user ID from the SMB and search for a match in the SMB user ID and "don't care" domain in a case insensitive manner. (These are entries in `smbidmap` that do not have a domain.)

If there is still no match, see if there is an *= entry. If yes, use the SMB user ID from the SMB as a z/OS user ID.

If a match was found, attempt a logon with the mapped ID and the password.

If there is still no z/OS user ID, or if the logon attempt was unsuccessful, see if the `_IOE_MVS_DFSDFLT` environment variable is specified. If yes, use its value as the z/OS user ID (without a password). This is sometimes known as a guest login.

The SMB client request is denied if the SMB server cannot determine a mapping to a z/OS user ID. For example, if the SMB user ID is unspecified 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 client request is denied.

How the SMB user ID is determined

The SMB user ID is determined from the user ID the user specifies when logging on to Windows. This user ID is mapped to a z/OS user ID, and the password is taken as the password for the z/OS user ID (when using clear passwords) or the user's SMB password in their RACF DCE segment (when using encrypted passwords). See "Logon considerations" on page 53 for information about clear passwords and encrypted passwords.

The simplest method for using the SMB server is to logon to Windows with your SMB user ID that the SMB server can map to a z/OS user ID. When a drive letter is mapped to a shared directory, that user ID is sent to the SMB server. If you are prompted to enter your password, you should enter your z/OS password (when using clear passwords) or your SMB password from your RACF DCE segment (when using encrypted passwords).

The password that you logged onto Windows with is sent to the SMB server. If your z/OS password (when using clear passwords) or your SMB password from your RACF DCE segment (when using encrypted passwords) is different than your Windows password, you should specify your z/OS or SMB password on the `net use` command. If your Windows password is different than your z/OS or SMB password and you do not specify it on the `net use` command (or you specify it incorrectly), you are logged in as the DFSDFLT user ID, if the `_IOE_MVS_DFSDFLT` environment variable is specified on the SMB server. If the `_IOE_MVS_DFSDFLT` environment variable is not specified and you specified an incorrect password, you might be prompted for the correct password or denied.

On Windows clients, you can specify your SMB user ID on the map network drive pull down and you can specify your SMB user ID (and password) on the `net use` command. This gives you the ability to use a different SMB user ID than the one used to logon to Windows.
Chapter 7. Sharing files

Before PCs can access files, a shared directory must be created. A shared directory represents the starting point or top directory of a “tree” of directories and files. A PC user can access the shared directory and the subdirectories and files based on the user’s authorization to those items. A PC user cannot reference a directory (or file) that is higher than the shared directory (except by accessing an absolute symbolic link when the _IOE_SMB_ABS_SYMLINK environment variable in the dfskern process is ON).

Before a shared directory can be created, the file system containing the directory to be shared must be exported. (A file system is identified by its file system name.) The following discussion assumes that you are familiar with HFS file systems, z/OS UNIX concepts, and data sets. The file system types supported by the SMB server are HFS, zFS, TFS, and AUTOMNT. If you are in a sysplex with Shared HFS files, SMB support of zFS is limited to zFS compatibility mode file systems. The NFS and DFSC file system types are not supported. Unless otherwise noted, when the term HFS is used, it includes all the supported file system types.

**Note:** In order to export an HFS file system, that file system must be owned by the system that the SMB server is running on. When export is attempted on a (sysplex) shared file system that is owned by a system other than the one that the SMB server is running on, the SMB server attempts to move the ownership of a shared HFS file system to the system that the SMB server is running on (when the _IOE_MOVE_SHARED_FILESYSTEM environment variable is ON in the dfskern process). If this is unsuccessful (when, for example, the file system is being exported by another SMB server on the other system), the SMB server is unable to export that file system and it is not available to PC clients. When the SMB server is running with dynamic export enabled and the move of ownership of file systems enabled, you should run a single SMB server on a sysplex. Otherwise, each SMB server might try to export (and move the ownership of) the same file system and one of them will fail.

The z/OS SMB server cannot export any zFS file systems on z/OS V1R11 that are sysplex-aware regardless of owner of the file system. If you attempt this, you will receive a message similar to the following:

```
IOEX18I09A Dfsexport: Unable to attach /dev/ufs105:hfs105
(EDCS1391 Operation not permitted. (errno2=0xC00B05B1)).
Ignoring request.
```

If you want to export zFS file systems in a shared file system environment in z/OS V1R11, zFS must be started with sysplex=off specified in the IOEFSPRM configuration file. You can check whether zFS is running sysplex-aware by using the zsadm configquery -sysplex_state command. If it returns 2, it is running sysplex-aware.

---

**Sharing HFS files**

This section discusses the HFS file system.

**Exporting and sharing HFS file systems**

An HFS file system must be exported before a directory contained in it can be shared. Exporting is done on an HFS file system basis and tells z/OS UNIX that a file system is being made available to clients by a File Exporter (that is, the Distributed File Services SMB File/Print Server).

PC clients access data on HFS through sharing that is done on a directory tree basis. For a directory to be shared, the file system that the directory is contained in must be exported. Suppose we had the following (purposely simplified) entire file hierarchy on a z/OS system:
This file hierarchy is made up of separate file systems mounted together as in the following representation:

The ovals represent the individual file systems. There are five file systems in this hierarchy. Each of these file systems has a data set name. The top file system is referred to as the root file system. Its data set name is OMVS.ROOT. Each of the lower file systems are mounted on top of a directory in the file system above it. There are three file systems mounted on three directories that are contained in the root file system. One is mounted on the directory /bin (data set OMVS.BIN), another is mounted on /abc (data set OMVS.ABC), and the third is mounted on /u (data set OMVS.U). The fifth file system is mounted on /abc/def (data set OMVS.ABC.DEF). The data set name of a file system can be displayed with the df command. The following diagram examines the /abc/def/ghi path a little more closely.
The `/abc/def/ghi` path consists of three file systems mounted as follows:

If you want to create a share for directory `/abc/def/ghi`, you must export the file system where the `/ghi` directory resides (OMVS.ABC.DEF).
In addition, if there are other file systems below the file system containing the shared directory, you might want to also export these so that PC clients can reference data in those file systems. They should be added to dfstab and devtab and exported using the dfsexport command (refer to "smbtab, dfstab, and devtab entries for HFS" for more information). Otherwise, a PC client is denied access to directories and files in file systems that are not exported.

smbtab, dfstab, and devtab entries for HFS

The files that the SMB File Server uses to relate the shared directory and its exported file system are the smbtab, the dfstab, and the devtab. (They are all located in /opt/dfslocal/var/dfs.) A common minor device number is used to tie related entries in these three files together. The smbtab is used to define the shared directory. The dfstab and the devtab are used to define the file system to be exported.

In smbtab, the minor device number is specified using the format /dev/ufs<n>, where n is a locally assigned unique minor device number that must refer to the HFS file system data set where the root of the shared directory path name resides (that is the file system where the first / resides). This should always be the file system where the directory that you want to share resides. Note that when you are sharing a mount point, you would specify the file system that is mounted on the mount point in devtab and / as the directory in smbtab.

In dfstab, the minor device number is specified using the format /dev/ufs<n> to identify the assigned unique identifiers for the HFS file system.

In devtab, the minor device number is specified using the format define_ufs<n> to identify the data set name for the HFS file system.

For example, if the shared directory is /ghi, then the smbtab, dfstab, and devtab entries might be:

**smbtab:**
/dev/ufs2 myshare ufs "My share description" r/w 100 /ghi

**dfstab:**
/dev/ufs2 hfs2 ufs 101 0,,1715

**devtab:**
define_ufs 2
OMVS.ABC.DEF

Notice that you did not need to export the file systems that are above the OMVS.ABC.DEF file system. That is, you did not need to create dfstab and devtab entries for OMVS.ROOT and OMVS.ABC.

If, however, there were additional file systems below the shared directory that you want to export, specify them in the dfstab and devtab. PC users might want to access these directories and files. Alternatively, you can use the dynamic export capability. Refer to "Dynamic export for HFS" on page 42.

As another example, if you wanted to share the entire file hierarchy with PC users from the root on down, then the smbtab, dfstab, and devtab entries might be:

**smbtab:**
/dev/ufs4 hfsroot ufs "My share description" r/w 100 /

**dfstab:**
/dev/ufs2 hfs2 ufs 101 0,,1715
/dev/ufs3 hfs3 ufs 102 0,,1718
/dev/ufs4 hfs4 ufs 103 0,,1721
/dev/ufs5 hfs5 ufs 104 0,,1724
/dev/ufs6 hfs6 ufs 105 0,,1727

**devtab:**
define_ufs 2
OMVS.ABC.DEF
define_ufs 3
OMVS.ABC
define_ufs 4
OMVS.ROOT
define_ufs 5
OMVS.BIN
define_ufs 6
OMVS. U

Notice that only one share is required (in smbtab) since only one directory is being shared but all the HFS file systems must be exported (by including them in the dstab and devtab or by using dynamic export) since we want the PC user to be able to reference any file or directory below the root directory. If you are including them in dstab and devtab, you should issue the dfsexport -all command to ensure that all the file systems are exported before the dfsshare -share hfsroot command is issued to share the directory. (For more information, refer to “dfsexport” on page 110 and “dfsshare” on page 113.)

Creating a shared directory for HFS

This section describes the steps that are involved in creating a shared directory for HFS data. The HFS file system must be locally mounted on the system. Refer to the z/OS UNIX System Services Planning document, GA22-7800, for information on how to create and mount an HFS file system. Refer to the z/OS Distributed File Service zSeries File System Administration for information on how to create and mount a zFS file system.

To create a shared directory, perform the following steps:

1. Choose the HFS directory that you want to share. (For example, /abc/def/ghi)
2. Determine the HFS file system data set name that the directory /abc/def/ghi resides in. If you do not know the HFS file system data set name, the z/OS UNIX df command can help with this task. The df command displays file system data set names and their mount points. Refer to the z/OS UNIX System Services Command Reference for information on the df command. (For example, data set OMVS.ABC.DEF might be mounted on /abc/def.)

The following is an example of the df command that might help with this determination:

```
# df /abc/def/ghi
Mounted on  Filesystem    Avail/Total  Files  Status
/abc/def   (OMVS.ABC.DEF)  544/1440   4294967295 Available
```

This shows the mount point and the HFS file system data set name of the file system mounted on that mount point. The HFS file system data set name is in parentheses (in this example, OMVS.ABC.DEF).

3. If there is no entry in devtab for this HFS file system, you must add an entry in /opt/dfslocal/var/dfs/devtab which maps a unique minor device number to the HFS file system you want to export and share. Choose a unique minor device number (for example, 2) that is not in any other define_ufs statement and put it in the define_ufs statement. Put the HFS file system name (in this example, OMVS.ABC.DEF) on the next line. An example of an entry for an HFS file system might look like the following:

```
* HFS devices
define_ufs 2
OMVS.ABC.DEF
```

4. If there is an entry in devtab for this HFS file system, you must add a corresponding entry in dstab. Use the same minor device number (in this example, 2) in the device parameter (so in this example, it would be /dev/ufs2). Use a unique file system name (for example, hfs2), a file system type of ufs and a unique file system ID (for example, 101). Finally, use a unique fileset ID (for example, 0,,1715). An example dstab entry might look like this:

```
/dev/ufs2  hfs2  ufs  101  0,,1715
```
Note: If you are using only SMB protocols (and not DCE DFS protocols), you can use the same
number for all the numeric values in a dfstab entry as long as the number used is unique. For
example, the dfstab entry might look like this:
/dev/ufs2 hfs2 ufs 2 0,,2

5. Add an entry in smbtab for the directory you want to share. Use the same minor device number (in
this example, 2) in the device name parameter (so in this example, it would be /dev/ufs2). Choose a
unique share name (for example, myshare), a file system type of ufs, a description, share permission
(for example, r/w), maximum users (for example, 100) and the directory name (in this example /ghi).
The directory name is relative to the root of the file system that the device name refers to.
/dev/ufs2 myshare ufs "My share description" r/w 100 /ghi

6. Issue the dfsshare command to cause the new share to be made available to PC users.
   # dfsshare -share myshare
At this point, a PC user can connect to this share.

If there were additional file systems below the shared directory, you can export those too so that PC users
can access those directories and files. That is, if there were one or more subdirectories below the ghi
directory, and there were one or more file systems mounted on those directories or their subdirectories,
those file systems could be exported to make them available to PC users of the share named myshare.
Those file systems would be specified in dfstab and devtab with different unique minor device numbers.
Alternatively, you can use the dynamic export capability. Refer to Dynamic export for HFS.

If there was an additional directory at the same level as the ghi directory and you wanted to share that
directory, the minor device number would be the same as the share for the ghi directory (that is, it would
be 2) since that directory is also contained in the OMVS.ABC.DEF file system.

If you mount (or unmount) a file system on (from) a directory that is in the directory path specified on an
smbtab entry and you have already shared the directory, then you should unshare (detach) and reshare
the shared directory using the dfsshare command.

Removing a shared directory for HFS
A shared directory can be made unavailable by issuing the dfsshare command with the -detach option.
For example, if you want to stop the shared directory named myshare from being available to PC clients,
you would issue the following dfsshare command:
   # dfsshare -share myshare -detach

This command makes the shared directory unavailable until the SMB server is restarted or the dfsshare
command is issued to make it available. Since the shared directory is still in the smbtab, the shared
directory would be made available again to PC clients. In order to make the shared directory permanently
unavailable, it must be removed from the smbtab file.

Making a shared directory unavailable does not affect whether the underlying file systems are exported
(that is, they remain exported). There might be other shares that apply to those underlying file systems.
Use the dfsexport command to unexport file systems.

Dynamic export for HFS
Previously, in order to make HFS file systems available to PC users through the SMB server, the
administrator had to create a dfstab and devtab entry for each HFS file system for the SMB server to

---
4. If you are using both SMB and DCE DFS protocols, the file system ID must be assigned by the fts crfldbentry command. Refer to Appendix C, “Using both SMB and DCE DFS,” on page 169 for more information.
export them at SMB server start up or on command. HFS file systems must also be mounted at the time the SMB server exports them. This meant that the SMB server could not support HFS automounted file systems.

The SMB server now has an optional function called dynamic export to support HFS automounted file systems. That is, when dynamic export is used, PC clients are able to access data in HFS file systems that are automounted. (Refer to the **z/OS UNIX System Services Planning** document for information on automounted file systems.)

**Note:** The SMB server attempts to move the ownership of a shared HFS file system to the system that the SMB server is running on when the shared HFS file system is owned by a different system if the _IOE.Move.Shared.Filesystem_ environment variable is ON in the dskern process. If this is unsuccessful (when, for example, the file system is being exported by another SMB server on the other system), the SMB server will not be able to export that file system and it will not be available to PC clients.

An environment variable (_IOE.Dynamic.Export=ON) in the dskern process, enables dynamic export. When dynamic export is enabled, the SMB server can “discover” file systems that are mounted but not yet exported, and can dynamically export them. A file system is discovered by the SMB server when a PC user references a directory (for example, via the cd command) that is a mount point. This causes the SMB server to “cross into” the mounted file system. The capability to cross file systems is controlled by the _IOE.SMB.Cross.Mounts_ environment variable in the dskern process. If it is determined that the file system is not yet exported, the SMB server (dynamically) exports it. The SMB server then continues to handle the PC user request. This dynamic export occurs even though there might be no dfstab or devtab entry for the file system. The information that would be in the dfstab and devtab entry can be determined (or assigned) by the SMB server. Note that when a PC user references a remote directory that is under control of automount, this causes the SMB server to reference the directory and this in turn causes the correct file system to be automounted. So, the file system is mounted by the time the SMB server gets control back from referencing it.

Dynamic export is actually independent of whether automount is used. That is, the discovery and dynamic export of file systems occurs whether the file system was automounted or statically mounted. This means that if you do not specify dfstab and devtab entries for file systems that are “crossed into”, the SMB server takes care of those file systems on its own. The only file systems that must have a dfstab and devtab entry are file systems that are the root of a share (that is, file systems that are represented as the first / in the smtab Directory Path Name entry).

As a simple case, when dynamic export is used, it is possible to make the entire HFS tree (including automounted file systems) available to PC users by specifying / of the root file system as the shared directory in smtab and the root file system in dfstab and devtab. (The root file system is also referred to as the version file system.) No other entries are required. Of course, PC users can only access data that they are authorized to. Also, they cannot write to file systems that are mounted read-only. Here is an example of what the smtab, dfstab, and devtab entries might look like:

```
smtab:
/dev/ufs4 hfsroot ufs "My share description" r/w 100 /

dfstab:
/dev/ufs4 hfs4 ufs 103 0,,1721

devtab:
define_ufs 4
OMVS.ROOT
```

When dynamic export is used, it is still allowable for the administrator to specify dfstab and devtab entries. You can use your current dfstab and devtab entries and still take advantage of the dynamic export function. When dynamic export assigns numbers for file systems, it uses large numbers such as:

- Minor device numbers start at 10000
• File system IDs start at 100000
• Fileset IDs start at 100000.

If you add `dfstab` and `devtab` entries, you should use numbers that are lower than these so as not to interfere with dynamically assigned numbers.

In addition, a `devtab` entry can be specified without a `dfstab` entry to set the translation option for a file system. This entry is used when the SMB server crosses into that file system and dynamically exports it. Alternatively, the translation option can be inherited from the parent file system when there is no `devtab` entry. The `_IOE_INHERIT_TRANSLATION=ON` environment variable in the `dfskern` process controls whether the file system translation option can be inherited from the parent file system.

If a file system is not referenced for a period of time (and you are using dynamic export), the administrator has the ability to control whether the SMB server should dynamically unexport a file system if it has not been referenced for a period of time. Only file systems that are not the root of a share are dynamically unexported. The `_IOE_EXPORT_TIMEOUT` environment variable in the `dfskern` process is used to specify this. Unexporting an unreferenced file system frees resources in the SMB server. Later, if the file system is referenced again, it is dynamically exported again. There is a relationship between the SMB server export timeout value and the z/OS Automount Facility `delay` timeout value. Automount waits (at least) the automount `delay` timeout period after the file system has been dynamically unexported before attempting to unmount it again.

As the PC user `cd`'s down the tree, they might encounter file systems of different file system types. Some are supported by the SMB server and some are not. SMB servers supports HFS, zFS, TFS, and AUTOMNT file systems. If you are in a sysplex with shared HFS, SMB support of zFS is limited to zFS compatibility mode file systems. SMB does not support sysplex-aware zFS (on V1R11 and above) or NFS and DFSC file system types.

If you are using DCE DFS and SMB (or DCE DFS alone), you cannot use the dynamic export capability. Dynamic export is disabled when `_IOE_PROTOCOL_RPC=ON`.

Working with automounted file systems and home directories:
It is common for an OMVS user's home directory to be automounted. This means that the user's home directory does not exist and the file system is not mounted until the home directory is referenced. Refer to the [z/OS UNIX System Services Planning](https://www.ibm.com/support/docview.wss?uid=swg21293941) document for information on automount. Using dynamic export, you can specify the root of the automount file system as the shared directory. For example, consider the following configuration files:

**Automount Facility Master File:**

```
/u /etc/home.map
```

**MapName Policy File: (/etc/home.map)**

```
Name  *       
Type  HFS     
Filesystem  OMVS.HOME.<uc_name>  
Mode    rdwr    
Duration  60     
Delay   0       
```

**smbidmap:**

```
smith
cmsmith
jones
TSJONES
```

**smbtab:**

```
/dev/ufs10 homeshare ufs "Root of Home Directories" r/w 100 /
```

**dfstab:**
Note: Notice that the cmsmith z/OS user ID is in lowercase letters. The case of the z/OS user ID in the smbidmap file has implications on directories that are mount points for automounted file systems and used in smtab entries with &USERID. Use the same case for the z/OS user ID in the smbidmap file as that used in the user ID portion of the user's home directory. (The z/OS user ID will be folded to upper case when it is used to logon to z/OS.)

With this configuration, a user could connect to the shared directory named homeshare (net use h: \computer1\homeshare), and she could then cd to her home directory (cd h:\cmsmith). This causes automount and dynamic export of the user's home file system to occur.

However, this has several possible usage problems:
- When a user connects to the shared directory, the shared directory is referenced (for example, /u on OMVS), but the actual home directory is not referenced (for example, /u/cmsmith on OMVS). Therefore, the cmsmith directory is not listed if a user does a DIR from an MS-DOS window or uses Windows Explorer to click on the shared directory drive letter (since the home directory does not exist yet). The home directory is not created by the Automount Facility until it is directly referenced by name. This requires the user to cd to the home directory in an MS-DOS window. (If the user tries to create a new folder with the home directory name, Windows first tries to create a folder by the name of New Folder. This causes the Automount Facility to try to mount a file system called OMVS.HOME.NEW FOLDER and this is an invalid name.)
- In addition, the name that the user needs to cd to is not the Windows user ID but rather the z/OS user ID (the z/OS user ID that is mapped by the smbidmap file; or the guest user ID). This name might not be obvious to the PC user.
- Even after the user issues cd to the correct directory, the file system can get unexported and unmounted after a while if there is no access to the file system for a period of time. This can put the user back into the situation where the home directory no longer exists.

To resolve this, you can create a separate shared directory for each user that points directly to the user's home directory. For example:

```
  smtab:
  /dev/ufs10  smith  ufs  "Smith's Home Directory"  r/w  100  /cmsmith
  /dev/ufs10  jones  ufs  "Jones' Home Directory"  r/w  100  /tsjones
  ...
  ...

  dfstab:
  /dev/ufs10  hfs10  ufs  10  0,,10

  devtab:
  define_ufs 10
  "*AMD/u"  text
```

This technique also has several disadvantages:
- You need a separate smtab entry for each user. As users are added and deleted from the system, you must add or delete an smtab entry.
- Since each shared directory resides in a corresponding automounted file system, none of those file systems ever get unexported (or unmounted) after they are referenced.
- Each PC user must net use to a different shared directory name.
Recommended technique for PC user access to automounted home directories

The SMB server defines a special keyword to be used in the Directory path name in an smbtab entry. The keyword is &USERID. It represents the PC user’s z/OS user ID. This keyword allows a single shared directory (that is, a single smbtab entry) to mean a different directory based on the z/OS user ID of the PC user connecting to the shared directory name. When a user connects to this shared directory name, a new (special) share is created. This share has a connection reference count that is incremented when the PC user connects to the shared directory name. The reference count is decremented when the PC user disconnects from the shared directory name by issuing a net use h: /d. When the reference count is zero, the user’s home file system is eligible to be unexported by the SMB server and then unmounted by the Automount Facility. The user would need to connect to the shared directory name again to access the home directory. In addition, if the user was inactive on the session for the _IOE_SMB_IDLE_TIMEOUT time period, the session would be disconnected. This would also decrement the reference count and if zero, an unexport and unmount would occur. In this case, when the user references the drive letter again, the mount and export would occur automatically. For example, with the following entries:

smbtab:
/dev/ufs10 myhomedir ufs "Each user's Home Directory" r/w 100 /&USERID
/dev/ufs10 homeshare ufs "Root of All Home Directories" r/w 100 /

dfstab:
/dev/ufs10 hfs10 ufs 10 0,,10

devtab:
define_ufs 10
"+AMD/u" text

User smith could issue the following net use command:
net use h: \computer1\myhomedir

This would cause the SMB server to reference the /u/cmsmith directory. This would cause the Automount Facility to create the cmsmith directory in the *AMD/u file system. Then the Automount Facility would mount smith’s home directory file system on the /u/cmsmith directory. The SMB server would then export smith’s home directory file system and then create a (special) share and increment the reference count. PC user smith would then be able to issue DIR in an MS-DOS window for the drive letter or use Windows Explorer to click on the drive letter to see the contents of the home directory.

If necessary, smith could still access jones’ home directory (assuming proper authorization) by issuing net use x: \computer1\homeshare and then cd x:\TSJONES.

Note: The z/OS user ID specified in the smbidmap file determines the directory name (with the exact case) that will be substituted for the &USERID keyword in the smbtab. If a net use to that shared directory in the smbtab causes the automount to occur, it will also cause the directory to be created in the *AMD/u file system with that exact case. You should specify the z/OS user ID in the smbidmap file exactly as the user ID in the user’s home directory is specified. Otherwise, you might create a directory that does not match the user’s home directory and that user will not be able to log on locally to OMVS.

File data translation for HFS

Distributed File Service SMB support provides basic support for character data translation. There are several mechanisms that give you the ability to specify when file data should be translated. Character data translation can be:

- Based on the HFS file tag (if Enhanced ASCII is active in the SMB server, that is, the _IOE_HFS_FILETAG environment variable is set to QUERY or SET in dskern),
- Based on the HFS file format attribute,
- A global specification based on the file name suffix (hfsattr file),
- Specified on a file system basis (devtab hfs-data-set-name text or binary option),
- Specified on a file system basis based on file’s contents (devtab hfs-data-set-name auto option),
Inherited from a parent file system’s translation option (_IOE_INHERIT_TRANSLATION environment variable is set to ON in dfskern),

A global specification to translate or not for all HFS file systems exported (_IOE_HFS_TRANSLATION environment variable set to ON or OFF in dfskern),

A global specification based on file’s contents for all HFS file systems exported (_IOE_HFS_TRANSLATION environment variable set to AUTO in dfskern).

The following sequence determines whether file data is translated:

1. If the file exists and Enhanced ASCII is active in the SMB server (_IOE_HFS_FILETAG is QUERY or SET in the dfskern process) and the file is tagged, then when the tag indicates binary, no translation is done. When the tag indicates text, translation is done based on the codepage in the tag.

2. If the file exists and has no file tag (or Enhanced ASCII is not active in the SMB server) and has a non-zero file format attribute, then a value of 1 (meaning binary) causes no translation to be done. A value greater than 1 causes translation to be done. Refer to the z/OS UNIX System Services Programming: Assembler Callable Services Reference for information on the stat and the chattr callable services. These callable services can be used to query or change a file format attribute. The SMB server does not convert between different end of line delimiters.

Note: The OMVS ISPF Shell (IShell) can be used to query or set the file format attribute for a file. Choose a file and then choose attributes. This shows the file format attribute (NA, Binary, NL, CR, LF, CRLF, LFCR, CRNL). Choose Edit and then File Format. You can then change the file format. The IShell uses choice numbers for the file format attributes that are one greater than those used for the actual file format value (that is, a file format of binary has a value of 1 but the Binary choice in the IShell is 2).

3. 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. Refer to the hfsattr” on page 98.

4. The file system’s devtab translation option (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. Refer to “devtab” on page 91.

5. The file system’s inherited translation option is used if _IOE_INHERIT_TRANSLATION is ON in the dfskern process and no devtab translation option is specified for the file system. Refer to “Dynamic export for HFS” on page 42 and to the _IOE_INHERIT_TRANSLATION environment variable on page 149.

6. The dfskern _IOE_HFS_TRANSLATION environment variable is used if the file system does not inherit a translation option. Refer to the _IOE_HFS_TRANSLATION environment variable on page 148.

The file format attribute 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 file system or globally) was used to determine whether to translate. If the file is determined to be text, the file format attribute is set to 2 (NL). If the file is determined to be binary, the file format attribute is set to 1 (Binary). In addition, if the file is determined to be text, the file is being created and the _IOE_HFS_FILETAG environment variable in the dfskern process equals SET, the file tag is set to the codepage from the MOUNT TAG (if it exists) or the local codepage of the dfskern process.

Translation is done using ISO8859-1 for network data (or the codepage specified in the _IOE_WIRE_CODEPAGE environment variable of dfskern) and the local code page for the dfskern process is used for data in HFS. See “ioepdcf” on page 100 for information on how to change the local code page for the dfskern process.

You should use caution if you change an HFS devtab option from binary to text or from text to binary. If you have already stored a file under one option (for example, text), and then you change the option (to, for example, binary), the data has been translated from ASCII to EBCDIC when it was originally written to HFS, but is not translated back from EBCDIC to ASCII when it is read. This causes the data to appear
garbled to the PC user. Also note that if you are using dynamic export and translation inheritance, the translation option is effectively changed if you first mount the file system (that has no devtab translation option) under a text file system and later under a binary file system.

**Authorization for HFS**

When a PC user attempts to access a directory or file, the normal HFS file authorization mechanism is used (including Access Control Lists (ACLs)). The PC user’s SMB user ID is mapped to a local z/OS user ID and that z/OS user ID is used to determine if the user is authorized to the file or directory. Refer to Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33 for information on how SMB users are mapped to z/OS users. A z/OS user ID’s authorization to a directory or file is determined by the permission bits and the user and group IDs of the directory or file. Refer to the z/OS UNIX System Services User’s Guide for more information on HFS security.

The only authorization that a PC user can directly change is the write (w) permission of a file. It can be changed by modifying the read-only attribute of a file. Setting the read-only attribute on turns the write permission off, and vice versa. To change the read-only attribute, use the **attrib** command or right click on the file from Windows Explorer and choose Properties. The PC user must be the owner of the file (or must have a **UID=0**) to change the write permission. The read-only attribute for a directory does not stop a user from creating or deleting files in the directory. So, modifying the read-only attribute of a directory does not change the write permission of a directory and is in effect, ignored by the SMB server.

**Note:** When the read-only attribute is turned on (which turns off write permission), all the write permissions (for user, group and other) are turned off. When the read-only attribute is turned off (which turns on write permission), only those write permissions that intersect with the default create permissions are turned on. Default create permissions are specified in the **smbtab** entry for the shared directory or globally, on the **_IOE_SMB_DIR_PERMS** and the **_IOE_SMB_FILE_PERMS** environment variables.

From experimentation, it appears that when a read-only file is copied via drag and drop, the read-only attribute is maintained on the new file. However, when the MS-DOS **copy** command is used, the read-only attribute is not maintained.

When a PC user does not have read and execute permission to a directory, the contents of the directory is not listed. The user is allowed to **cd** to the directory but when a **dir** is issued, access is denied or the contents of the directory shows up as if it were empty. The user is not allowed to access files in that directory. When a PC user does not have write and execute permission to a directory, they are not able to create, erase, or rename a file (or directory) that is contained in that directory. When a PC user does not have read permission to a file, they are not able to read (or execute) the file. When a PC user does not have write permission to a file, they are not able to change the contents of the file.

**Free space for HFS**

The amount of free space that is reported for a drive letter that is mapped to a shared directory that resides in an HFS file system is based on the amount of free space in the file system that contains the shared directory. That is, if you cross into a lower file system by referencing subdirectories that reside in a lower file system, the free space does not reflect the amount of free space in the lower file system. Rather, the free space left in the file system that contains the shared directory is reported.

**Sharing RFS files**

This section discusses the RFS file system.

**Note:** The RFS file system supports reading and writing, but it is more suitable for applications that only read files. (You can limit an RFS shared directory to read-only access by making it read-only in the **smbtab**.) Applications can write to RFS files, but there are many restrictions. An application that...
writes to or creates RFS files must abide by all the restrictions imposed by RFS. Refer to “RFS restrictions” on page 69 for a full description of RFS restrictions.

Exporting and sharing RFS file systems

A Record File System (RFS) is a collection of data sets with a common data set name prefix. The data sets supported include sequential data sets (on DASD), partitioned data sets (PDS), partitioned data sets extended (PDSE) and Virtual Storage Access Method (VSAM) data sets. The supported VSAM data set types are key sequential (KSDS), relative record (RRDS) and entry sequence (ESDS). These data sets are then exported by the SMB server as a single “file system”. An RFS file system is not locally mounted so it is not part of the HFS hierarchy. It can still be exported by the SMB server, even though it is not part of the HFS hierarchy. An RFS file system must be exported by the SMB server before a directory contained in it can be shared. Exporting is done on an RFS file system basis.

Sharing is done on a directory tree basis and allows PC clients to access data in an RFS file system. In order to allow a directory to be shared, the file system that the directory is contained in must be exported.

Suppose we had the following data sets on a z/OS system:

- USERA.PCDSNS.SEQ1
- USERA.PCDSNS.SEQ2.DATA
- USERA.PCDSNS.VSAM1
- USERA.PCDSNS.PDS1() with members M1 and M2
- USERA.PCDSNS.PDS2.SOURCE() with members S1 and S2

These data sets would be represented by the following RFS hierarchy:

```
/  
|  
seq1 seq2.data vsam1 pds1 pds2.source
 |       
|   
m1  m2
|   
s1  s2
```

This hierarchy is made up of data sets that all begin with the same prefix (USERA.PCDSNS). Notice that the RFS file names do not include the prefix. The files are all directly below the root of the file system. PDSs and PDSEs appear as directories with their members as files in the directory. All file and directory names appear in lower case. (This is controlled by a setting in the rfstab file for this file system.)

The RFS file system is defined by the data set name prefix. The RFS file system includes all data sets that begin with that data set name prefix. Refer to the following representation of this:
smbtab, dfstab, and devtab entries for RFS

The files that the SMB server uses to relate the shared directory and its exported file system are the smbtab, the dfstab, and the devtab. (They are all located in /opt/dfslocal/var/dfs.) A common minor device number is used to tie related entries in these three files together. The smbtab is used to define the shared directory. The dfstab and the devtab are used to define the file system to be exported.

In smbtab, the minor device number is specified using the format /dev/ufs$n$, where $n$ is a locally assigned unique minor device number that must refer to the file system data set where the root of the shared directory path name resides (that is the file system where the first / resides). This should always be the file system where the directory that you want to share resides.

In dfstab, the minor device number is specified using the format /dev/ufs$n$ to identify the assigned unique identifiers for the file system.

In devtab, the minor device number is specified using the format define_ufs $n$ to identify the data set name prefix for the RFS file system.

If the shared directory is /, then the smbtab, dfstab, and devtab entries might be:

smbtab:
/dev/ufs10 myrfsshare ufs "My rfs share description" r/w 100 /

dfstab:
/dev/ufs10 rfs10 ufs 110 0,,1730

devtab:
define_ufs 10 rfs
USERA.PCDSNS

Creating a shared directory for RFS

This section describes the steps that are involved in creating a shared directory for RFS data. Refer to z/OS DFSMS Using Data Sets for information on how to use data sets. (RFS file systems are not locally mounted in the HFS hierarchy.)

To create a shared directory, perform the following steps:

1. Choose a set of data sets with a common data set name prefix that you want to share with PC clients (for example, USERA.PCDSNS).
2. Choose the RFS directory that you want to share. Usually, it is / is a virtual directory that is the root of the RFS file system and contains all the data sets that begin with the common data set name prefix.
3. Add an entry to the `devtab` (/opt/dfslocal/var/dfs/devtab) for this RFS file system. This entry maps a unique minor device number to the RFS file system you want to export and share. Choose a unique minor device number (for example, 10) that is not in any other `define_ufs` statement and put it in the `define_ufs` statement for this RFS file system. Put the RFS file system data set name prefix (in this example, `USERA.PCDSNS`) on the next line. An example entry for an RFS file system might look like the following:

```
* RFS devices
define_ufs 10 rfs
USERA.PCDSNS
```

4. You must add a corresponding entry in `dfstab` (/opt/dfslocal/var/dfs/dfstab) for this RFS file system. Use the same minor device number (in this example, 10) in the device parameter (so in this example, it would be `/dev/ufs10`). Use a unique file system name (for example, `rfs10`), a file system type of `ufs` and a unique file system ID (for example, 110). Finally, use a unique fileset ID (for example, 0,,1730). An example `dfstab` entry might look like this:

```
/dev/ufs10 rfs10 ufs 110 0,,1730
```

Note: If you are using only SMB protocols (and not DCE DFS protocols), you can use the same number for all the numeric values in a `dfstab` entry as long as the number is unique. For example, the `dfstab` entry might look like this:

```
/dev/ufs10 rfs10 ufs 10 0,,10
```

5. Add an entry in `smbtab` (/opt/dfslocal/var/dfs/smbtab) for the directory you want to share (the directory you chose in Step 2 on page 50). Use the same minor device number (in this example, 10) in the device parameter (so in this example, it would be `/dev/ufs10`). Choose a unique share name (for example, `myrfsshare`), a file system type of `ufs`, a description, share permission (for example, `r/w`), maximum users (for example, 100) and the directory name (in this example, `/`). For example, the `smbtab` entry might look like this:

```
/dev/ufs10 myrfsshare ufs "My rfs share description" r/w 100 /
```

6. Issue the `dfsshare` command

```
# dfsshare -share myrfsshare
```

At this point, a PC user can connect to this share.

**Removing a shared directory for RFS**

A shared directory can be made unavailable by issuing the `dfsshare` command with the `-detach` option. For example, if you want to stop the shared directory named `myrfsshare` from being available to PC clients, you would issue the following `dfsshare` command:

```
# dfsshare -share myrfsshare -detach
```

This command makes the shared directory unavailable until the SMB server is restarted or the `dfsshare` command is issued to make it available. Since the shared directory is still in the `smbtab`, the shared directory would be made available again to PC clients. In order to make the shared directory permanently unavailable, it must be removed from the `smbtab` file.

Making a shared directory unavailable does not affect whether the underlying file systems are exported (that is, they remain exported). There be other shares that apply to those underlying file systems. The `dfsexport` command be used to unexport file systems.

---

5. If you are using SMB and DCE DFS protocols, the file system ID must be assigned by the `fts crfldbentry` command. Refer to Appendix C, "Using both SMB and DCE DFS," on page 169 for more information.
File data translation for RFS

Distributed File Service SMB support provides basic support for character data translation. Character data translation can be specified globally (that is, for all RFS file systems) or on a per RFS file system basis. Character data translation can be:

- Specified on an RFS file system basis (devtab rfs_data_set_name_prefix text or binary option),
- Specified in the attributes file (rfstab) on an RFS file system basis (devtab rfs_data_set_name_prefix attrfile option),
- Specified in a global attributes file (rfstab) (_IOE_RFS_ATTRIBUTES_FILE environment variable in the dfskern process),
- Specified globally (_IOE_RFS_TRANSLATION environment variable in the dfskern process).

The following sequence determines whether RFS file data is translated:

1. The RFS file system’s devtab translation parameter (text or binary) is used to determine if translation is done.
2. If there is no translation parameter on the devtab for the RFS file system, then translation is controlled by the attributes file (rfstab) for the RFS file system’s text or binary specification.
3. If there is no attributes file (rfstab) for the RFS file system, or if there is no text or binary specification in the rfstab, then translation is controlled by the global attributes file (rfstab). The global attributes file is specified in the _IOE_RFS_ATTRIBUTES_FILE environment variable of the dfskern process.
4. If there is no global attributes file (rfstab), then translation is controlled by the global translation specification for RFS file systems. The global translation specification for RFS file systems is specified in the _IOE_RFS_TRANSLATION environment variable of the dfskern process.

Translation is done using ISO8859-1 for network data (or the codepage specified in the _IOE_WIRE_CODEPAGE environment variable of dfskern) and the local code page for the dfskern process is used for data in RFS. See “ioepd$” on page 100 for information on how to change the local code page for the dfskern process.

When translation of RFS data occurs, the end of line character used when the data set records are converted to byte stream data is controlled by the attributes file (rfstab lf or crlf entry) that is controlling the RFS file system.

You should use caution if you change a translation option from binary to text or from text to binary. If you have already stored data under one option (for example, text), and then you change the option (to, for example, binary), the data has been translated from ASCII to EBCDIC when it was stored in RFS, but is not translated back from EBCDIC to ASCII when it is read. This causes the data to appear garbled to the PC user.

Authorization for RFS

When a PC user attempts to access a directory or file, the normal data set authorization mechanism is used. The PC user’s SMB user ID is mapped to a local z/OS user ID and that z/OS user ID is used to determine if the user is authorized to the data set. Refer to Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33 for information on how SMB users are mapped to z/OS users. A z/OS user ID’s authorization to a data set is determined by the local security subsystem (for example, RACF).

A PC user cannot directly change any attributes of an RFS file. Attempting to change an RFS file attribute using the attrib command or right clicking on the file from Windows Explorer and choosing Properties appears successful but is ignored by the SMB server.

When a PC user does not have authority to a PDS, the contents of the directory is not listed. The user is allowed to cd to the directory but when a dir is issued, access is denied or the contents of the directory shows up as if it were empty. The user is not allowed to access files in that directory. When a PC user does not have update authority to a PDS, they are not able to create new members or delete or update
existing members. When a PC user does not have read authority to an RFS file, they are not able to read
the file. When a PC user does not have update authority to an RFS file, they are not able to change the
contents of the file.

Free space for RFS

The amount of free space that is reported for a drive letter that is mapped to a shared directory that
resides in an RFS file system is a fabricated number. This is due to the fact that there really is no RFS file
system, but only a set of data sets with the same data set name prefix. The SMB server always reports a
capacity of 122,880,000 bytes with half (61,440,000 bytes) available. Free space for RFS is actually
controlled by various aspects of the system such as z/OS DFSMS System Managed Storage, free space
on a volume, primary and secondary allocations, etc.

Logon considerations

In order for you to create a session and access resources (files or printers) from a PC, the SMB server
must be able to identify you in terms of a local z/OS user ID. This session creation occurs on the first
command or communication with the SMB server. The SMB server can identify you either by
authenticating you by a user ID and password that you supply, or if that fails, by allowing you to run
unauthenticated with a guest z/OS user identification. If both of these fail, your session is denied.

1. In order to authenticate you, the SMB server must receive an SMB user ID and password. The SMB
   user ID that is received is normally the user ID you specified when you logged on to your PC. You can,
   however, specify a user ID on a Windows net use command. The SMB user ID that is received must
   be mapped to a z/OS user ID. This mapping is accomplished through the smbidmap file. Refer to
   Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33 for information on the smbidmap
   file.

   The password is either a clear text password or it is an encrypted password. The type of password
   expected by the SMB server depends on whether passthrough authentication is enabled (refer to the
   _IOE_SMB_AUTH_SERVER dfskern environment variable) and if not, the setting of the
   _IOE_SMB_CLEAR_PW environment variable for the dfskern process.

   a. When passthrough authentication is enabled, this means that when the SMB server receives a
      logon request, the SMB server forwards the logon request to a Windows 2003 domain controller for
      authentication. In this case, the user ID and password must be the user’s Windows Domain user
      ID and password. If the domain controller successfully authenticates the user, then the SMB server
      maps the SMB user ID to a local z/OS user ID using the smbidmap file.

   b. If _IOE_SMB_CLEAR_PW is set to REQUIRED (or it is unspecified), then the SMB server tells
      your PC to send a clear password for authentication. This password needs to be your z/OS
      password. Normally, your PC sends the password that you specified when you logged on to your
      PC and therefore, it must match your z/OS password. The SMB server supports mixed case
      passwords for your z/OS password.

      Some levels of Windows do not send a clear password unless a Registry entry is added. Refer to
      Appendix B, “Additional information about using the SMB server,” on page 163 for more information
      on this topic.

      If, however, you specify a password on a net use command, then that is sent to the SMB server
      and that password must match your z/OS password. In this case, your PC logon password and
      your z/OS password do not need to be the same.

      If the password is not specified on net use, you be prompted for a password and that password is
      sent to the SMB server. Some commands do not prompt (for example, net view) and therefore fail
to create a session. You should use net use or Find Computer as the first communication in order
to be prompted for the password and create a session.
c. If `_IOE_SMB_CLEAR_PW` is set to NOTALLOWED, then the SMB server tells your PC to send an encrypted form of the password. Your PC encrypts the password that you specified when you logged on to your PC. Because the SMB server needs to determine your actual SMB password for authentication, your PC logon password must be stored into your RACF DCE segment by using the OMVS `smbpw` command prior to creating a session with the SMB server. If you change your PC logon password, you must also change the SMB password in your RACF DCE segment.

If, however, you specify a password on a `net use` command, then that is used for authentication and must match the SMB password in your RACF DCE segment. In this case, your PC logon password and your SMB password do not need to be the same.

2. If authentication fails, then you be allowed to run as a guest z/OS user ID. This is determined by the `_IOE_MVS_DFSDFLT` environment variable for the `dfsckern` process. If `_IOE_MVS_DFSDFLT` is set to a valid z/OS user ID, then the SMB server allows you to run with this user ID. Otherwise, your session is denied.

Using Windows Terminal Server as a client to the z/OS SMB server

Normally, when a PC connects to the z/OS SMB server, each PC has its own communications session to the z/OS SMB server. See Figure 9 for an example.

![Figure 9. SMB server with a single user session on each communications session](image)

When the Windows Terminal Server is used by a PC, multiple PC users connect to the Windows Terminal Server. They see another entire Windows system screen in a window of their local PC. This window represents applications running on the Windows Terminal Server machine. If the PC user issues a `net use` command inside that window, that `net use` command will actually be executed on the Windows Terminal Server machine. The Windows Terminal Server machine can create multiple user sessions (one for each Windows Terminal Server client) on a single communication session to the z/OS SMB server. See Figure 10 on page 55 for an example.

---

6. The PC does not actually send an encrypted password over the network. An algorithm call "challenge/response authentication" is used that does not actually send any passwords over the network.

7. Using the RACF DCE segment does not imply that DCE needs to be active.
Using passthrough authentication

Passthrough authentication gives you the ability to use your Windows domain controller to authenticate PC users in your domain for access to data through the SMB server. When PC users change their password on the domain controller, there is no other password that needs to be changed to access data through the SMB server. You must have a domain controller and your PC users must be enrolled in it. Also, PC users must be mapped to a local z/OS user ID.

Passthrough authentication is enabled by using the following environment variables in the /opt/dfslocal/home/dfskern/envar file:

- **_IOE_SMB_AUTH_SERVER**
  Used to set the IP address of the primary domain controller to authenticate PC users.

- **_IOE_SMB_BACKUP_AUTH_SERVER**
  Used to set the IP address of the backup domain controller (if it exists).

- **_IOE_SMB_AUTH_SERVER_COMPUTER_NAME**
  Used to set the computer name of the primary domain controller.

- **_IOE_SMB_BACKUP_AUTH_SERVER_COMPUTER_NAME**
  Used to set the computer name of the backup domain controller (if it exists).

- **_IOE_SMB_AUTH_DOMAIN_NAME**
  Used to set the domain name of the Windows domain.

If authentication at the domain controller fails, and the domain of the client is not the same as the domain of the domain controller, the login is attempted locally (encrypted or clear based on the method chosen by the domain controller), otherwise, the authentication fails. If the authentication fails (including the case where a local authentication failed), you be able to run as a guest z/OS user ID as described in item 2 on page 54.

When passthrough authentication is enabled, the SMB server forwards the logon request to a Windows domain controller to perform the authentication of the user. The domain controller determines what security protocol (for example; Kerberos, NTLM, or NTLMv2) will be used during authentication. The domain controller replies back to the SMB server with the method which it requires for the authentication. The SMB server implementation supports the NTLM and NTLMv2 authentication protocol. If the authentication method does not allow for the use of the NTLM or NTLMv2 protocol, the authentication to the domain controller will fail and local authentication might be attempted. During the authentication to the domain controller, audit records are produced in the Windows security event log of successful and unsuccessful.
authentication attempts. Reviewing this log can be used to determine the reason for the failed passthrough
authentication attempt. For details on setting the authentication levels for Windows XP and Vista, see
"Adjusting the authentication level on Vista and XP."

Passthrough authentication does not allow multiple sessions when digital signing is active. An example of
this would be an SMB client concurrently running multiple Windows processes to the SMB server, such as
applications running as scheduled tasks. Therefore, multiple sessions are only supported with passthrough
authentication when digital signing is not required.

To properly authenticate using passthrough authentication, the SMB server must establish a
communications session with the domain controller. This communications session must be maintained in
order for the SMB server to handle new user authentications. If the communications session that the SMB
server has with the domain controller drops, new users coming to the SMB server (and existing users
attempting to re-authenticate due to idle timeout of the PC communications session) will not be able to be
authenticated until a domain controller communications session can be reestablished. In addition, new
users on an existing PC communications session will not be able to authenticate until that PC
communications session has been reestablished (that is, re-synchronized with the domain controller
communications session). The SMB server will attempt to reestablish a domain controller communications
session and to resynchronize existing PC communication sessions as soon as possible.

Note: Passthrough authentication will not be used if the domain controller allows guests. In this case,
local authentication is used.

### Adjusting the authentication level on Vista and XP

To adjust the authentication level on Vista and XP:

1. Click the Start button
2. Select Control Panel, and then Administrative Tools.
4. Using Table 3, select the correct setting for your environment.

Here are the possible level settings and descriptions for the LAN Manager authentication:

<table>
<thead>
<tr>
<th>Value</th>
<th>Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Send LM &amp; NTLM responses</td>
<td>Clients use LM and NTLM authentication and never use NTLMv2 session security; domain controllers accept LM, NTLM, and NTLMv2 authentication.</td>
</tr>
<tr>
<td>1</td>
<td>Send LM &amp; NTLM - use NTLMv2 session security if negotiated</td>
<td>Clients use LM and NTLM authentication, and use NTLMv2 session security if the server supports it; domain controllers accept LM, NTLM, and NTLMv2 authentication.</td>
</tr>
<tr>
<td>2</td>
<td>Send NTLM response only</td>
<td>Clients use NTLM authentication only and use NTLMv2 session security if the server supports it; domain controllers accept LM, NTLM, and NTLMv2 authentication.</td>
</tr>
<tr>
<td>3</td>
<td>Send NTLMv2 response only</td>
<td>Clients use NTLMv2 authentication only and use NTLMv2 session security if the server supports it; domain controllers accept LM, NTLM, and NTLMv2 authentication.</td>
</tr>
</tbody>
</table>
### Table 3. LAN Manager authentication settings and descriptions (continued)

<table>
<thead>
<tr>
<th>Value</th>
<th>Settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Send NTLMv2 response only/refuse LM</td>
<td>Clients use NTLMv2 authentication only and use NTLMv2 session security if the server supports it. Domain controllers refuse LM and accept only NTLM and NTLMv2 authentication.</td>
</tr>
<tr>
<td>5</td>
<td>Send NTLMv2 response only/refuse LM &amp; NTLM</td>
<td>Clients use NTLMv2 authentication only and use NTLMv2 session security if the server supports it; domain controllers refuse LM and NTLM (they accept only NTLMv2 authentication).</td>
</tr>
</tbody>
</table>

---

### RACF DCE segments for SMB encrypted password support

This section assumes that you are using the IBM RACF support. For further information on using the RACF commands that are referenced in the following instructions, see [z/OS Security Server RACF Security Administrator’s Guide](https://www.ibm.com/docs/en/zos?topic=sec-admin-guide). If you are using an equivalent security product, refer to the appropriate documentation to perform the equivalent functions.

To permit the SMB server to use encrypted passwords, each z/OS user (that a PC user is mapped to) must be set up for SMB encrypted password support. Setting up an z/OS user for SMB encrypted password support requires defining a RACF DCE segment and storing the SMB password for the z/OS user into the user’s RACF DCE segment. The RACF commands are issued from TSO.

An outline of the steps required to set up the z/OS system to have the SMB server use encrypted passwords processing is as follows:

1. **Activate class KEYSMSTR**
   ```
   SETROPTS CLASSACT(KEYSMSTR)
   ```
2. **Activate class DCEUUIDS**
   ```
   SETROPTS CLASSACT(DCEUUIDS)
   ```
3. **Define entry DCE.PASSWORD.KEY in class KEYSMSTR** being sure to supply a 16 position KEYMASKED value
   ```
   RDEFINE KEYSMSTR DCE.PASSWORD.KEY SSIGNON(KEYMASKED(nnnnnnnnnnnnnnnnn))
   ```
   A complete example of this command is:
   ```
   RDEFINE KEYSMSTR DCE.PASSWORD.KEY SSIGNON(KEYMASKED(0034639986ACCFDE))
   ```
4. **Define a RACF DCE segment for each z/OS user_id that requires the SMB encrypted password capability using the command:**
   
   ```
   ALTUSER user_id DCE
   ```
   A complete example of this command is:
   ```
   ALTUSER g1dfst2 DCE
   ```
5. **You can display the RACF DCE segment for a user by using the command:**
   ```
   LISTUSER user_id NORACF DCE
   ```
   A complete example of this command is:
   ```
   LISTUSER g1dfst2 NORACF DCE
   ```
6. **To enable the SMB server to use encrypted passwords, set the _IOE_SMB_CLEAR_PW dfskern environment variable to _IOE_SMB_CLEAR_PW=NOTALLOWED and restart the SMB server.**
7. **After encrypted passwords have been enabled, each user must enter the following command from OMVS to connect to the SMB server:**
   ```
   $ smbpw smb_login_password smb_login_password
   ```
   where `smb_login_password` is the PC user’s SMB password (specified twice).
Chapter 8. Sharing printers

Before a PC client can print to a z/OS Infoprint Server printer through the SMB server, a shared printer must be created. A shared printer represents a z/OS printer controlled by the Infoprint Server. Once the PC client is connected to the shared printer, PC users can print to that z/OS printer as though it were a local printer.

Steps for creating a shared printer

This section describes the steps that are involved in creating a shared printer for an Infoprint Server printer.

Before you begin: The printer must be defined on the z/OS system. Refer to the z/OS Infoprint Server [Operation and Administration] document for information on how to define printers on z/OS. The SMB server can be running during this procedure.

Perform the following steps to create a shared printer:

1. Choose the Infoprint Server printer that you want to share (for example, printname1). You can determine the printer definitions that are available on the system by using the OMVS lpstat -p command. Refer to the z/OS Infoprint Server User’s Guide for information on the lpstat command.
2. Add an entry in smbtab for the printer you want to share.
   a. Choose a unique minor device number (for example, 1) in the device parameter (in this example, it would be /dev/prt1).
   b. Choose a unique share name (for example, myprt), a file system type of prt, a description, and put the printer definition name in the entry (for example, printname1).
   c. In addition, put the printer type information in the entry (for example, “Generic / Text Only”). You should choose a printer type that is compatible with the printer that you chose in step [1]. Refer to the z/OS Infoprint Server Operation and Administration document for information on z/OS printers and their supported printer types.

   Note: The dfstab and devtab files do not apply to shared printers.

   Example: An smbtab entry follows:
   /dev/prt1 myprt prt "Department printer" printname1 "Generic / Text Only"

3. If the SMB server is running, issue the dfsshare command to cause the new shared printer to be made available to PC users:
   # dfsshare -share myprt
4. If the SMB server is not running, start the SMB server with the following z/OS system command.
   start dfs

You know you have created a shared printer when you can connect to this share or you can run the Windows Add Print Wizard to connect to this shared printer.

Steps for removing a shared printer

A shared printer may be made unavailable by issuing the dfsshare command with the -detach option. For example, if you want to stop the shared printer, named myprt, from being available to PC clients, you would issue the following dfsshare command:

# dfsshare -share myprt -detach
This command makes the shared printer unavailable until the SMB server is restarted or the *dfsshare* command is issued to make it available. Since the shared printer is still in the *smbtab*, the shared printer would be made available again to PC clients. In order to make the shared printer permanently unavailable, it must be removed from the *smbtab* file.

**Print data translation**

When a PC user prints a file, no data translation is done by the SMB server. Any data translation is provided by the Infoprint Server. The Infoprint Server provides data transforms (for example, PDF to AFP, PostScript to AFP, and PCL to AFP). It also provides conversion from one code page to another (such as, ASCII to EBCDIC). For print requests that come through the SMB server, the SMB server specifies ISO8859-1 (or the code page specified in the *_IOE_WIRE_CODEPAGE* environment variable of the *dfs kern* process) in the *document-codepage* job attribute. For more information, refer to the [z/OS Infoprint Server Operation and Administration](z/OS Infoprint Server Operation and Administration) document.

**Authorization**

When a PC user prints a file, the PC user’s SMB user ID is mapped to a local z/OS user ID. The z/OS user ID shows the owner of the print request on the SDSF view of the JES output queue. Refer to [Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33](Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33) for information on how SMB users are mapped to z/OS users. If you are displaying a printer queue from a PC, refer to [Chapter 11, “Accessing printers,” on page 71](Chapter 11, “Accessing printers,” on page 71).
Chapter 9. Locating the SMB server

You can use the SMB server with your PC and Linux clients to access z/OS files and printers.

Selecting and configuring the PC client connection to the SMB server ensures that network clients can use the server properly.

Set up the SMB server

This section discusses setting up the SMB server for use with the following PC and Linux clients:

- Windows XP and Vista
- Linux SAMBA

PC clients can use the following methods of connecting to the SMB server on a TCP/IP network:

- User Datagram Protocol (UDP) Broadcast
- Domain Name Service (DNS)
- Windows Internet Naming Service (WINS)
- LMHOSTS static configuration files.

To easily access the SMB server and shared resources from your Windows client, use the Windows XP My Network Places or the Windows Vista Network and Sharing Center.

**Note:** If the SMB server and your Windows client are in the same workgroup (domain) and the same subnet (network segment), then no additional set up on the client is necessary because the Windows client finds the SMB server by using UDP broadcast. If you are using Dynamic Host Configuration Protocol (DHCP) to assign client machine IP addresses, the steps to assign DNS addresses and WINS addresses on your clients might be unnecessary.

Steps for Windows XP and Vista using DNS, WINS, or LMHOSTS file

**Before you begin:** You must ensure that the client can locate the SMB server on the network when you set up your Windows client to use the SMB server.

**DNS:**

Perform the following steps to configure your Windows XP or Vista client using DNS:

1. Click the Start button.
2. Point to Settings and click Control Panel.
3. Windows XP: Double-click the Network Connections.
4. Windows Vista: Click Network and Sharing Center and then select Manage network connections from the tasks pane.
5. Right-click the Local Area Connections icon and choose Properties.
6. Click on Internet Protocol (TCP/IP).
7. Click on Properties.
8. Click the Advanced button.
9. Click the DNS tab.
10. Click the Add button in the DNS Server Addresses: in order of use area to enter the IP address(es) of one or more of your TCP/IP DNS Servers.
11. Click the radio button for either Append primary and connection specific DNS suffixes or Append these DNS suffixes (in order) and click the Add button after entering one or more TCP/IP Domain Suffixes.

**Note:** The previous information may already be supplied for you.

12. Click the OK button.
12. Click the OK button on the General tab of the Internet Protocol (TCP/IP) Properties window.

13. Windows XP: Click the Close button of the General tab of the Local Area Connection Properties window.

You should have now located the SMB server on your Windows XP client after using DNS.

**WINS:**

Perform the following steps to configure your Windows XP or Vista client using WINS:

1. Click the Start button.
2. Point to Settings and click Control Panel.
3. Windows XP: Double-click the Network Connections.
   Windows Vista: Click Network and Sharing Center and then select Manage network connections from the tasks pane.
4. Right click the Local Area Connections icon and choose Properties.
5. Click on Internet Protocol (TCP/IP).
6. Click on Properties.
7. Click the Advanced button.
8. Click the WINS tab.
9. Enter one or more IP addresses of WINS server machines in the WINS addresses: in order of use area for each WINS IP address and click Add for each.
10. Make sure the Enable NETBIOS over TCP/IP button is selected. If it is not selected, click on the radio button.
11. Click the OK button.
13. Windows XP: Click Close on the General tab of the Local Area Connection Properties window.

You should have now located the SMB server on your Windows XP client after using WINS.

**LMHOSTS file:**

If you are using the LMHOSTS file, configure the LMHOSTS file with the IP address and the computer name of the SMB server.

Perform the following steps to configure your Windows XP or Vista client using the LMHOSTS file:

1. Click the Start button.
2. Point to Settings and click Control Panel.
3. Windows XP: Double-click the Network Connections.
   Windows Vista: Click Network and Sharing Center and then select Manage network connections from the tasks pane.
4. Right click the Local Area Connection icon and choose Properties.
5. Click on Internet Protocol (TCP/IP).
6. Click on Properties.
7. Click the Advanced button.
8. Click the WINS tab.
9. Make sure the Enable LMHOSTS Lookup box is checked. If it is not checked, click on the box.
10. Click the OK button.
11. Click the OK button on the General tab of the Internet Protocol (TCP/IP) Properties window.
12. Windows XP: Click Close on the General tab of the Local Area Connection Properties window.

You should have now located the SMB server on your Windows XP client after using the LMHOSTS file.
Find the SMB server

This section discusses how to find the SMB server and access shared resources.

The Linux client can use the fully qualified name in the mount command. For examples, see "Linux Samba clients" on page 66.

The Windows PC clients can use one of the following methods for finding the SMB server:

- Windows Vista: Network and Sharing Center
- Windows XP: My Network Places
- Search Computer.

Steps for using Network and Sharing Center on Windows Vista

Before you begin: If the SMB server and your PC client are in the same workgroup (domain), perform the following steps using Network and Sharing Center to find the server:

1. Click on Start, and then Network.
2. Select Network and Sharing Center from the toolbar.
3. Click View computers and devices from the task pane.
4. The computer name of the SMB server, for example \ZOSS1, should be visible.
5. If the SMB server is not visible, turn on network discovery by selecting it from the toolbar.
6. If the SMB server is still not visible, right-click on Network and follow the instructions for "Mapping shared directories to logical drives" in "Windows Vista" on page 66.

You should now be able to locate the server.

Steps for using My Network Places on Windows XP

Before you begin: If the SMB server and your PC client are in the same workgroup (domain), perform the following steps using My Network Places to find the server:

1. Click on My Network Places.
2. Double click on Add a network place.
3. Enter the computer name of the SMB server, for example \ZOSS1.
4. Click on Next.
5. Click Choose another network location
6. Enter the computer name of the SMB server, and the name of an active SMB Share. For example, \ZOSS1\Share1.
7. Click Next.

You should be able to locate the server.

Steps for using search to find the server on Windows XP

1. Click the Start button.
2. Point to Search and click For File or Folders.
3. Click Computers or people in the Other search options area.
4. Select A computer on the network and enter the Computer Name for the SMB server.
5. Click the Search button.

You should have now located the server after using Search.
Chapter 10. Accessing data

When your PC client has located the SMB server, it can then access the shared directories to read and write file data on the z/OS system. This chapter covers the following topics:
- "Using SMB server shared directories," which includes mapping drives using Windows Vista, Windows XP, and Linux Samba
- "Accessing HFS data" on page 67
- "Accessing RFS data" on page 68

Using SMB server shared directories

This section discusses how you can access shared directories on the SMB server from the following clients:
- "Windows XP"
- "Universal Naming Convention (UNC) mapping:" on page 66
- "Linux Samba clients" on page 66
- "Windows Vista" on page 66

Note: If you are using clear text passwords in the SMB server (_IOE_SMB_CLEAR_PW=REQUIRED), you might need to update your Windows registry. See "Client does not communicate" on page 163.

Windows XP

You can use a Microsoft Windows XP client to access shared directories on the SMB server by using one of the following methods:
- Map shared directories to logical drives
- Universal Naming Convention (UNC) mapping.

You might find it easier, however, to work with logical drive letters as opposed to UNC mapping.

Mapping shared directories to logical drives:

You can map an z/OS SMB server shared directory to a logical drive by performing the following steps:
1. Click on the Start button.
2. Click Programs.
3. Click Accessories.
4. Click Windows Explorer.
5. Click the Tools pull-down menu on the Windows Explorer and click Map Network Drive.
6. Choose the letter of a free drive for the shared directory (it might already be filled in).
7. Enter the Path name of an SMB shared directory. For example, you enter the following:

   \zOSS1\myshare

   where zOSS1 is the computer name and myshare is the shared directory name.
8. Click the Finish button.

However, if your z/OS password (when using clear passwords) or your SMB password in your RACF DCE segment (when using encrypted passwords) is not the same as your Windows password, you should use the net use command with your z/OS or SMB password from the command prompt. This avoids the possibility of being logged in as DFSDFLT (that is, a guest user) if you specify DFSDFLT on the _IOE_MVS_DFSDFLT environment variable. For example,

   net use x: \zOSS1\myshare mypassword

where x is an available drive letter, zOSS1 is the computer name, myshare is the shared directory name, and mypassword is your password.
Universal Naming Convention (UNC) mapping:
You can also use Universal Naming Convention (UNC) mapping to access SMB server shared directories
by performing the following steps:
1. Enter the following command:
   `net use \\zOSS1\myshare`
   where zOSS1 is the computer name and myshare is the shared directory name.
2. Enter the following command to display the computer name and the shared directory.
   `net use`
   The following output appears:
<table>
<thead>
<tr>
<th>Status</th>
<th>Local name</th>
<th>Remote name</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td></td>
<td>\zOSS1\myshare</td>
</tr>
</tbody>
</table>
3. You can enter the following to delete the shared directory:
   `net use \\zOSS1\myshare /d`

Linux Samba clients
You can use a Linux Samba client to access shared directories on the SMB server by mapping shared
directories to mount points.
Use the Linux `mount` command or the Samba `smbmount` command to mount shared file systems to
mount points.
The following examples show the use of Linux Samba command line to mount and unmount SMB
files shares.
   `mount -t smbfs //smbserver.com/fileshare /mtpt -o username=user,password=userpw`
   `umount /mtpt`
   `smbmount //smbserver.com/fileshare /mtpt -o username=user,password=userpw`
   `umount /mtpt`

Figure 11. Linux mount and unmount commands issued from a Linux UID 0 user

Windows Vista
You can use a Microsoft Windows Vista client to access shared directories on the SMB server by mapping
shared directories to logical drives

Mapping shared directories to logical drives:
You can map a z/OS SMB server shared directory to a logical drive on the PC by performing the
following steps:
1. Click on the Start button.
2. Click on Computer and then select Map Network Drive from the toolbar. (Press ALT if you do not see
   the Tools menu.)
3. Choose the drive letter from the drive list.
4. In the Folder box, type the path name of the SMB shared directory folder or computer, or click
   Browse to locate the SMB shared directory. For example, you might type the computer name and
directory such as in the following example path name:
   \\zOSS1\myshare
   where zOSS1 is the computer name and myshare is the shared directory name.
5. Click the Finish button.
However, if your z/OS password (when using clear passwords) or your SMB password in your RACF DCE segment (when using encrypted passwords) is not the same as your Windows password, you should use the `net use` command with your z/OS or SMB password from the command prompt. This avoids you being logged in as DFSDFLT (that is, a guest user). For example,

```
net use x:\\zOSS1\myshare mypassword
```

where `x` is an available drive letter, `zOSS1` is the computer name, `myshare` is the shared directory name, and `mypassword` is your password.

You might also need to specify your domain name on the `net use` command. For example:

```
\user:mydotted.domain.name\user user
```

### Accessing HFS data

This section discusses accessing HFS data.

#### HFS directory and file name case sensitivity considerations

The SMB server makes HFS file system data available to PC clients. The HFS file system is case-sensitive. PC clients are not case-sensitive (that is, they treat uppercase letters and lowercase letters as the same when you are searching for a file).

The case of file names is significant in case-sensitive file systems and can consist of both upper-case and lower-case characters. For example, the HFS file system might have three files in it with the following names:

- `DFSSMB.DAT`
- `DFSmb.dat`
- `dfssmb.dat`

These three files have technically different names (because the HFS file system is case-sensitive) and they represent three distinct, separate objects on z/OS.

All the PC clients that the SMB server supports are not case-sensitive. This means that the case of file names is insignificant. For example, from the three example files that are listed above, only one is recognized. Which one depends on how the user specified the name and whether the PC client code folded the name to uppercase.

The following algorithm is used by the SMB server when searching for a file or directory name in HFS:

- The name received over the network is searched as is. If found, that name is returned. If not, the name is folded to lowercase letters. Then the HFS directory is searched for that name. If found, that name is returned. If not, then the name received in the SMB (still folded to lowercase) is compared against each name in the HFS directory folded to lowercase letters. The first match found is returned. Otherwise, the name is not found.
- Note that many PC clients can fold a name to uppercase letters before sending the name over the network.

When a new file or directory is created, the name received over the network is used as the name of the object. When a file or directory is renamed, the object is located as described above and then, if located, its name is replaced with the new name as received over the network.

#### HFS symbolic links

This section discusses how symbolic links are treated on the SMB server. See [z/OS UNIX System Services User's Guide](https://www.ibm.com) for information about symbolic links.
A symbolic link is a special “file” that contains another path name. It can be created by z/OS UNIX commands (for example, `ln -s old new`) and applications. It is used to redirect access to another file or directory in the file system hierarchy. The closest analogy to a symbolic link in Windows terminology is a shortcut. However, they are not the same thing. A Windows application cannot create a symbolic link. However, if a symbolic link already exists, a Windows application can access it through the SMB server with some restrictions. The SMB server supports relative symbolic links (that is, symbolic links that do not begin with a `/`). In addition, the SMB server supports absolute symbolic links (that is, symbolic links that begin with a `/`) when the `_IOE_SMB_ABS_SYMLINK` environment variable is set to `ON` in the `/opt/dfslocal/home/dfskern/envar` file. Windows applications can then refer to a directory/file that is outside the shared directory tree. Directories and files can only be accessed if the PC user is authorized to the data.

In general, symbolic links can be referenced by Windows applications if they are accessing them in a read-only manner. However, this access fails if the symbolic link contains one of the following:

- A path name that begins with `/` (that is, it is an absolute path name) unless the `_IOE_SMB_ABS_SYMLINK` environment variable is set to `ON` in the `/opt/dfslocal/home/dfskern/envar` file.
- Is circular (it traverses more than 50 symbolic links).
- A path to an object that does not exist.
- A path to an object that goes outside the shared directory tree unless the `_IOE_SMB_ABS_SYMLINK` environment variable is set to `ON` in the `/opt/dfslocal/home/dfskern/envar` file.

For the cases listed above, the path name does not display as a file of zero length (for example, if listed with a `dir` command).

Other than these restrictions, symbolic links can be used during path name resolution. For example, if you `cd` to a path name that contains a symbolic link as part of the name resolution, it is successful (and follows the symbolic link) if none of the restrictions above apply and the user is authorized to all the components in the path name. Reading a file whose name is really a symbolic link that points to another file that exists is also successful. Note that, in general, a Windows user is not able to tell when the path name being used is a symbolic link or traverses one or more symbolic links.

Writing to a file whose path name is a symbolic link is allowed in some cases. If the file exists (that is, if the symbolic link and the linked file both exist), then opening the file for update, append, or truncate succeeds. An open for create fails (regardless of whether the linked file exists or not) because the symbolic link exists.

The symbolic link itself can be renamed if it remains in the same directory. Otherwise, it is denied. A symbolic link cannot be removed by a PC user.

**HFS restrictions**

The z/OS SMB server does not permit the MS-DOS `move` command to move files from one directory to another when the two directories are in different z/OS UNIX file systems. This includes using Windows Explorer to drag and drop the files. If you want to do this, use the `copy` command to copy the files to the target directory and then erase to delete them from the source directory.

**Accessing RFS data**

This section discusses accessing RFS (Record File System) data.

**RFS directory and file name considerations**

The SMB server makes RFS data available to PC clients. The data sets supported include sequential data sets (on DASD), partitioned data sets (PDS), partitioned data sets extended (PDSE) and Virtual Storage Access Method (VSAM) data sets. The supported VSAM data set types are key sequential (KSDS), relative record (RRDS) and entry sequence (ESDS).
RFS restrictions

RFS has many restrictions because RFS files are not really hierarchical, byte stream files. They are really z/OS data sets that store record oriented data that must follow data set rules. Among the restrictions are the following:

- Data set names are always upper case.
- The data set name prefix plus the file name is limited to 44 characters. (Note that the data set name prefix is not included in file names displayed at the PC.)
- A data set name segment (characters between the dots) cannot be longer than 8 characters.
- The characters in a data set name are more limited than file names in HFS. See [z/OS DFSMS Using Data Sets](z/OS DFSMS Using Data Sets) for more information of data set naming rules.
- Use lower case file names at the PC. Data set names are mapped to upper case and are displayed as lower case if `maplower` is specified in `rfstab` (this is the default). Mixed case file names should not be used.
- A PDS or PDSE cannot be created under another PDS or PDSE.
- PDS and PDSE member names are limited to 8 characters.
- Only one PDS member can be open for write at a time.
- PDS members cannot be moved between PDSs. Copy and erase must be used.
- PDS member aliases and data set aliases are not supported and are not displayed.
- If an attempt is made to write a record that exceeds the maximum record size, the write fails.
- Editors or commands (for example, Wordpad or copy) that truncate the file to zero length before rewriting a file typically work. Editors or commands that try to replace without truncate to zero attempts to replace each record and this requires that each record be exactly the same size it was before. This is not likely and therefore the writes probably fail.
- Some editors (for example, Notepad and WordPad) do not automatically put an end of line character at the end of the last line. To RFS, this is an incomplete record. When RFS closes the data set, the incomplete record will be discarded. To avoid this problem, hit enter at the end of the last line before saving the file or use an editor that automatically places an end of line character at the end of the last line (for example, Visual SlickEdit).
- SMB access is limited to RFS data sets that are no larger than 4 GB.

In general, data sets that contain variable length text data work correctly as far as record processing is concerned (although VSAM files do not support zero length records).

See Appendix E, “Using data sets,” on page 173 for more information about accessing RFS files.
Chapter 11. Accessing printers

Once your PC client can locate the SMB server, it can then access shared printers that have been created. This allows the PC client to print data on z/OS printers. This chapter explains how to access the SMB server shared printers and how to add a printer from the following clients:

- Windows XP
- Linux SAMBA

You can map an SMB server shared printer to a logical printer on the Windows PC clients by performing the following steps:

1. Open an **MS-DOS** window.
2. Enter the following command at the DOS prompt to get a list of SMB server shared printers:
   ```plaintext
   net view \zOSS1
   ```
   where `zOSS1` is the computer name of the SMB server.
3. Choose a shared printer name from the list provided.
4. Enter the following command:
   ```plaintext
   net use lpt2: \zOSS1\myprt /persistent:yes
   ```
   where `lpt2` is the name of the logical printer that you want to map the shared printer to, `zOSS1` is the computer name of the SMB server, `myprt` is the name of the desired shared printer, and `/persistent:yes` is an optional parameter specifying that the shared printer connection should be restarted by the client after reboot.

You can map an SMB server shared printer to a logical printer on Linux clients through interactive or command line access. Use the following steps for **interactive access**:

1. Open a Linux command line window.
2. Enter the following command to obtain a list of SMB server shared printers:
   ```plaintext
   smbclient -L //\zOSS1 -U username%password
   ```
   where `\zOSS1` is the computer name of the SMB server, `username` is the name of a Linux user mapped to a TSO user, and `password` is the CLEAR or encrypted password for the mapped user.
3. Enter the following command to gain interactive access to a shared printer:
   ```plaintext
   smbclient //\zOSS1/shared-printer -U username%password
   ```
   where `shared-printer` is the name of the SMB server’s shared printer.
4. Enter the following command to print a local Linux file whose fully-qualified pathname is `/root/.profile`:
   ```plaintext
   print /root/.profile
   ```
5. Enter `queue` to view the queue of print job requests.
6. Enter `quit` to exit the interactive session.

Use the following steps for **command line** access:

1. Open a Linux command line window.
2. Enter the following command to obtain a list of SMB server shared printers:
   ```plaintext
   smbclient -L //\zOSS1 -U username%password
   ```
   where `\zOSS1` is the computer name of the SMB server.
3. Enter the following command to print a local Linux file whose fully-qualified pathname is `/root/.profile`:
   ```plaintext
   smbclient //server/prt1 -U user%pw -c "print /root/.profile"
   ```
   where `prtl` is the name of the SMB server's shared printer.
4. Enter the following command to view the queue of print job requests made through the print command above:
**Accessing shared printers**

This section shows you how to access shared printers using the Windows XP and Linux clients.

**Windows XP client**

You can access shared printers on the SMB server using a Windows XP client by performing the following steps:

1. Click the **Start** button.
2. Select **Programs**.
3. Select **Accessories**.
4. Select **Windows Explorer** and release the button.
5. Click on the **Search** button on the status bar.
6. Click **Computers or people** in the **Search for other items** area.
7. Click on **A computer on the network**.
8. Enter the **Computer Name** of the SMB server.
9. Click the **Search** button.
10. Double-click on the found computer name.
11. You might need to enter a Windows XP user ID and password associated with that users credentials.
13. If the printer you selected has been defined to the PC client, then you get a list of jobs that are queued to print for that printer.
14. If the printer you selected has not been defined to the PC client, click **Yes** to set up the printer on your computer. If you are prompted further, select the appropriate printer characteristics to get the drivers set up.
15. Click the **Finish** button.

**Adding a printer**

This section explains how to add a printer using a Windows XP client.

You might get prompted to specify a print driver if the **Printer type** specified on the **smbtab** entry for this shared printer does not exist on your PC.

Authentication occurs during the add printer wizard process. For successful authentication to complete, you might be prompted to enter a Windows XP user ID and password. If authentication fails, Windows issues a message indicating it could not connect to the printer and that access is denied.

**Windows XP or Vista client**

You can add a printer from a Windows XP or Vista client by performing the following steps:

1. Click the **Start** button.
2. Select **Control Panel**.
3. Select **Printers**.
4. Click on **Add a printer**.
5. Windows XP: Click **Next**.
6. Choose the **Network Printer** or a printer attached to another computer radio button, and click **Next**.
7. Click the correct radio button to **Browse for a printer** or **Connect to this printer** (and enter the SMB printer name or the network path), and click **Next**. For example, you could enter the following network path:

```
\zOSS1\myprt
```

where *zOSS1* is the computer name of the SMB server and *myprt* is the name of the shared printer.

8. Select the correct radio button to determine if this will be your default printer.

9. Install the driver, if necessary.

10. Click the **Finish** button.

---

**Displaying a printer queue**

Windows PC and Linux clients have the ability to display a printer queue.

Perform the following steps to display a Windows PC printer queue:

1. Click the **Start** button.

2. Open the **Control Panel** and click **Printers**.

3. Determine which printer you want to display and double-click on it.

For print requests that come through the SMB server, the SMB server specifies the SMB user ID of the submitter in the **name-text** job attribute (refer to the *z/OS Infoprint Server Operation and Administration* for more information on the **name-text** job attribute). When the printer is displayed, it shows one line for each print request for that printer that has not printed yet. When a print request contains a **name-text** job attribute, the following fields are displayed:

**Document Name**

The JES Jobname followed by the JES Jobid (as shown on the SDSF output queue panel). For a print request submitted through the SMB server, the JES Jobname is the z/OS user ID for the SMB user that submitted the print request. Refer to Chapter 6, “Mapping SMB user IDs to z/OS user IDs,” on page 33 for information on how SMB user IDs are mapped. The JES Jobid is normally the characters PS followed by a number assigned by the Infoprint Server. Print jobs submitted by using the SMB server always has JES Jobids less than 65536.

**Owner**

The **name-text** job attribute. For a print request submitted using the SMB server, this is the SMB user ID of the submitter.

For print requests that do not have a **name-text** job attribute, the **Document Name** normally contains the file name of the print request and the **Owner** contains the z/OS user ID of the submitter.

**Note:** It is best to avoid leaving a window for a remote printer open for long periods of time. While the remote printer window is open, the server is continually queried in order to update the window with the latest printer status. This might lead to unnecessary network contention.

Perform the following steps to display a Linux printer queue in **interactive** mode:

1. Open a Linux command line window.

2. Enter the following command to get a list of SMB server shared printers:

```
smbclient -L //zOSS1 -U username\password
```

where *zOSS1* is the computer name of the SMB server.

3. Enter the following command to gain interactive access to a shared printer:

```
smbclient //zOSS1/shared-printer -U username\password
```

where *shared-printer* is the name of the SMB server’s shared printer, *username* is the name of a Linux user mapped to a TSO user, and *password* is the CLEAR or encrypted password for the mapped user.
4. Enter the following command to view the queue of print job requested through the above command:
   ```
   queue
   ```

5. Enter the following command to exit the interactive session:
   ```
   quit
   ```

Perform the following steps to display a Linux printer queue in **command line** mode:

1. Open a Linux command line window.
2. Enter the following command to get a list of SMB server shared printers:
   ```
   smbclient -L //zOSS1 -U username%password
   ```
   where `zOSS1` is the computer name of the SMB server.
3. Enter the following command to view the queue of print job requested through the above command:
   ```
   smbclient //server/prt1 -U user%pw -c "queue"
   ```
   where `prt1` is the name of an SMB server shared printer, `user` is the name of a Linux user mapped to a TSO user and `pw` is the CLEAR or encrypted password for the mapped user.

---

**Using PC client print drivers with SMB server shared printers**

The SMB server acts as a print server that makes the services of the Infoprint Server available to PC clients. This allows clients with the proper print drivers to submit print jobs to the Infoprint Server through the SMB server. The available print types include the following:

- Postscript
- PCL
- text
- Advanced Function Printing (AFP).

You can access print drivers for supported Windows PC clients in either of these two ways:

- If the printer type specified on the `smbtab` (for example, **Generic / Text Only**) is available on the Windows system, Windows automatically uses that printer type (and print driver) when the printer is added by the Add Print Wizard.
Part 2. SMB support reference

This section covers reference information. Each chapter begins with a short introduction and is followed by information about the processes, the files and the commands, which are ordered alphabetically, and SMB tuning and debugging information. This part is organized into the following chapters:

- Chapter 12, “z/OS system commands,” on page 77
- Chapter 13, “Distributed File Service SMB files,” on page 83
- Chapter 14, “SMB commands,” on page 109
- Chapter 15, “Tuning and debugging guidelines,” on page 119.
Chapter 12. z/OS system commands

This section covers the following commands:

- **MODIFY**, a system command which enables you to start, stop, and query the status of the SMB daemons.
- **START** and **STOP**, system commands that enable you to start and stop the DFS Control Task (**DFS**).

You can use these commands from the operator console or from a System Display and Search Facility (SDFS) screen.
modify dfs processes

Purpose
The modify dfs command can start, stop, or query the status of DFS control task daemons or send a command string to the dfskern daemon. For additional modify dfs command options, see Chapter 15, “Tuning and debugging guidelines,” on page 119.

Format
You can use any of the following formats for this command.
modify procname,{start | stop | query} daemon
modify procname,send dfskern,[reload,{print | smbmap}] [trace,{reset | print | tsize,nnM | qtab}]

Parameters
procname The name of the DFS control task. On DFS, the default procname is DFS.
command The action that is performed on DFS daemon or daemons. This parameter can have one of the following values:
start Starts either a single DFS daemon or all the DFS daemons.
stop Stops either a single DFS daemon or all the DFS daemons.
query Displays the status and process identifier (PID) of the DFS daemon or DFS daemons.
send Sends a command string to the dfskern daemon.
daemon The name of the DFS control task daemon for which the action is being requested. This parameter can have one of the following parameters:
dfskern The dfskern daemon is the DFS File Exporter process. The dfskern daemon parameter of the send command can be further modified through the following parameters:
reload Reload or reinitialization based on the following options:
print Causes reload or reinitialization of the Infoprint Server DLL. Refer to the z/OS Program Directory for more information on the Infoprint Server DLL.
smbmap Ensures that new SMB identity mappings are used by first eliminating any cached SMB identity mappings and then reloading the updated smbidmap file.
trace Reset, print, resize, or query the internal SMB trace table based on the following options:
reset Resets the internal (wrap around) trace table to empty.
print Format and print the trace records.
tssize,nnM Sets the size specification for the trace table. The variable nn represents a numeric multiplier and M represents the multipliers KB (1024 bytes), or MB (1024 x 1024 = 1 048 576 bytes).
The `modify dfs` command is used to manually start or stop one or more DFS daemons, or to view the status of the daemons. It is especially useful in situations where a daemon has stopped abnormally. On DFS, the DFS daemons are contained in the `dfs` address space (with the possible exception of `dfskern`).

**Starting and Stopping z/OS DFS Daemons:** Using the `MODIFY` system command, you can start or stop either a single daemon or all the daemons configured on the host.

**Viewing the Status of z/OS DFS Daemons:** Using the `MODIFY` system command, you can view the status of the DFS daemons from the operator console using the query option.

**Sending a Parameter to the dfskern Daemon:** Using the `MODIFY` system command, you can send a parameter to the `dfskern` daemon.

The `reload,print` parameter causes the print DLL used by the SMB File/Print Server to communicate with the Infoprint Server to be reloaded and re-initialized. Refer to the [z/OS Program Directory](https://www.ibm.com) for more information on the Infoprint Server DLL. This enabled or updates the Infoprint Server without the need to restart `dfskern`. (You may need to update `smbtab` and you may need to issue the `dfsshare` command.) Print jobs in the process of being submitted may need to be resubmitted. An open window for a remote printer may need to be refreshed.

The `reload,smbmap` parameter eliminates any cached identity mappings and reloads a new `smbidmap` file to update identity mappings between SMB user IDs and z/OS user IDs. The new identity mappings take effect for new connections.

**Privilege Required**
This command is a z/OS system command.

**Examples**
The following example starts the DFS `dfskern` process:

```
modify dfs,start dfskern
```

The following example starts all the DFS daemons:

```
modify dfs,start all
```

The following example views the status of all the DFS Control Task daemons that are currently active on the host:

```
modify dfs,query all
```

The following example resets the internal trace table to empty:

```
modify dfs,send dfskern,trace,reset
```
modify dfs processes

In the following example, the dfskern daemon is instructed to eliminate existing identity mappings and to reload an updated smbidmap file that includes recently added or deleted user identity mapping information:
modify dfs,send dfskern,reload,smbmap

Related Information
Files:
  ioepdcf
  smbidmap
**start dfs**

**Purpose**
Starts the DFS Control Task.

**Format**
```
start procname[,start_options]
```

**Parameters**
- **procname**
  The name of the DFS Control Task. On DFS, the default procname is **DFS**.
- **start_options**
  The value passed to the **START** command. On DFS, **start_options** has only one value, `parm=' -nodfs'`. You can use the `parm=' -nodfs'` option to start the DFS Control Task without starting the DFS Control Task daemons. (This is usually used during installation verification.)

**Usage**
The **START** command is used to initiate the DFS Control Task.

You can use the `parm=' -nodfs'` option to start the DFS Control Task without starting the DFS Control Task daemons.

**Note:** In DFS, **start_options** values must be within single quotation marks. The **START** command converts all user-supplied **start_options** values to uppercase characters, unless they are within single quotation marks.

**Privilege Required**
This command is a z/OS system command.

**Examples**
The following command starts the DFS Control Task and all daemons on DFS:
```
start dfs
```

The following command starts the DFS Control Task without starting the DFS daemons:
```
start dfs,parm=' -nodfs'
```

**Related Information**
File:
- `ioepdcf`
stop dfs

Purpose
Stops the DFS Control Task processes and the DFS Control Task.

Format
stop procname

Parameters
procname The name of the DFS Control Task. On DFS, the default procname is DFS.

Usage
The STOP command stops the DFS Control Task (dfscntl) and the processes controlled by the dfscntl process. These processes are:
dfskern The dfskern daemon. The dfskern server daemon is the SMB File Server process.
export The export daemon. The export daemon permits exporting of HFS file systems.

Privilege Required
This command is a z/OS system command.

Examples
The following command stops the DFS Control Task and all daemons on DFS:
stop dfs

Related Information
File:
ioepdcf
Chapter 13. Distributed File Service SMB files

This information introduces you to the Distributed File Service SMB files. These files contain configuration parameters for server processes and define HFS files and Infoprint Server printers for sharing with Windows clients. These are EBCDIC files and each line is typically delimited by newline (X'15'). They can also be delimited by carriage return/line feed (X'0D15') as is the case if they are edited from a Windows application.

This chapter provides an alphabetic listing of all relevant SMB files.
Attributes file (rfstab)

Purpose
Contains tables describing the attributes used to manipulate RFS files in the SMB server. It also contains descriptions for:
- Data set creation attributes
- Processing attributes
- Site attributes.

The PC user can also modify data set creation attributes that provide information about a data set to the SMB server, such as the type of data set, or how the data set is allocated (for example, blocks, cylinders, or tracks). Processing attributes and site attributes can only be modified by the system administrator.

Note: The attributes file is sometimes referred to as the rfstab file.

Specifying the location of the attributes file (rfstab)
The attributes file is in either an HFS file, a fixed-block partitioned data set, or a fixed-block sequential data set with a record length of 80. The file’s location is specified by the _IOE_RFS_ATTRIBUTES_FILE environment variable for the dfskern process. For example, _IOE_RFS_ATTRIBUTES_FILE=/opt/dfslocal/var/dfs/rfstab is the default location.

The SMB server can use the same attributes file as the DFSMS/MVS™ NFS server. For example, _IOE_RFS_ATTRIBUTES_FILE="/NFSADMIN.NFSS(NFSSATT)”, causes the SMB server to use the member NFSSATT in PDS NFSADMIN.NFSS for the RFS attributes. (The NFS server attributes file is specified in the NFSATTR DD statement of the NFS server startup procedure.) NFS server attributes that are not supported by the SMB server are ignored. See "Unsupported attributes" on page 89 for a list of attributes that are not supported.

An attributes file can also be specified on an RFS file system basis. Its location can be specified in the devtab entry for the RFS file system on the same line as the data set prefix name: attrfile attributes_file (where attributes_file is the path name of the attributes file that controls the data set creation, processing and site attributes for this RFS file system). An example devtab entry might display as in the following example:

* RFS devices
define_ufs 10 rfs
USERA.PCDSNS attrfile /opt/dfslocal/var/dfs/rfstab2

If no attrfile parameter is specified in the devtab entry for the RFS file system, the attributes are taken from the global attributes file specified in the _IOE_RFS_ATTRIBUTES_FILE environment variable in the dfskern process.

If no _IOE_RFS_ATTRIBUTES_FILE environment variable is specified, then the SMB server system defaults are used for RFS attributes.

Using multipliers
Instead of entering numeric values for the attributes, you can use the multipliers K (1024), M (1024 x 1024), or G (1024 x 1024 x 1024) for specifying sizes. For example, lrecl(8192) is the same as lrecl(8K).

Data set creation attributes
The data set creation attributes are used to define the structure of data sets when creating a file. These attributes correspond to the data control block (DCB) or the job control language (JCL) parameters used to define a data set when it is created. See z/OS MVS JCL Reference for more detailed information about data set creation attributes.

The data set creation attributes are described in the following table. Defaults are underlined.
Attributes file (rfstab)

You can override these attributes by specifying an attributes file in the devtab entry for the RFS file system or by using a file creation command. For PDS and PDSE, members have the same attributes as the data set attributes, therefore the file creation attributes for members are ignored.

Table 4. Data set creation attributes

<table>
<thead>
<tr>
<th>Data Set Creation Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blks</td>
<td>Specifies that disk space (see the space attribute in this table) is allocated by blocks, except for VSAM data sets.</td>
</tr>
<tr>
<td>cyls</td>
<td>Specifies that disk space (see the space attribute in this table) is allocated by blocks, except for VSAM data sets.</td>
</tr>
<tr>
<td>recs</td>
<td>Specifies that disk space is allocated by records for VSAM data sets.blks and recs are identical for VSAM data sets.</td>
</tr>
<tr>
<td>trks</td>
<td>Specifies that disk space is allocated by tracks.</td>
</tr>
<tr>
<td>blks(0</td>
<td>quan)</td>
</tr>
<tr>
<td>dataclas(class_name)</td>
<td>Specifies the data class associated with the file creation. The class_name must be defined to DFSMS/MVS before it can be used by the client. The system-managed storage automatic class selection routine must also assign a storage class to the file being created. For more information about data classes, see <a href="https://www.ibm.com/support/docview.wss?uid=swg27016981">z/OS DFSMSdfp Storage Administration</a>.</td>
</tr>
<tr>
<td>dir(27</td>
<td>quan)</td>
</tr>
<tr>
<td>dsntype(library</td>
<td>pds)</td>
</tr>
<tr>
<td>dsorg(org)</td>
<td>Specifies the organization of a data set. org can be a physical sequential (ps) data set, direct access (DA) data set, VSAM KSDS (indexed), VSAM RRDS (numbered), or VSAM ESDS (nonindexed). This attribute is ignored for directory-oriented client commands. If you are using VSAM data sets in binary mode, then nonindexed is recommended.</td>
</tr>
<tr>
<td>keys(len,off)</td>
<td>Specifies the length and offset of the keys for VSAM KSDS data sets. Keys can only be specified when using dsorg(indexed). len and off are specified in bytes. len is in the range of 1 - 255 (the default is 64). off is in the range of 0 - 32,760 (the default is 0). When you create a VSAM KSDS data set, the records you are loading into it must be keye sequenced or the write fails. Each write of the data set is treated as a first load, and requires that the records being loaded are in ascending key sequence.</td>
</tr>
</tbody>
</table>
| lrecl(8196 | quan)                | Specifies:  
| | • The length, in bytes, for fixed-length records.  
| | • The maximum length, in bytes, for variable-length records. If the blks attribute is specified, the value must be at least 4 bytes less than the blks quantity. |
| | quan is a number from 1 to 32,760 (the default is 8196). |
| mgmtclas(mgmt_class_name)   | Specifies the management class associated with the file creation. The mgmt_class_name must be defined to DFSMS/MVS before it can be used by the client. The system managed storage automatic class selection (ACS) routine must also assign a storage class to the file being created. For more information about management classes, see [z/OS DFSMSdfp Storage Administration](https://www.ibm.com/support/docview.wss?uid=swg27016981). |
### Attributes file (rfstab)

<table>
<thead>
<tr>
<th>Data Set Creation Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>model(dsname)</strong></td>
<td>The name of the cataloged VSAM data set from which to copy data set creation attributes when creating a new VSAM data set. <em>dsname</em> is a fully qualified MVS™ data set name without quotation marks. The <em>model</em> attribute must be used with one of the <em>dsorg</em> attributes which imply a VSAM organization. You can do this by specifying the <em>dsorg</em> attributed for a VSAM in the command. See the <em>dsorg</em> entry in this table.</td>
</tr>
</tbody>
</table>
| **recfm(cccc)** | Specifies the format and characteristics of the records in the data set. *cccc* can be 1 to 4 characters, in one of the following combinations:  
- *b* | Blocked  
- *f* | Fixed-length records  
- *s* | Spanned for variable records, standard format for fixed records  
- *u* | Undefined-length records  
- *v* | Variable-length records  

Valid record format characters:  
- *b*  
- *f*  
- *s*  
- *u*  
- *v*  

In *recfm*, codes *v*, *f*, and *u* are mutually exclusive. The *s* code is not allowed for a PDS or PDSE. |
| **recordsize(avg,max)** | The average and maximum record size for VSAM data sets. *avg* and *max* are specified in bytes. They can each range from 1 - 32,760 (the defaults are 512 and 4096, respectively). These values must be equal for VSAM RRDS. |
| **rlse** | Specifies that unused space should be released from the data set the first time a new data set is closed. For slow clients with long pauses between writes, the *rlse* attribute causes space to be released from the primary extent prematurely. Further writes cause secondary space to be allocated. |
| **norlse** | Specifies that unused space should not be released from the data set. |
| **shareoptions(xreg,xsys)** | Specifies the cross-region and cross-system share options for a VSAM data set. *xreg* is a number from 1 to 4; *xsys* is either 3 or 4. The defaults are 1 and 3, respectively. |
| **spanned** | This applies to VSAM data sets only. For spanned records of non-VSAM detests, see the entry for *recfm* in this table. |
| **nonspanned** | Specifies that VSAM KSDS or ESDS data sets can contain records that span control intervals (spanned records). |
| **space(prim[,aux])** | Specifies the amount of primary and auxiliary space allocated for a new data set on a direct access volume. *prim* is the number (from 0 to 16,777,215) of primary tracks, cylinders, or data blocks in the data set. *aux* (optional) is the number (from 0 to 16,777,215) of additional tracks, cylinders, or blocks allocated if more space is needed. If this attribute is not specified, the default is used. The defaults are 100 and 10, respectively. |
| **storclas(class_name)** | Specifies the storage class associated with the file creation. The *class_name* must be defined to the DFSMS/MVS before it can be used by the client. For more information about storage classes, see [z/OS DFSMSdfp Storage Administration](https://www.ibm.com/docs/en/zos/2.4.0?topic=dfsmsdftp-admin). |
Attributes file (rfstab)

Table 4. Data set creation attributes (continued)

<table>
<thead>
<tr>
<th>Data Set Creation Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unit(unit_name)</td>
<td>Specifies the unit on which to create a data set. unit_name is a generic or symbolic name of a group of DASD devices. The unit_name must be specified as 3390 for extended format data sets. <strong>Note:</strong> You cannot create or access tape data sets on a z/OS host using the SMB Server. You cannot create extended format data sets with the SMB Server, except by ACS routines.</td>
</tr>
<tr>
<td>vol(volser)</td>
<td>Specifies the name of the DASD volume to be used to store the created data set. vol is the keyword and volser represents the volume name. If a data set is to be system-managed, as is determined by the DFSMS/MVS automatic class selection (ACS) routines, you can omit this attribute.</td>
</tr>
</tbody>
</table>

**Processing attributes**

Processing attributes are used to control how files are accessed by clients. The processing attributes are described in the following table. Defaults are underlined. The administrator can override the default processing attributes in the devtab entry for the file set. The client user cannot override processing attributes.

Table 5. Processing attributes

<table>
<thead>
<tr>
<th>Processing Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary</td>
<td>Indicates that the data is processed between the client and server using binary format and no data conversion occurs between ASCII and EBCDIC formats.</td>
</tr>
</tbody>
</table>
| text                 | Converts the contents in the data set between EBCDIC and ASCII formats. Use this format to share text between clients and z/OS applications.  
  In text mode, the following attributes apply:  
  • **blankstrip** and **noblankstrip**. See the entry for blankstrip in this table.  
  • End-of-line specifiers (lf, cr, lfc, crlf, or noeol) are used to indicate the logical record boundary. See the entry for lf in this table. |
| blankstrip           | With text mode, strips trailing blanks at the end of each record of a fixed-length text file when the file is read. Pads the end of each record or file with blanks when a text file is written. |
| noblankstrip         | Does not strip trailing blanks at the end of fixed-length records when a fixed-length text file is read. Does not pad records when writing a text file. The file must be of the correct size or an I/O error is reported to the client.  
  For information about the text mode, see the text option of “devtab” on page 81. |
| executebiton         | Turns on the execute bits in user, group, and other (as reported with the ls (list) AIX or z/OS UNIX command) for files. Use when storing executables or shell scripts on the z/OS system.  
  **Note:** This is ignored for Windows clients. |
Attributes file (rfstab)

Table 5. Processing attributes (continued)

<table>
<thead>
<tr>
<th>Processing Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>executebitoff</td>
<td>Turns off the execute bits in user, group, and other for the mount point's files. Note: This is ignored for Windows clients.</td>
</tr>
<tr>
<td>fastfilesize</td>
<td>Causes the SMB server to approximate the file size. For more information, see &quot;Handling of the file size value&quot; on page 185.</td>
</tr>
<tr>
<td>nofastfilesize</td>
<td>For Direct Access data sets (PDSs and PDSEs) and non-system managed data sets, this specifies to read the entire file or member to get the file size. Using this attribute might cause a noticeable delay when first accessing very large data sets. For more information, see &quot;Using fastfilesize to avoid read-for-size:&quot; on page 186.</td>
</tr>
<tr>
<td>mapleaddot</td>
<td>Turns on mapping of a single leading &quot;.&quot; from a client file name to a leading &quot;$&quot; on z/OS. This option is typically enabled for access by AIX and z/OS UNIX clients.</td>
</tr>
<tr>
<td>nomapleaddot</td>
<td>Turns off mapping of a single leading &quot;.&quot; from a client to a leading &quot;$&quot; on z/OS.</td>
</tr>
<tr>
<td>maplower</td>
<td>Turns on mapping of lower case file names to upper case when accessing files on z/OS, and back when sending to the network. This option is typically enabled for access by PC, AIX, or z/OS UNIX clients.</td>
</tr>
<tr>
<td>nomaplower</td>
<td>Turns off mapping of lower case file names to upper case and back when using files on z/OS.</td>
</tr>
<tr>
<td>retrieve(nowait)</td>
<td>The SMB server uses DFSMS/Shsm™ to recall or delete migrated files. The action that the server takes against the migrated files depends on which of the retrieve or noretrieve attributes is active. When the retrieve(nowait) attribute is active, the server does not wait for the recall to finish, and immediately returns a “device not available” message. You can try accessing the file later when the recall has completed.</td>
</tr>
<tr>
<td>noretrieve</td>
<td>When the noretrieve attribute is active, the server does not recall the file, and can return “device not available” upon a lookup, an rdwr, or a create request for a file. For more information, see &quot;Retrieve attributes.&quot;</td>
</tr>
<tr>
<td>setownerroot</td>
<td>Sets the user ID in a file's attributes to root.</td>
</tr>
<tr>
<td>setownernobody</td>
<td>Sets the user ID in a file's attributes to nobody.</td>
</tr>
</tbody>
</table>

Retrieve attributes

The server deletes the migrated file upon a remove request for a file, regardless of whether the retrieve or the noretrieve attribute is active. Typically, a remove request is preceded by a lookup request. If the data set was migrated with DFSMS/MVS 1.2 or below, retrieve attribute causes a recall because lookup processing needs to open the data set and read for size. If the data set was migrated under DFSMS/MVS 1.3 and DFSMShsm 1.3 or later, and is SMS managed, its attributes were saved on DASD; therefore it is not always necessary to recall the data set to read for size and the data set can be deleted without recall. If the noretrieve attribute is active, the lookup can return a “device not available” message. If the client code decides to ignore the error and go with the remove, the migrated file is then deleted.

The z/OS UNIX command ls mvsusera does not issue requests for individual files under the mvsusera directory. Migrated files under the mvsusera directory are displayed, but are not recalled. However, the z/OS UNIX command ls -l mvsusera issues lookup requests for individual files under the mvsusera directory.

Site attributes

The site attributes are used to control SMB server resources. These attributes are described in the following table. Site attributes can be overridden in the devtab entry for the file set.
### Table 6. Site attributes

<table>
<thead>
<tr>
<th>Site Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bufhigh(n)</td>
<td>Specifies the maximum size (in bytes) of allocated buffers before buffer reclamation (see the <code>percentsteal</code> attribute in this table) is initiated. ( n ) is an integer from 1MB to 128MB (the default is 2MB). If the combined total specified in the <code>bufhigh</code> and <code>logicalcache</code> attributes is greater than the available storage in the extended private area (implied by the REGION parameter in your procedure) at startup, the server shuts down immediately. A higher number means more caching and potentially better read performance.</td>
</tr>
<tr>
<td>filetimeout(n)</td>
<td>Specifies the amount of time, in seconds, before a data set is closed and data is written to DASD. The minimum specification is 30. The default is 30.</td>
</tr>
<tr>
<td>percentsteal(n)</td>
<td>Specifies the percent of the buffers reclaimed for use when the <code>bufhigh\(n\)</code> limit has been reached. A higher value means a reclaim operation is performed less often, but the cached data is significantly trimmed on each reclaim. This can result in poor read performance because readahead buffers might be stolen. Lower values result in more frequent reclaim operations, but the cached data typical water mark is higher, meaning possibly better performance by reading out of cached data. ( n ) is an integer from 1 to 99 (the default is 20).</td>
</tr>
</tbody>
</table>

#### Unsupported attributes

The following NFS Server attributes are not supported by the SMB server and are ignored.

- `attrtimeout`
- `cachewindow`
- `logicalcache`
- `maxrdforzleft`
- `noatttimeout`
- `noreadtimeout`
- `nowritetimeout`
- `readaheadmax`
- `readtimeout`
- `retrieve`
- `retrieve(wait)`
- `writetimeout`.

#### Examples

The following is an example of an attributes (`rfstab`) file:

```plaintext
blksize(6160)
dir(256)
dsnype(pds)
dsorg(ps)
keys(64,0)
lrecl(80)
recfm(fb)
recordsize(512,4K)
nonspanned
space(100,10),blks
blankstrip
if
executebiton
nofastfilesize
filetimeout(30)
mapleaddot
maplower
norettrieve
setownerroot
bufhigh(2M)
percentsteal(20)
```
Attributes file (rfstab)

Note: An example attributes file can be found in /opt/dfsglobal/examples.

Implementation Specifics

The attributes file is in either an HFS file, a fixed-block partitioned data set, or a fixed-block sequential data set with a record length of 80. The file's location is specified by the _IOE_RFS_ATTRIBUTES_FILE environment variable for the dskern process. For example, _IOE_RFS_ATTRIBUTES_FILE=/opt/dfslocal/var/dfs/rfstab is the default location.

The SMB server can use the same attributes file as the DFSMS/MVS NFS server. For example, _IOE_RFS_ATTRIBUTES_FILE=//NFSADMIN.NFSS(NFSSATT), causes the SMB server to use the member NFSSATT in PDS NFSADMIN.NFSS for the RFS attributes. (The NFS server attributes file is specified in the NFSATTR DD statement of the NFS server startup procedure.) NFS server attributes that are not supported by the SMB server are ignored. See "Unsupported attributes" on page 89 for a list of attributes that are not supported.

An attributes file can also be specified on an RFS file system basis. Its location can be specified in the devtab entry for the RFS file system on the same line as the data set prefix name: attrfile attributes_file (where attributes_file is the path name of the attributes file that controls the data set creation, processing and site attributes for this RFS file system). An example devtab entry might display as in the following example:

* RFS devices
define_ufs 10 rfs
USERA.PCDSNS attrfile /opt/dfslocal/var/dfs/rfstab2

If no attrfile parameter is specified in the devtab entry for the RFS file system, the attributes are taken from the global attributes file specified in the _IOE_RFS_ATTRIBUTES_FILE environment variable in the dskern process.

If no _IOE_RFS_ATTRIBUTES_FILE environment variable is specified, then the SMB server system defaults are used for RFS attributes.

Related Information

Commands:
   dfsexport
dfsshare

Files:
   devtab
dfstab
   hfsattr
   smbtab
devtab

Purpose
Stores identifying information for all HFS and RFS file systems to be exported (and shared). (Unless otherwise noted, HFS includes file systems of type HFS, zFS, TFS, and AUTOMNT. If you are in a sysplex with Shared HFS, SMB support of zFS is limited to zFS compatibility mode file systems.)

Format
The devtab file contains the following lines:

* comment
define_ufs n [hfs | rfs]
{hfs-file-system-name [text | binary | auto] |
  rfs-data-set-prefix [text | binary] [attrfile attributes-file] |
  rfs-data-set-name [text | binary] [attrfile attributes-file]}

Options
* comment Specifies a comment line.

define_ufs n [hfs | rfs]
  Defines an HFS file system or RFS file system, n specifies a minor device number, which is a unique number 1 - 65535. Specifying hfs (Hierarchical File System) or rfs (Record File System) determines the type of file system. hfs is the default.

  Note: hfs includes file systems of type HFS, zFS, TFS, and AUTOMNT. If you are in a sysplex with Shared HFS, SMB support of zFS is limited to zFS compatibility mode file systems.

hfs-file-system-name
  Identifies the file system name of the HFS, zFS, TFS, or AUTOMNT file system that you want exported. If you are in a sysplex with Shared HFS, SMB support of zFS is limited to zFS compatibility mode file systems.

  Note: If you do not want the hfs-file-system-name to be folded to upper case, put the name within double quotation marks. For example, "/tmp". This is typically appropriate for file systems of type TFS and AUTOMNT. It might be appropriate for HFS or zFS if the HFS file system was mounted with lowercase characters in the file system name (using an environment that does not fold the file system name to upper case, for example, ishell). To determine if a mounted file system's name includes lowercase characters, use the OMVS df command.

rfs-data-set-prefix
  Identifies the prefix of the record data sets that you want exported.

rfs-data-set-name
  Identifies the data set name of a Partitioned Data Set (PDS) or Partitioned Data Set Extended (PDSE) that you want exported.

  Note: If you export a PDS or PDSE rather than a data set name prefix, the data set will remain allocated for the entire time it is exported.

attrfile attributes-file
  Identifies the name of the attributes file (rfstab) to be used for this RFS file system.

text | binary Specifies whether the data needs to be translated (text) or not (binary). The default is controlled by the _IOE_HFS_TRANSLATION environment variable setting in the dfskern process (for HFS file systems) and by the _IOE_RFS_TRANSLATION environment variable setting in the dfskern process (for RFS file systems).
Specifies that the decision to translate HFS data is based on whether the first 255 bytes of
the file are deemed to be valid characters.

Usage
The devtab file is used to define HFS file systems and RFS file systems to be exported. The devtab file
resides in the directory named /opt/dfslocal/var/dfs. HFS and RFS file systems must be exported in order
to be accessible by PC users. The file system containing the shared directory is automatically exported
when the dfsshare command is issued (assuming it has the appropriate dstab, devtab, and smtab
entries). HFS file systems below the shared directory must be explicitly exported by using the dfsexport
command if they are being made available to PC users (unless you are using dynamic export).

The devtab file is an EBCDIC file that can be edited with a text editor. You must have write (w) and
execute (x, sometimes called search) permissions on the /opt/dfslocal/var/dfs directory to create the file.
You must have write permission on the file to edit it.

To export an HFS or RFS file system, it must be defined to SMB. It is defined by creating an entry in the
devtab file. Entering define_ufs n in the devtab file defines the type of file system as ufs, an HFS file
system, and maps a unique minor device number, n, to the HFS file system you want to export. You can
also enter define_ufs n rfs to the devtab, where rfs indicates that the file system is an RFS file system.
The minor device number is a unique identifier that can be any number greater than zero. This number
becomes part of the name of the device name (in the dfstab entry). Each HFS or RFS file system being
exported must have a unique device number defined. The file system name of the HFS file system or the
data set name prefix for RFS is entered in the devtab file on the next line after the device number
definition. Before exporting an HFS file system, the HFS file system must be locally mounted. For
information about allocating and mounting zFS file systems, see z/OS Distributed File Service zSeries File
System Administration. For information about allocating and mounting HFS file systems, see z/OS UNIX
System Services Planning. RFS file systems are not locally mounted.

You can also specify an optional character data translation parameter on the same line after the file
system name of the HFS file system or the data set name prefix of the RFS file system. The possible
values for the translation control parameter are the following:

- **binary**: Do not translate the data.
- **text**: Translate the data with the default translation tables. The default for local data is the local
code page for the dfskern process. The code page for network data is ISO8859-1 (or the
code page specified in the _IOE_WIRE_CODEPAGE environment variable of the dfskern
process). See Examples” on page 93 for an example using the text parameter.

An additional translation control parameter value (auto) for HFS is available for HFS file systems and an
additional translation configuration file (hfsattr) is available for HFS.

The additional translation control parameter value for HFS is:

- **auto**: Determine whether to translate the data based on the contents of the data. The algorithm
is as follows:
  - Outgoing data (client read) - if the first 255 bytes of data are valid EBCDIC characters
    then translate to ASCII
  - Incoming data (client write) - if the first 255 bytes of data are valid ASCII characters
    then translate to EBCDIC.

Valid characters include POSIX C printable characters plus carriage-return, newline, and
tab. In EBCDIC, the POSIX C printable characters include X'40', X'4B'-X'50', X'5A'-X'61',
X'6B'-X'6F', X'79'-X'7E', X'81'-X'89', X'91'-X'99', X'A1'-X'A9', X'AD', X'BD', X'C0'-X'C9',
X'D0'-X'D9', X'E0', X'E2'-X'E9', X'F0'-X'F9'. Tab is X'05', carriage-return is X'0D', and
newline is X'15'. In ASCII, the POSIX C printable characters include X'20'-X'2F'. Tab is
X'09', carriage-return is X'0D', and newline is X'0A'.
If the translation control parameter is omitted, the default is controlled by the _IOE_HFS_TRANSLATION environment variable setting in the dfskern process (for HFS file systems) and by the _IOE_RFS_TRANSLATION environment variable setting in the dfskern process (for RFS file systems).

In the case of an RFS file system, you can also optionally specify the name of an attributes file (rfstab) on the same line after the rfs-data-set-prefix or rfs-data-set-name by using the attrfile keyword. See "Attributes file (rfstab)" on page 84 for more information about the attributes file.

Examples

The following examples show devtab entries for HFS and RFS file systems. All entries beginning with an asterisk (*) are comment lines.

The following example shows a devtab entry for an HFS file system. The second line, define_ufs 2, defines the type of file system as ufs, an HFS file system. A unique minor device number of 2 is assigned. The line following the definition of the HFS file system minor device number, omvs.user.abc, identifies the name of the HFS file system being exported and specifies a translation control parameter of text. A translation control parameter of text means data is translated. The HFS file system device name is /dev/ufs2 with 2 representing the device’s minor number.

* HFS devices
define_ufs 2
omvs.user.abc text

Examples of other file system types:

* ZFS devices
define_ufs 6
omvs.prv.compat.aggr001 text
* TFS devices
define_ufs 4
"/dev" text
* AUTOMNT devices
define_ufs 5
"*AMD/Home" text

The following example shows a devtab entry for an RFS file system. The second line, define_ufs 3 rfs, defines the type of file system as ufs and the rfs parameter qualifies it as an RFS file system. A unique minor device number of 3 is assigned. The next line, USERA.PCDSNS, is the prefix of the record data sets that you want to export as a single RFS file system. The RFS file system device name is /dev/ufs3 with 3 representing the device’s minor number.

* RFS devices
define_ufs 3 rfs
USERA.PCDSNS text

Note: An example devtab file can be found in /opt/dfsglobal/examples.

In summary, the define_ufs 2 entry in the devtab corresponds to the /dev/ufs2 entry in the dfstab and smbtab. The define_ufs 3 rfs entry in the devtab corresponds to the /dev/ufs3 entry in the dfstab and smbtab.

Implementation Specifics

The devtab file is stored as an EBCDIC file in HFS.

Related Information

Commands:
  * dfsexport
  * dfsshare
devtab

Files:
  dfstab
  hfsattr
  rfstab
  smbtab
Purpose
Specifies HFS and RFS file systems that can be exported.

Usage
The dfstab file includes information about each file system that can be exported from the local disk and then shared with SMB clients. The file is read by the dfsexport command, which exports specified HFS and RFS file systems. (It is also read by the dfsshare command, which initializes SMB shares.) The dfstab file must be in the directory named /opt/dfslocal/var/dfs. The dfsexport command looks in that directory for the file; if the file is not there, no file systems can be exported.

The dfstab file is an EBCDIC file that can be edited with a text editor. You must have write (w) and execute (x, sometimes called search) permissions on the /opt/dfslocal/var/dfs directory to create the file. You must have write permission on the file to edit it.

The file contains a one-line entry for each HFS or RFS file system available for exporting and sharing. Each entry in the file must display on its own line.

The fields in the following list must appear for each entry; they must appear in the order listed, and each field must be separated by at least one space or tab.

Device name The device name of the HFS or RFS file system being exported; for example, /dev/ufs2.

File system name
The name associated with the HFS or RFS file system being exported. A file system name can contain any characters, but it can be no longer than 31 characters. It must be different from any other file system name in the dfstab file. File system names cannot be abbreviated, therefore you should use a short, descriptive name; for example, hfs2.

File system type
The identifier for the type of file system. For HFS and RFS file systems, it must be ufs. Enter the identifier in all lowercase letters.

File system ID
A positive integer different from any other file system ID in the dfstab file.

File set ID
The unique file set ID number associated with the HFS or RFS file set. File set ID numbers are represented as two positive integers separated by a pair of commas. For example, the file set ID number of the first file set is 0,,1. When specifying a new file set, increment the file set ID. When the integer after the commas becomes greater than $2^{32}$, the integer before the commas becomes 1 and the integer after the commas returns to 0 (zero) (that is, 1,,0). This number must be different from any other file set ID in the dfstab.

Note: The file system parameters are referred to as aggregate parameters in the Z/OS Distributed File Service DFS Administration document. If DCE DFS protocols are used along with the SMB protocols, the file set ID parameter must be assigned by the fsserver and stored in the FLDB (file set location database) by using the fts crldbentry command (see Appendix C, “Using both SMB and DCE DFS,” on page 169 for information about using the SMB protocol with the DCE DFS protocol). If DCE DFS protocols are not being used, the file set ID needs only to be unique from other file set IDs.

When the dfsexport command is executed, it reads the dfstab file to verify that each HFS and RFS file system being exported is listed in the file. A file system must have an entry in the dfstab file if it is being exported unless you are using dynamic export. To ensure that it does not export a file system that is currently exported, the dfsexport command refers to a list (in memory) of all currently exported file systems.
dfstab

**Examples**
The following `dfstab` file specifies that an HFS file system and an RFS file system (/dev/ufs2 and /dev/ufs3) can be exported:

```
/dev/ufs2 hfs2 ufs 101 0,1715
/dev/ufs3 rfs3 ufs 102 0,1718
```

There is nothing in a `dfstab` entry that distinguishes between an HFS file system and an RFS file system. It is the corresponding entry in the `devtab (define_ufs 3 rfs)` that indicates to the SMB server that the file system is an RFS file system.

**Note:** You can put comments in the `dfstab` file. Comments have a # in column 1. An example `dfstab` file can be found in `/opt/dfsglobal/examples`.

**Implementation Specifics**
The `dfstab` file is stored as an EBCDIC file in HFS.

**Related Information**
Commands:
- `dfsexport`
- `dfsshare`

Files:
- `devtab`
- `smbtab`
envar

Purpose
Specifies the environment variables for a process.

Usage
The envar file contains the environment variables for each Distributed File Service process. There is one envar file for each process. Each envar file is located in the corresponding home directory of each process. (For example, the environment variables for the dfskern process are contained in the /opt/dfslocal/home/dfskern/envar file.) See Chapter 3, “SMB post installation processing,” on page 11 for information about process home directories. The envar file is read during process initialization and each environment variable specified in the envar file is set for the process. Environment variables are specified in the envar file in the following format:

variable-name=value

Note: There should be no space between the variable-name or the value and the equal sign that separates them.

An environment variable:
- cannot be longer than 4096 characters
- must have an =
- can be continued by terminating the line with \n- must begin in column 1

A line that begins with # is treated as a comment.

The envar file is an EBCDIC file that can be edited with a text editor. You must have write and execute permissions on the process home directory to create the file. You must have write permission on the file to edit it. See Table 7 on page 143, for information about all the environment variables supported for SMB processing.

Examples
The following envar file entry (located in the /opt/dfslocal/home/dfskern/envar) specifies that SMB processing should be on:

_IOE_PROTOCOL_SMB=ON

Note: Example envar files can be found in /opt/dfsglobal/examples.

Implementation Specifics
The envar file is stored as an EBCDIC file in HFS.
hfsattr

Purpose
Contains directives that map a file name extension (suffix) to an indication whether the SMB File Server (dfskern) should translate the file data from ASCII to EBCDIC and vice versa (encoding).

Usage
The hfsattr file is a text file that is stored in HFS. The File Server locates the hfsattr file during startup (or restart) by examining the _IOE_HFS_ATTRIBUTES_FILE environment variable for the dfskern process. The hfsattr file has the following format:
AddType .suffix representation encoding [quality]

The hfsattr AddType directive has the same format as the WebSphere® Application Server uses in its configuration file (httpd.conf). You can point the File Server to this file. All other directives are ignored. The File Server only examines the first, second, and fourth fields. The others are ignored. Comments can be created by using the # character. All fields are case sensitive.

AddType A keyword that indicates a directive to map a suffix to an encoding.
.suffix The file name suffix. Wildcard characters are not permitted.
representation The MIME type and subtype you want to bind to files that match the corresponding suffix. This field is ignored by DFS.
encoding The type of data the file contains. The only value that the File Server looks for is ebc dic. This means that incoming data should be translated from ASCII (ISO8859-1 or the code page specified in the _IOE_WIRE_CODEPAGE environment variable of the dfskern process) to EBCDIC (IBM-1047 or the current code page for the dfskern process). Outgoing data should be translated from EBCDIC to ASCII. All other values for encoding are ignored and the data is not translated. Before data is translated, it is checked for valid characters to avoid translating data that is already in the correct format.
quality This is an optional indicator of the relative importance (on a scale of 0.0 to 1.0) for the content type. This field is ignored by DFS.

If the file name suffix is not found in the hfsattr file, or the file name has no suffix, then translation is determined by the devtab translation control parameter or the dfskern _IOE_HFS_TRANSLATION environment variable. For more information about devtab see "devtab on page 91", and for the _IOE_HFS_TRANSLATION environment variable, see 148.

Examples
The following is an example of an hfsattr file:

# Map suffixes to the encoding
AddType .bin application/octet-stream binary 1.0
AddType .ps application/postscript ebc dic 0.8 # PostScript
AddType .PS application/postscript ebc dic 0.8 # PostScript
AddType .c text/plain ebc dic 0.5 # C source
AddType .html text/html ebc dic 1.0 # HTML
AddType .htm text/html ebc dic 1.0 # HTML on PCs
AddType .gif image/gif binary 1.0 # GIF

Implementation Specifics
The hfsattr file is stored as an EBCDIC file in HFS.
Related Information

File:
  devtab
**Purpose**
Specifies the processes to be started by the DFS Control Task.

**Usage**
The `ioepdcf` file, also referred to as the Daemon Configuration File, is an EBCDIC text file used by the DFS Control Task to determine which daemons can be started during the initialization of DFS. The file is located in the `/opt/dfslocal/etc` directory. The information contained in the `ioepdcf` file includes the following:

**Process Name**
The name of the process to be entered in the `ioepdcf` file. Valid processes associated with SMB are:

- **dfskern**: The `dfskern` daemon. The `dfskern` server daemon is the SMB file and print server process.
- **export**: The `export` daemon. The `export` server daemon permits exporting of HFS file systems.

**Configuration Type**
The configuration type for each server. Available types for z/OS are:

- **CONFIGURED=Y**
  Specifies that the process starts during initialization. If the process abends or fails for any reason, it is automatically restarted by the DFS Control Task.
  
  **Note**: Do not use this configuration type for the `export` process. This process executes one time and then stops. The `export` process must not be automatically restarted after it has run. Use `CONFIGURED=I` or `CONFIGURED=M`.

- **CONFIGURED=N**
  Specifies that the process is not started by the DFS Control Task nor can the process be started manually.

- **CONFIGURED=I**
  Specifies that the process is started during initialization or when the `MODIFY` command, `START ALL` is issued. The process can be started manually. The process does not restart if it abends or ends for any reason.

- **CONFIGURED=M**
  The process is not started by the DFS Control Task, but can be manually started by using the `MODIFY` system command. The specified process does not restart if it ends for any reason.

**Load Module Name**
The name of the load module (LMD) in a partitioned data set. The load module refers to the name of the member in the `xxx.SIOELMOD` (where `xxx` is installation dependent) data set created during installation (for further information, see the [z/OS Program Directory](https://z10iadz00.okea.net/TS/COM/85222/G110-0669.pdf). The following are the PDS member names for the processes started by the DFS Control Task:

- **dfskern**: The `dfskern` daemon load module name. The name, `dfskern`, is an alias for the load library entry, `IOEDFSKN`. 
In addition, the `export` process is started by the DFS Control Task and executes the load library entry, `IOEDFSXP`. The `export` process has no alias.

**Special Parameters**

Parameters that are 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 can 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.

The `ioepdcf` file can be used to override the default parameter options for the following daemons:

**dfskern**

The `dfskern` server daemon is the SMB file and print server. There are no options that need to be overridden for this process for SMB file and print serving.

**export**

The `export` daemon permits exporting of HFS file systems. See “dfsexport” on page 110 for information about options available for this process.

Additional special parameters that control restart and timeout intervals might also be entered in the `ioepdcf` file. These parameters are:

**Restart**

The interval defined for the DFS Control Task to attempt a restart of the process. Restart values are entered in seconds.

**Timeout**

The maximum interval that the Control Task waits for the process to complete initialization. Timeout values are entered in seconds. If this interval is exceeded with no confirmation of successful completion received by the Control Task, the status of the process is set to `UNKNOWN`.

**Examples**

The following example is 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 `IOEDFSKN`. 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`, and the local code page is specified as `LANG=En_US.IBM-1047` (which is the default). An LE runtime option is specified: `HEAPP(ON)`. Parameters for the `dfskern` process follow the `/`. The `>` symbol is a redirection character which indicates that the output is redirected to the DD name that follows. In this example, the redirection of the `STDERR` to `STDOUT` of the process (`dfskern`) is specified by: `>DD:dfskern 2>&1`.

```
RESTART=300 TIMEOUT=300
```

**Note:** The previous example should be entered on one line even though multiple lines are used in this example.

**Implementation Specifics**

The `ioepdcf` file is stored as an EBCDIC file in HFS.

This file is optional. If it does not exist, the following default values are used:
ioepDCF

DFSKERN CONFIGURED=Y LMD=DFSKERN ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/dfskern')/ admingroup
subsys/dce/dfs-admin>DD:DFSKERN2>&1" RESTART=300 TIMEOUT=300
EXPORT CONFIGURED=I LMD=IOEDFSXP ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/dfsexport')/ all -verbose
>DD:EXPORT2>&1" RESTART=300 TIMEOUT=300
UNEXPORT CONFIGURED=M LMD=IOEDFSXP ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/dfsexport')/ -detach -all -ver
>DD:UNEXPORT2>&1" RESTART=300 TIMEOUT=300
BOSERVER CONFIGURED=M LMD=BOSERVER ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/boserver')/ >DD:BOSERVER 2>&1"
RESTART=300 TIMEOUT=300
BUTC01 CONFIGURED=M LMD=BUTC01 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc01')/ -tcid 0 >DD:BUTC01 2>&1"
RESTART=300 TIMEOUT=300
BUTC02 CONFIGURED=M LMD=BUTC02 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc02')/ -tcid 1 >DD:BUTC02 2>&1"
RESTART=300 TIMEOUT=300
BUTC03 CONFIGURED=M LMD=BUTC03 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc03')/ -tcid 2 >DD:BUTC03 2>&1"
RESTART=300 TIMEOUT=300
BUTC04 CONFIGURED=M LMD=BUTC04 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc04')/ -tcid 3 >DD:BUTC04 2>&1"
RESTART=300 TIMEOUT=300
BUTC05 CONFIGURED=M LMD=BUTC05 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc05')/ -tcid 4 >DD:BUTC05 2>&1"
RESTART=300 TIMEOUT=300
BUTC06 CONFIGURED=M LMD=BUTC06 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc06')/ -tcid 5 >DD:BUTC06 2>&1"
RESTART=300 TIMEOUT=300
BUTC07 CONFIGURED=M LMD=BUTC07 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc07')/ -tcid 6 >DD:BUTC07 2>&1"
RESTART=300 TIMEOUT=300
BUTC08 CONFIGURED=M LMD=BUTC08 ARG="ENVAR('_EUV_HOME=/opt/dfslocal/home/butc08')/ -tcid 7 >DD:BUTC08 2>&1"
RESTART=300 TIMEOUT=300
rfstab

Purpose
See "Attributes file (rfstab)" on page 84.
**smbidmap**

**smbidmap**

**Purpose**
Maps an SMB user ID to a z/OS user ID. The mapping is used to determine the SMB user’s corresponding z/OS user ID. This determines access permissions for shared HFS and RFS directories and files and owners for print requests sent to shared printers. If no `smbidmap` file is specified or it does not exist, then the SMB user ID is mapped to the default user ID (specified in the `_IOE_MVS_DFSDFLT` environment variable of `dfskern`). If no default user ID is specified, the request is denied.

**Usage**
The `smbidmap` file is a text file that the administrator creates and maintains. This can be created as a z/OS UNIX file, a sequential data set, or a member of a partitioned data set. The location of this file is specified in the `_IOE_SMB_IDMAP` environment variable of `dfskern`.

The `smbidmap` file contains one or more identity mapping declarations and has the following general format:

```
SMB-user-ID1
z/OS-user-ID1
...
```

This format illustrates an SMB user ID mapping entry. Each entry has two elements: SMB user ID and z/OS user ID. A blank line is required between entries. The following explains each element in an SMB user ID mapping entry:

- **SMB-user-ID or Domain/SMB-user-ID or Workgroup/SMB-user-ID**
  Specifies the client’s SMB identity. This can either be a simple SMB user ID (when you do not care what the domain of the SMB user ID is) or a fully qualified name (for clients within and outside the domain/workgroup). The SMB user ID can be up to 20 characters in length. A Domain/Workgroup name can be up to 15 characters in length.
  - **SMB-user-ID** is assumed to be in any domain.
  - **Domain/SMB-user-ID** is assumed to be in the specified domain.
  - **Workgroup/SMB-user-ID** is assumed to be in the specified workgroup.

- **z/OS-user-ID**
  Is the z/OS user ID of the client. All potential SMB clients must have z/OS user IDs on the system where the DFS server is running.

One of the following entries can also be in `smbidmap`:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>* or Domain/* =</td>
<td>If = is used in the second line, the z/OS userid is used exactly as is.</td>
</tr>
<tr>
<td>* or Domain/* &lt;</td>
<td>If &lt; is used in the second line, the z/OS userid is used in lowercase.</td>
</tr>
<tr>
<td>* or Domain/* &gt;</td>
<td>If &gt; is used in the second line, the z/OS userid is used in uppercase.</td>
</tr>
</tbody>
</table>

**Note:** If a Domain/* entry is used (the Domain from the SMB matches, but no Domain/SMB-user-ID was found in the smbidmap file), any SMB-user-ID from that Domain is used as the z/OS user ID.

If no z/OS user ID can be determined (because the SMB user ID cannot be found in the smbidmap file) and the SMB user ID (in the SMB) is eight characters or less, the SMB user ID (passed in the SMB) is used as the z/OS user ID. If you want, this can be the only entry in `smbidmap`. This is also relevant for use with `&USERID` in the smbtab.
The order of search in the smbidmap file for a matching domain and SMB-user-ID is:

- Domain/userid
- Domain/*
- userid
- *

See “Working with automounted file systems and home directories:” on page 44 and “Recommended technique for PC user access to automounted home directories” on page 46 for information about &USERID.

Examples

In the following example, the SMB user ID smith in domain1 is mapped to the z/OS user ID CMSMITH and SMB user ID jones (in any domain) is mapped to the z/OS user ID TSJONES.

domain1/smith
  CMSMITH
jones
  TSJONES

Note: You can put comments in smbidmap file. Use a semicolon (;) to begin the comment. (The semicolon is not valid as an SMB-user-ID and as a z/OS-user-ID.) Comments can begin in any column. After the semicolon, all data on the line is ignored. The following are some examples of comment usage:
  ; this whole line is a comment
  usera ; the userid is usera and the rest of the line is a comment
  abc#def; the userid is abc#def and the rest of the line is a comment

Implementation Specifics

The smbidmap file is stored as an EBCDIC file in HFS, or a sequential data set or a member of a PDS.
**smbtab**

**Purpose**
Specifies HFS and RFS shared directories and shared printers being made available to PC clients.

**Usage**
The `smbtab` file includes information about each shared directory and each shared printer that can be made available to PC clients. It resides in the directory `/opt/dfslocal/var/dfs`.

The file contains a one-line entry for each shared directory or shared printer. Each entry in the file must appear on its own line.

The shared directory fields in the following list must appear for each entry; they must appear in the order listed, and each field must be separated by at least one space or tab.

- **Device Name**  For shared directories, the device name of the HFS or RFS file system that contains the root of the directory path name being shared; for example, `/dev/ufs2`. This must match the device name of the file system in the `dfstab`.

- **Share name**  The name to be associated with the HFS or RFS directory being shared. A share name can contain numbers (0-9), letters (A-Z), and the following special characters: $ % ' @ ~ ` ! ( ) ^ # & . To ensure that a client can connect to an SMB share, you should limit yourself to these characters. A share name can be up to 12 characters.

- **Device type**  The device type identifier for the type of device housing the share. For HFS and RFS directories this must be `ufs`. Enter the identifier in all lowercase letters.

- **Share description**  The text description of the share. It can be up to 40 characters and is within quotation marks; for example, “Department HFS files”.

- **Shared directories permissions**  For shared directories, specifies whether the share is limited to read-only access or whether read-write access is permitted. Read-only access is specified as `r/o`. Read-write access is specified as `r/w`. In addition, permissions used during a create of a file or directory can be specified. They are specified directly after `r/w`. For example, you might specify `r/w,f=700,d=755`. These permissions overide the global create permissions (`_IOE_SMB_DIR_PERMS` and `_IOE_SMB_FILE_PERMS` environment variables in `dfskern`) for this shared directory. Create permissions cannot be specified for `r/o` access.

- **Maximum users**  For shared directories, the maximum number of users that can be connected to a share name. It can be a number in the range of 0 - 4294967295. 0 means that there is no limit to the number of users.

- **Directory path name**  For shared directories, specifies the path name of the HFS or RFS directory (relative to the root of the file system referred to by the **Device name** that this share represents. For HFS, the directory must be in a locally mounted HFS file system. For RFS, the directory must be the root of the RFS file system (/) or a directory (that is, PDS or PDSE) within the RFS file system. The file system must be exported (see "dfstab" on page 95). The directory path name can be up to 1024 characters. It must be within quotation marks if it contains embedded blanks. For HFS, you might want to export HFS file systems below this directory in order to permit all path names below this directory to be accessible by this share or you might want to use dynamic export. See "Dynamic export for HFS" on page 42 for information about using the dynamic export capability of the SMB server.
If you are using dynamic export, you can specify a special keyword in the Directory path name field. The keyword is &USERID. It represents the PC user's z/OS user ID. It permits a single smbtab entry to mean a different directory depending on which user connects to the shared directory. See "Recommended technique for PC user access to automounted home directories" on page 46 for more information about the usage of the &USERID keyword.

The shared printer fields in the following list must appear for each entry; they must appear in the order listed, and each field must be separated by at least one space or tab.

**Device name**  
For shared printers, the device name of the printer being shared; for example, /dev/prt1.

**Share name**  
The name to be associated with the printer queue being shared. A share name can contain numbers (0-9), letters (A-Z), and the following special characters: $ % ' _ @ ~ ` ! ( ) ^ # & . To ensure that a client can connect to an SMB share, you should limit yourself to these characters. A share name can be up to 12 characters.

**Device Type**  
The device type identifier for the type of device housing the share. For printers this must be prt. Enter the identifier in all lowercase letters.

**Share description**  
The text description of the share. It can be up to 40 characters and within quotation marks. For example, “Department printer”.

**Printer definition name**  
For shared printers, specifies the name of the printer definition. The printer definition name is created during the definition of the printer. See z/OS Infoprint Server Operation and Administration.

**Printer type**  
For shared printers, specifies the type of printer. This can be the name of a printer type supplied by Windows. See z/OS Infoprint Server Operation and Administration.

When the dfsshare command is executed, it reads the smbtab file to verify that each directory or printer to be shared is listed in the file. A directory or printer must have an entry in the smbtab file if it is being shared. If a directory or printer is currently shared, it is not shared again. The smbtab file is also read and shared directories and shared printers are created during server initialization.

**Examples**

The following smbtab file specifies an HFS directory, an RFS directory, and one printer should be shared.

```
/dev/ufs2 myshare ufs "My share description" r/w 100 /ghi
/dev/ufs3 rfsshare1 ufs "USERA.PCDSNS z/OS data sets" r/o 50 /
/dev/prt1 myprt prt "Department printer" printname1 "Generic / Text Only"
```

**Note:** You can put comments in smbtab file. Comments start with # in column 1.

**Implementation Specifics**

The smbtab file is stored as an EBCDIC file in HFS. An example smbtab file can be found in /opt/dfsglobal/examples.

**Related Information**

Commands:
- dfsexport
- dfsshare

Files:
- devtab
- dfstab
smbtab
Chapter 14. SMB commands

This section provides an alphabetical listing of all relevant SMB commands.

These commands are issued from the OMVS environment. OMVS users that are not using DCE should have the following line in their HFS .profile file in their home directory:

```bash
export _EUV_AUTOLOG=N0
```

Alternatively, if no OMVS users are using DCE, then the above line may be placed in the /etc/profile file. This causes it to take effect for all OMVS users.
dfsexport

Purpose
An OMVS command that exports (or unexports) HFS and RFS file systems. This makes underlying file
systems available so that directories contained in them may be shared (see "dfsshare" on page 113).

Format
dfsexport [[-all | -aggregate name | -filesystem name]] [-detach] [-verbose] [-force] [-type]
[-level] [-help]

Options
-all Specifies that all HFS and RFS file systems listed in the /opt/dftslocal/var/dfs/dfstab file
are being exported. Use this option or use -filesystem; omit both options to list all file
systems currently exported.

-aggregate name | -filesystem name
Specifies the file system name of the HFS or RFS file system to be exported. This name
is specified in the first or second field of the entry for the HFS or RFS file system in the
dfstab file. Use this option or use -all; omit both options to list all HFS file systems
currently exported.

Note: If you enter this command in TSO/E, the name parameter must be within quotation
marks (""). Also, the -filesystem option can also be specified as -aggregate as
shown in z/OS Distributed File Service DFS Administration

-detach Used with the -all option, specifies that the file system(s) indicated with the command’s
other options are to be detached (no longer exported), making the corresponding shares
unavailable to Windows clients. Use -all or -filesystem with this option indicating the
exports are to be detached.

Use the -detach option only when no users are accessing data on the HFS or RFS file
systems to be detached or when a serious emergency warrants its use. When the -detach
option is used, the command breaks all oplocks for data on a corresponding share before
the file system is detached.

To permanently detach a file system, it must also be removed from the dfstab file.
Otherwise, the dfsexport command exports file systems the next time it is run.

-force The -force option on the dfsexport command only applies to DCE DFS protocols.

-level Prints the level of the dfsexport command. This is useful when you are diagnosing a
problem. All other valid options specified with this option are ignored.

-type Specifies that only aggregates or file systems whose file system types match the type
specified with this option are to be exported. The type can be specified as lfs to export
only DCE Local File System aggregates, or it can be specified as ufs to export only file
systems. The type of each aggregate or file system appears in the third field of the entry
for the device in the dfstab file. The type must be specified in lowercase letters (as it
appears in the dfstab file). The SMB server only supports -type ufs.

Use this option only with the -all option; it is ignored if it is used without the -all option. If it
is omitted and -all is used, the command exports both lfs and ufs devices.

-verbose Directs the command to report on its actions as it executes.

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

The `dfsexport` command exports underlying HFS and RFS file systems so that directories may be shared from z/OS to PC clients. Directories and printers that are shared are still available to other z/OS users. Issue this command with no options to list the file systems already exported. The binary file for the `dfsexport` command resides in `/opt/dfsglobal/bin/dfsexport`.

The `dfsexport` command exports HFS and RFS file systems based on the values provided with its options. If the `-all` option is provided, the command shares all file systems listed in the `/opt/dfslocal/var/dfs/dfstab` file. If the `-filesystem` option is provided, it exports only the file system whose name is specified with the option. The specified name must be listed in the `dfstab` file.

When `dfsexport` executes, it reads the `dfstab` file on the local disk of the machine to determine the file systems available to be exported. A file system must have an entry in the `dfstab` file if it is being exported. Because this command reads the `dfstab` file, information supplied with its options must match exactly the information for a file system specified in that file.

The `dfsexport` command reads a list of all currently exported file systems that is maintained in the SMB server of the local machine. The command does not export a file system that is currently exported.

Issuing the `dfsexport` command with no options lists the file systems currently exported from the local file server.

The `dfsexport` command is normally executed automatically when the SMB server is started. This is controlled by the `export` entry of the `ioepdcf` file, see `ioepdcf` on page 100 for more information. When export is enabled, all indicated HFS and RFS file systems listed in the `dfstab` file are exported and all HFS and RFS directories listed in `smbtab` that are contained in those exported file systems are shared.

Prior to using this command to export a file system for the first time, perform the following:

1. For HFS file systems, ensure that the file system is mounted locally; it can contain data or it can be empty. The SMB server must be running on the z/OS system that owns the HFS file system. (RFS file systems cannot and need not be locally mounted.)
2. Create an entry for the file system in the `devtab` file on the z/OS system on which the file system resides. This entry maps the minor device number to the HFS or RFS file system data set you wish to export. It also allows you to (optionally) specify whether character data translation should occur when the data is read or written by PC clients. For further information, see `devtab` on page 91.
3. Create an entry for the file system in the `dfstab` on the machine on which the partition resides. Use a unique file system name and ID number and a unique fileset ID number in the appropriate fields of the entry for the file system.

Before exporting a file system, also make sure that no local users have files open on the file system. The SMB server cannot effectively synchronize file access between users who opened files from a file system before the file system was exported and users who open files from the file system after the file system is exported because only the latter have tokens.

**Privilege required**

The issuer must be logged in as `root` on the local machine. On z/OS, `root` refers to a user with a UID = 0 or BPX.SUPERUSER authority.

**Cautions**

Before using the `-detach` option with this command, make sure no users are currently accessing data from file systems to be detached. The command does not verify that a device is not in use before removing it from the namespace. A user who is accessing data housed on a file system when it is detached is not able to save the data back to the device. Any attempt to perform an action that involves a detached file system elicits a message reporting that the device is Unknown.
dfsexport

Examples
On SMB, the dfsexport process, by default, is started automatically by the DFS control task program, dfscntl. dfscntl and its child processes are, in turn, controlled in SMB by the z/OS system command, MODIFY.

You can also use the MODIFY system command to export and unexport file systems SMB.

The following command causes the dfsexport program to run with the parameters specified in the ioepdcf configuration file. By default, this command exports all of the file systems that have entries in the machine's dfstab file:

modify dfs,start export

The following command causes the dfsexport program to run with the parameters specified in the ioepdcf configuration file. In this example, the command unexports all of the file systems that have entries in the dfstab file:

modify dfs,start unexport

The following command exports the file system whose device name (as it appears in the dfstab file) is /dev/ufs1:

# dfsexport /dev/ufs1

Implementation specifics
The dfsexport process on SMB, by default, is started automatically by the DFS control task program, dfscntl. The dfsexport process on DFS runs under the control of the DFS control task, dfscntl. dfscntl and its child processes are controlled on DFS by the MODIFY system command. For further information, see "modify dfs processes" on page 78.

After dfsexport exports the file system(s) specified, it creates any shared directories defined in smbtab that refer to those file systems. No shared printers are created. When detach is specified, dfsexport unshares any shared directories that refer to file systems being unexported before unexporting the file systems.

This command may be issued from TSO/E or a z/OS shell.

Note: In order to export an HFS file system, that file system must be owned by the system that the SMB server is running on. When export is attempted on a (sysplex) shared file system that is owned by a system other than the one that the SMB server is running on, the SMB server attempts to move the ownership of a shared HFS file system to the system that the SMB server is running on (when the _IOE_MOVE_SHARED_FILESITEM environment variable is ON in the dfskern process). If this is unsuccessful (when, for example, the file system is being exported by another SMB server on the other system), the SMB server will not be able to export that file system and it will not be available to PC clients.

Related information
Files:
  devtab
dfstab
smbtab
dfsshare

Purpose
An OMVS command that shares (or unshares) HFS and RFS directories or printers with SMB clients.

Format
dfsshare [(-all | -share name)] [-type name] [-detach] [-verbose] [-help]

Options
-all Specifies that all HFS and RFS directories and printers listed in the /opt/dfslocal/var/dfs/smbtab file are shared with SMB clients. Use the -type option with this option to export only HFS and RFS files or only Infoprint Server printers. Use this option or use -share; omit both options to list all shared files and shared printers currently shared with SMB clients.

-share name Specifies the share name of the HFS or RFS shared directory or shared printer. This name is specified in the second field of the entry for the HFS directory or printer in the smbtab file. Use this option or use -all; omit both options to list all HFS directories and printers currently shared with SMB clients.

Note: Some share names cannot be specified on OMVS commands or must be enclosed in single quotation marks. A share name that begins with dollar ($) must be enclosed in single quotation marks.

-type name Specifies that only shared directories or shared printers whose type matches the type specified with this option are shared. The type can be specified as ufs to share only HFS and RFS directories, or it can be specified as prt to share only printers. The type of each share appears in the third field of the entry for the device in the smbtab file.

Use this option only with the -all option; it is ignored if it is used without the -all option. If it is omitted and -all is used, the command shares both ufs and prt types.

-detach Used with the -all option, specifies that the shared directories and shared printers indicated with the command’s other options are detached (no longer shared), making them unavailable to SMB clients. Use -all or -share with this option to indicate the shares to be detached; use the -type option with -all to detach only one type of share.

Use the -detach option only when no users are accessing data on the HFS or RFS shared directories to be detached or when a serious emergency warrants its use.

To permanently detach a share, it must also be removed from the smbtab file. Otherwise, the dfsshare command shares the directories and printers the next time it is run (provided the directories or printers are included in the specification for the types to be shared).

-verbose Directs the command to report on its actions as it executes.

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

Usage
The dfsshare command shares HFS and RFS directories and printers from z/OS to Windows clients. Directories and printers that are shared are still available to other z/OS users. Issue this command with no options to list the directories and printers already shared. The binary file for the dfsshare command resides in /opt/dfsglobal/bin/dfsshare.

The dfsshare command shares HFS and RFS directories, printers, or both based on the values provided with its options. If the -all option is provided, the command shares all directories and printers listed in the
dfsshare

/opt/dfslocal/var/dfs/smbtab file. If the -share option is provided, it shares only the directory or printer whose share name is specified with the option. The specified name must be listed in the smbtab file.

The -type option can be used with the -all option to indicate that only directories or only printers are shared. If ufs is provided with the -type option, the command exports only directories; if prt is provided with the -type option, it exports only printers. If the -type option is used, the -all option must also be included; otherwise, the -type option is ignored.

When dfsshare executes, it reads the smbtab file to determine the directories and printers available to be shared. A directory or printer must have an entry in the smbtab file if it is shared. This command also invokes dfsexport to ensure that the underlying HFS or RFS file system that contains the shared directory is exported. Therefore, the corresponding HFS or RFS file system information must exist in dfstab and devtab for the shared directories that are being created. If there are additional file systems mounted below the shared directory that you want to allow PC users to access, those file systems must be exported also. The dfsexport command can be used for this purpose.

The dfsshare command reads a list of all currently shared directories and printers that is maintained in the SMB Server of the local machine. The command does not share a directory or printer that is currently shared. If you want to change the parameters for a printer or an HFS or RFS directory that is already shared, you must first detach (by using the dfsshare -share name -detach command) and then reshare (by using the dfsshare -share name command) for the new parameters to take effect.

Issuing the dfsshare command with no options lists the directories and printers currently shared to SMB clients.

The dfsshare command is automatically executed when the SMB server is started (with SMB processing enabled). This is controlled by the export entry of the ioepdcf file, see [ioepdcf on page 100] for more information. When export is enabled, all indicated HFS and RFS directories listed in the smbtab file are shared and all HFS and RFS file systems listed in dfstab and devtab are exported. In addition, all printers listed in the smbtab are shared (if the Infoprint Server is enabled) regardless of the export entry in the ioepdcf file.

Privilege required
The issuer must be logged in as root on the local machine. On z/OS, root refers to a user with a UID = 0 or BPX.SUPERUSER authority.

Implementation specifics
The dfsshare command exports the file system that is referred to by the shared directory (if it is not already exported) before creating the shared directory. When -detach is specified, the shared directory is unshared but no file systems are unexported.

The dfsshare command can only be issued from the z/OS shell. It is not supported as a TSO/E command.

Note: In order to export an HFS file system, that file system must be owned by the system that the SMB server is running on. When export is attempted on a (sysplex) shared file system that is owned by a system other than the one that the SMB server is running on, the SMB server attempts to move the ownership of a shared HFS file system to the system that the SMB server is running on (when the _IOE_MOVE_SHARED_FILESYSTEM environment variable is ON in the dfskern process). If this is unsuccessful (when, for example, the file system is being exported by another SMB server on the other system), the SMB server will not be able to export that file system and it will not be available to PC clients.
Related information
Files:
  devtab
dfstab
smbtab

Command:
  dfsexport
The \texttt{dfssyntax} command is an SMB command that checks the syntax of an environment variable file.

\textbf{Format}
\texttt{dfssyntax [filename]}

\textbf{Options}
\textit{filename} \hspace{1em} The name of the environment variable file that will be checked for syntax errors.

\textbf{Usage}
The \texttt{dfssyntax} command performs automatic syntax checking on an environment variable. If no other environment variable file is specified, it checks the default file \texttt{/opt/dfslocal/home/dfsckern/envar}. The \texttt{dfssyntax} command writes messages to the console explaining the line number where the syntax mistake is located and any other problems it finds. Examples of syntax errors include blank lines, wrong environment variable name, wrong value for environment variable, trailing blanks, and others.

\textbf{Privilege required}
No privileges are required.

\textbf{Related information}
Files:
- \texttt{envar}

Command:
- \texttt{dfssyntax}
smbpw

Purpose
An OMVS command that stores a z/OS user’s SMB password in the RACF database for use with SMB encrypted password support. Encrypted password support is used when the SMB server is configured to support encrypted passwords by using the _IOE_SMB_CLEAR_PW environment variable in the dfskern process.

Format
smbpw [-pw1 newpassword -pw2 newpassword] [-help]

Options
-pw1 newpassword
Specifies the SMB password to be stored in the current z/OS user’s RACF DCE segment. The SMB password can be up to 14 characters. If the user issues the smbpw command without providing the password, smbpw prompts the user for the SMB password.

-pw2 newpassword
Specifies the same password again. This is to verify that the password was entered correctly.

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

Usage
During SMB login processing, when encrypted password processing is configured, the SMB server retrieves the user’s SMB password from the user’s RACF DCE segment. The smbpw command is used when a user needs to initially store or change the SMB password in their RACF DCE segment. The password entered is case sensitive. That is, it is stored in the RACF DCE segment exactly as the user typed it. In general, the password should be entered in lower case. It is folded to upper case by the SMB server, if necessary. This command requires that the changed password is entered twice.

The user may provide none or both occurrences of the password on the command line. If none is supplied, the system prompts for the changed password and then prompts for the changed password again. These passwords are verified to match.

The smbpw command does not verify that the password specified is the same as your Windows password. If you enter an incorrect password twice, the password is saved. If you determine that you have stored an incorrect password, use smbpw again, supplying the correct password.

Note: If you cancel out of the smbpw command and are left in hidden mode, you can reset your terminal to make input invisible by usage of the stty -sane command. See z/OS UNIX System Services Command Reference for additional information on the stty command.

If you are using SMB encrypted passwords and DCE single sign-on, your SMB password and your DCE password must be the same since both of these come from the RACF DCE segment. However, if the DCE Security Server is running on the same system as the DCE user, then the DCE single sign-on processing does not require the DCE password to be stored in RACF. So, in this case, the SMB password and the DCE password can be different.

Privilege required
No privileges are required.
**Examples**

In this example, `mynewpw` is the SMB password to be stored in the RACF DCE segment:

```
$ smbpw mynewpw mynewpw
```

The following example is the same as above except that the user is prompted for the new password:

```
TEST1:/home/test1/> smbpw
IOEW16057I Enter Password: mynewpw
IOEW16058I reenter Password: mynewpw
IOEW16055I Your password has been updated.
```

**Note:** The passwords are not displayed when they are entered at the prompt. If you use a shell metacharacter (for example, `$`) in your password and you specify it on the command line (as opposed to allowing `smbpw` to prompt for the password), you must use the escape character to keep the shell from interpreting it.

For example, a password of `p$w` must be specified: `smbpw p\$w p\$w`


**Implementation specifics**

The `smbpw` command can only be issued from the z/OS shell. It is not supported as a TSO/E command. The user issuing the `smbpw` command must have a RACF DCE segment.

z/OS users that do not use DCE should put the following line in their `.profile` file in their home directory:

```
export _EUV_AUTOLOG=NO
```

Alternatively, if no z/OS users are using DCE, the above line may be placed in the `/etc/profile` file.
Chapter 15. Tuning and debugging guidelines

This information describes the SMB tuning and debugging guidelines. The examples presented here are for illustrative purposes only—it is normal for the output of some reports to wrap.

Introduction

This information is provided for administrators of the z/OS Distributed File Service SMB server.

SMB, like most file system servers, is dependent on many factors. If problems arise, SMB provides diagnosis information to help determine sources of bottlenecks or the affect of a hardware or software change on SMB performance. SMB also provides environment variables and the output of QUERY commands. Note that the default settings are sufficient for most installations.

Figure 12. SMB server components
Tuning considerations and recommendations

z/OS Distributed File Service SMB server provides a **QUERY** command that provides information that is useful for gauging SMB performance and determining the affect of a change to a tuning option.

In this document, tuning of the SMB server means:

- Determining the size of thread pools responsible for handling work
- Determining the amount of storage to reserve for various types of caches

Many of the SMB default settings or values work for most installations, but there are some defaults that might not be sufficient for larger installations. The most important tuning options are described here.

**Threading**

If the SMB server 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 response time, therefore ensuring that the SMB server has enough defined processing threads. Distributed File Service default values for the number of processing threads is sufficient for smaller installations, but might be inadequate for larger production use. Each processing thread requires a certain amount of storage, therefore increasing the number of threads increases the storage requirements for the SMB server.

The following section describes the tuning options that control threading.

**_IOE_SMB_MAIN_POOL**

This dfskern environment variable controls the number of threads available to process incoming SMBs for the 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 data set open or close requests. This is important for customers using RFS on a larger scale.

**Caching**

The following section describes tuning options for disk caching:

**_IOE_VM_CACHE_SIZE**

This environment variable indicates how much file data to cache for SMB servers using disk caching. This is applicable to all server workloads, SMB, and all file systems such as zFS and HFS.

**bufhigh**

This rfstab (attribute file) site attribute indicates how much data to store in the RFS data set cache for RFS workloads. High RFS file read/write activity might require an increase in the size of this cache. See Table 6 on page 89.

**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 SMB **QUERY** commands can possibly point out a bottleneck in a particular disk or the overall network. I/O response times reported by SMB might show the affect of a bottleneck or the affect of the removal of a bottleneck.
QUERY command

SMB provides a QUERY command that is useful for debugging problems and capturing performance information. The output of the QUERY command goes to the system log and the job output. The syntax of the command is:

modify dfs,send dfskern,query,<REPORT>

Reports available for SMB

The following SMB query reports are in alphabetical order. References to z/OS UNIX physical file system (PFS) in this section includes zFS, HFS, and TFS.

ADAPTERS    This report shows the storage usage and performance of the SMB file server file status, directory contents, file name lookup, and file security cache used by the PFS and RFS. It also shows the number of calls made by the SMB protocol to the PFS and RFS and the average response time of each call.

AFS4INT     This report shows the number and type of remote DFS/DCE 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. This report applies only to DFS/DCE.

ALL         This report shows every report described in this section.

FILESETS    This report queries the fileset information. It includes the fileset name (file system), aggregate ID, flag field (A = attached, M = mounted, E = exported), number of SMB’s, and RPC’s (normally zero and local).

LEVEL       This report queries the SMB service level. It includes the module prefix, z/OS version and release identifier, service level, and FMID.

modify dfs,send dfskern,query,level
IOEN00195I DFKERN: z/OS     Distributed File Service
Version 00.00.00 Service Level 0000000 0000000
Created on DAY DATE TT:TT:TT TIME ZONE YEAR.
IOEN00119I DFKERN: SEND command - QUERY,LEVEL completed successfully.

LFS         This report shows the number of calls made by the SMB protocols to the local file system (LFS) along with the performance of the 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.

LOCKING     This report shows the SMB 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.

RFS         This report shows the access method counts and response times for the RFS physical file system.

SERVTHREADS This report queries the running service threads. It includes the thread ID (TCB), thread start time, and thread run time.

modify dfs,send dfskern,query,ervthreads

SMB Service Threads

-------------------------------------

Coordinate Universal Time (UTC) is Mon Aug 07, 2006 13:09:54.452857

TCB                        Thread Start Time (UTC)                        Run Time (sec)
-----------------------------------------------------------------------------------------------
007C9968  Mon Aug 07, 2006 13:09:54.215317                          0.237540
007C9E88  Mon Aug 07, 2006 13:08:55.643607                          58.809250

IOEN00119I DFKERN: SEND command - QUERY,SERVTHREADS completed successfully.

SESSFS       This report queries the SMB file systems per session. It includes the computer name,
number of open files, socket number, IP address, SMB userid, username, number of SMB requests, and file system names with open files for each session.

f dfs,send dfskern,query,seessfs

DFS/SMB Filesystems Per Session
---------------------------------  
Number of active SMB clients = 4  

<table>
<thead>
<tr>
<th>Computer Name</th>
<th>Open Files</th>
<th>Socket</th>
<th>IP Address</th>
<th>Uid Username</th>
<th>Requests</th>
<th>Filesystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMBCLIENT2</td>
<td>0</td>
<td>24</td>
<td>127.0.0.1</td>
<td>000E TTT2</td>
<td>1</td>
<td>ABCDEFG.PRIVATE.IAM.TESTSHR1</td>
</tr>
<tr>
<td>12345MBCLI4</td>
<td>1</td>
<td>26</td>
<td>127.0.0.1</td>
<td>0014 TTT1</td>
<td>517</td>
<td>ABCDEFG.PRIVATE.IAM.TESTSHR1</td>
</tr>
<tr>
<td>12345MBCLI4</td>
<td>4</td>
<td>10</td>
<td>127.0.0.1</td>
<td>0001 JUS1</td>
<td>398228</td>
<td>ABCDEFG.PRIVATE.IAM.TESTSHR1</td>
</tr>
<tr>
<td>SMBCCLIENT1</td>
<td>0</td>
<td>23</td>
<td>127.0.0.1</td>
<td>0018 JUS2</td>
<td>333</td>
<td>ABCDEFG.PRIVATE.IAM.TESTSHR1</td>
</tr>
</tbody>
</table>

IOEN00119I DFSKERN: SEND command - QUERY,SESSFS completed successfully.

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.

SMBINT  For the SMB file server, this report shows the number and type of remote Windows client SMB file requests made to the server, the average file server response and CPU times to process those requests, and the TCP/IP send and receive times that can be used to determine the network overhead for file server response times.

SMBMAP  This report queries the SMB map table and displays the current mappings of SMB user to z/OS user.

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.

SMBSESS  This report queries the SMB sessions. It includes the computer name, number of open files, socket number, IP address, SMB userid, username, and number of SMB requests for each session.

f dfs,send dfskern,query,smbseess

DFS/SMB Session Information
---------------------------  
Number of active SMB clients = 4  

<table>
<thead>
<tr>
<th>Computer Name</th>
<th>Open Files</th>
<th>Socket</th>
<th>IP Address</th>
<th>Uid Username</th>
<th>Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMBCCLIENT1ABCD</td>
<td>2</td>
<td>10</td>
<td>127.0.0.1</td>
<td>00CD TTT1</td>
<td>65</td>
</tr>
<tr>
<td>12345MBCLI4</td>
<td>2</td>
<td>22</td>
<td>127.0.0.1</td>
<td>005E TTT5</td>
<td>2191</td>
</tr>
<tr>
<td>12345MBCLI4</td>
<td>0</td>
<td>20</td>
<td>127.0.0.1</td>
<td>0186 STU1</td>
<td>25766</td>
</tr>
<tr>
<td>JUSER</td>
<td>1</td>
<td>23</td>
<td>127.0.0.1</td>
<td>018F TSU2</td>
<td>73</td>
</tr>
</tbody>
</table>

IOEN00119I DFSKERN: SEND command - QUERY,SESSFS completed successfully.

SMBUSERIDSHARES  This report queries the automounted file systems. It includes the auto-mounted file system name, the z/OS userid that was used to create the directory, and the number of users accessing the file system.

f dfs,send dfskern,query,smbuseridshares

SMB Dynamic UserID Shares
--------------------------  
Name/UserID  Users  
------------  ------  
MYHOME/suimgko  1  

IOEN00119I DFSKERN: SEND command - QUERY,SMBUSERIDSHARES completed successfully.

================================================

122  z/OS V1R11.0 Distributed File Service SMB Administration
**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.

**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.

**THREADS**
This report queries currently running threads. It includes the thread ID (TCB), the stack the thread is running on, the stack size, the routine that is running, the state of the thread, and the offset within the routine.

**TKM**
This report shows the token manager statistics for the SMB file server. It includes the size of the token manager cache, the token manager request rates, the number of token revoke RPC calls (RPC calls apply to the DFS client only) to clients, and the average response time for a token revoke RPC call.

**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.

**VNODE**
This report queries the vnodes. It includes the maximum number of vnodes, number of vnodes in use, number of root vnodes, and number of directory vnodes. These fields are not mutually exclusive.

```
modify dfs,send dfskern,query,vnode
```

```
VNODE Statistics
-------------------
Max vnodes: 6144
Vnodes in use: 16
Root vnodes: 12
Directory vnodes: 17
-------------------
```

```
IOEN00119I DFSKERN: SEND command - QUERY,VNODE completed successfully.
```

**VOLREG**
This report queries the file system information in the volume registry. It includes the volume ID to file system name association and one the following status indicators:
- incorrect association
- exported
- not exported

---

**RESET command**
The **RESET** command clears the statistics monitored by the SMB file server. The **QUERY** command always displays the statistics since the server startup or since the last **RESET**. This allows an administrator to query performance for a given time period, such as during peak usage times. The syntax for the server is:

```
F DFS,SEND DFSKERN,RESET,<REPORT>
```

where `<REPORT>` is the name of the following report:

- ADAPTERS
- AFS4INT
- FILESETS
- LFS
- LOCKING
- RFS
The following example clears the statistics kept for the STORAGE report:

```
F DFS,SEND DFSKERN,RESET,STORAGE
```

### TRACE command

**Restriction:** You should only change tracing options under the advisement of your IBM service representative. An important aspect of SMB problem determination is its tracing capability. SMB has an internal (wrap around) trace table that is always tracing certain events. The initial size of the trace table is controlled by the _IOE_TRACE_TABLE_SIZE environment variable, but can be adjusted using the `tsize` subcommand on the `TRACE` command. For complete details, see "modify dfs processes" on page 78.

### SMB hang detection

**Restriction:** Hang detection environment variables should only be changed under the advisement of your IBM service representative.

The SMB hang detector monitors the current location of the various tasks that are processing in SMB. At intervals, set through the _IOE_HANG_DETECT_INTERVAL environment variable, the hang detector scans the current user requests called into SMB. When the hang detector determines that a thread has remained in the same location for a predefined period of time, it flags the task as a potential hang and produces a dump to preserve problem data.

Occasionally, SMB hang detection might produce false hang dumps and an actual hang might go undetected. If dfskern issued a hang detection dump, or you suspect that dfskern is hung for some other reason (users are not getting a response), you can use the following procedure to determine if you have a hung thread.

1. In dfskern, issue the `f dfs,send dfskern,query,threads` command at 30 second intervals for two to three minutes and compare the output.
2. Compare tasks that remain in the same status (tcb,stack,routine,state,offset) across several different instances of query,threads output. If an entry is identical in each of query,threads output, there is presumably a hang.

The example in Figure 13 on page 125 shows some typical threads in wait for work states that are common to see persist across multiple instances of query,threads output:
The example in Figure 14 shows an entry for task 6856D8 that indicates a hang in dfskern when its status persists across multiple instances of query,threads output:

<table>
<thead>
<tr>
<th>TCB</th>
<th>Stack</th>
<th>Size</th>
<th>Routine</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>006D0370</td>
<td>1440FC20</td>
<td>77824</td>
<td>async_service_thread</td>
<td>SLEEP off=F366CE</td>
</tr>
<tr>
<td>0068E688</td>
<td>1459AC20</td>
<td>8192</td>
<td>keep_alive_thread</td>
<td>SLEEP off=00E8</td>
</tr>
<tr>
<td>00686090</td>
<td>0C4EC20</td>
<td>8192</td>
<td>tpq_HelperThread</td>
<td>SLEEP off=0E6E</td>
</tr>
<tr>
<td>006D3738</td>
<td>0CE6E5FO</td>
<td>8192</td>
<td>async_service_thread</td>
<td>SLEEP off=0C30</td>
</tr>
<tr>
<td>006CDE88</td>
<td>1363EC20</td>
<td>8192</td>
<td>ioeglue_service_thread</td>
<td>WAITECB off=F366CE</td>
</tr>
<tr>
<td>006D3738</td>
<td>0CE6E5FO</td>
<td>8192</td>
<td>tkm_TokenGcThread</td>
<td>SLEEP off=0C30</td>
</tr>
<tr>
<td>006CDE88</td>
<td>1363EC20</td>
<td>8192</td>
<td>keep_alive_thread</td>
<td>SLEEP off=00E8</td>
</tr>
<tr>
<td>006CDE88</td>
<td>1363EC20</td>
<td>8192</td>
<td>tpq_DispatcherThread</td>
<td>SLEEP off=0E6E</td>
</tr>
<tr>
<td>006D3738</td>
<td>0CE6E5FO</td>
<td>8192</td>
<td>async_service_thread</td>
<td>SLEEP off=0C30</td>
</tr>
<tr>
<td>006CDE88</td>
<td>1363EC20</td>
<td>8192</td>
<td>ioeglue_service_thread</td>
<td>WAITECB off=F366CE</td>
</tr>
<tr>
<td>006D3738</td>
<td>0CE6E5FO</td>
<td>8192</td>
<td>tkm_TokenGcThread</td>
<td>SLEEP off=0C30</td>
</tr>
</tbody>
</table>

Conversely, if you reviewed the outputs and task 6856D8 changes status between outputs, you do not have a hang. For example, in the first query,threads shows:

<table>
<thead>
<tr>
<th>TCB</th>
<th>Stack</th>
<th>Size</th>
<th>Routine</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>006D0370</td>
<td>1440FC20</td>
<td>77824</td>
<td>async_service_thread</td>
<td>SLEEP off=F366CE</td>
</tr>
<tr>
<td>0068E688</td>
<td>1459AC20</td>
<td>8192</td>
<td>keep_alive_thread</td>
<td>SLEEP off=00E8</td>
</tr>
<tr>
<td>00686090</td>
<td>0C4EC20</td>
<td>8192</td>
<td>tpq_HelperThread</td>
<td>SLEEP off=0E6E</td>
</tr>
<tr>
<td>006D3738</td>
<td>0CE6E5FO</td>
<td>8192</td>
<td>async_service_thread</td>
<td>SLEEP off=0C30</td>
</tr>
<tr>
<td>006CDE88</td>
<td>1363EC20</td>
<td>8192</td>
<td>ioeglue_service_thread</td>
<td>WAITECB off=F366CE</td>
</tr>
<tr>
<td>006D3738</td>
<td>0CE6E5FO</td>
<td>8192</td>
<td>tkm_TokenGcThread</td>
<td>SLEEP off=0C30</td>
</tr>
<tr>
<td>006CDE88</td>
<td>1363EC20</td>
<td>8192</td>
<td>keep_alive_thread</td>
<td>SLEEP off=00E8</td>
</tr>
<tr>
<td>006CDE88</td>
<td>1363EC20</td>
<td>8192</td>
<td>tpq_DispatcherThread</td>
<td>SLEEP off=0E6E</td>
</tr>
<tr>
<td>006D3738</td>
<td>0CE6E5FO</td>
<td>8192</td>
<td>async_service_thread</td>
<td>SLEEP off=0C30</td>
</tr>
<tr>
<td>006CDE88</td>
<td>1363EC20</td>
<td>8192</td>
<td>ioeglue_service_thread</td>
<td>WAITECB off=F366CE</td>
</tr>
<tr>
<td>006D3738</td>
<td>0CE6E5FO</td>
<td>8192</td>
<td>tkm_TokenGcThread</td>
<td>SLEEP off=0C30</td>
</tr>
</tbody>
</table>

This indicates the thread is progressing and not hung in rtn=internal_item_obtain.
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, 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 SMB lock wait time per SMB you could perform the following calculation:

\[
\text{Avg. Lock Waits per SMB} = \frac{\text{Total waits for locks}}{\text{Total SMB calls}} = 0.242
\]

\[
= \frac{\text{Avg. Lock Waits per SMB} \times \text{Average lock wait time}}{0.522}
\]

SMB 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 call for the LFS file system, or zFS call time per SMB by looking at data from the LFS and ADAPTERS report, respectively.

SMB server tuning

This section discusses SMB server tuning.

Workloads

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 or deletions. Other factors include the file read to write ratio and write intensity of the workload.

SMB workloads

SMB Windows clients do not cache file status or directory contents nearly as often as other clients do, therefore SMB workloads characteristically send much more file lookups and directory reads than other clients. While SMB workloads have much higher message rates per client user task or command, SMB caches data to turn around 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. Refer to Appendix A, “Environment variables in SMB,” on page 143 for possible values.

_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 SMB allows the administrator to disable raw mode transmission.

_IOE_SMB_MAXXMT

This option controls how much data can be sent in one SMB. 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.

**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.

_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 NTLMv2, which is the highest level of authentication protocol available.

_IOE_SMB_NT_SMBS

This option determines the capabilities for the NTLM authentication protocol.

**Diagnostics**

The following sample output is from the SMBINT report. Use this information to determine the amount and type of file related calls made to the SMB file server by clients.

<table>
<thead>
<tr>
<th>SMB Call</th>
<th>Count</th>
<th>Average DFS Time (msecs)</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>smb_mkdir</td>
<td>20</td>
<td>144.581</td>
<td>31.926</td>
</tr>
<tr>
<td>smb_rmdir</td>
<td>20</td>
<td>96.246</td>
<td>19.172</td>
</tr>
<tr>
<td>smb_close</td>
<td>1779</td>
<td>3.310</td>
<td>2.195</td>
</tr>
<tr>
<td>smb_unlink</td>
<td>207</td>
<td>151.634</td>
<td>35.641</td>
</tr>
<tr>
<td>smb_mv</td>
<td>15</td>
<td>133.500</td>
<td>14.208</td>
</tr>
<tr>
<td>smb_getattr</td>
<td>1872</td>
<td>11.284</td>
<td>0.778</td>
</tr>
<tr>
<td>smb_setattr</td>
<td>469</td>
<td>8.206</td>
<td>0.872</td>
</tr>
<tr>
<td>smb_write</td>
<td>8644</td>
<td>343.186</td>
<td>12.129</td>
</tr>
<tr>
<td>smb_chkpath</td>
<td>45</td>
<td>6.176</td>
<td>0.516</td>
</tr>
<tr>
<td>smb_readBraw</td>
<td>154</td>
<td>6.149</td>
<td>4.993</td>
</tr>
<tr>
<td>smb_writeBraw</td>
<td>21</td>
<td>274.119</td>
<td>63.503</td>
</tr>
<tr>
<td>smb_setattrE</td>
<td>404</td>
<td>46.993</td>
<td>0.544</td>
</tr>
<tr>
<td>smb_getattrE</td>
<td>445</td>
<td>0.556</td>
<td>0.445</td>
</tr>
<tr>
<td>smb_lockingX</td>
<td>37</td>
<td>0.305</td>
<td>0.245</td>
</tr>
<tr>
<td>smb_openX</td>
<td>2008</td>
<td>30.339</td>
<td>6.214</td>
</tr>
<tr>
<td>smb_readX</td>
<td>20083</td>
<td>4.295</td>
<td>1.850</td>
</tr>
<tr>
<td>smb_trans2</td>
<td>2219</td>
<td>10.424</td>
<td>0.636</td>
</tr>
<tr>
<td>T2: find first</td>
<td>941</td>
<td>12.764</td>
<td>0.366</td>
</tr>
<tr>
<td>T2: find next</td>
<td>20</td>
<td>0.173</td>
<td>0.151</td>
</tr>
<tr>
<td>T2: qfsinfo</td>
<td>344</td>
<td>0.030</td>
<td>0.027</td>
</tr>
<tr>
<td>T2: qpathinfo</td>
<td>540</td>
<td>5.778</td>
<td>0.183</td>
</tr>
<tr>
<td>smb_findclose</td>
<td>20</td>
<td>3.793</td>
<td>3.670</td>
</tr>
<tr>
<td>smb_negprot</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>smb_sesssetupX</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

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

Note: Headings that refer to HFS are in fact any z/OS UNIX PFS: HFS, zFS, and others.

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 call time is included in the overall SMB response time but is not controllable by the SMB 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 might 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 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 SMB response time represented by Infoprint Server calls.

<table>
<thead>
<tr>
<th>Infoprint Server API</th>
<th>Average Print Time (msecs)</th>
<th>Count</th>
<th>Failed</th>
<th>Response Time</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbortPrintFile</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>BeginEnumJobs</td>
<td></td>
<td>30</td>
<td>0</td>
<td>1.395</td>
<td>0.496</td>
</tr>
<tr>
<td>BeginEnumPrinters</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>CancelJob</td>
<td></td>
<td>1</td>
<td>0</td>
<td>23.560</td>
<td>2.879</td>
</tr>
<tr>
<td>ClosePrintFile</td>
<td></td>
<td>3</td>
<td>0</td>
<td>31.473</td>
<td>2.806</td>
</tr>
<tr>
<td>CreatePrintFile</td>
<td></td>
<td>3</td>
<td>0</td>
<td>15.222</td>
<td>2.475</td>
</tr>
<tr>
<td>EndEnumJobs</td>
<td></td>
<td>30</td>
<td>0</td>
<td>0.042</td>
<td>0.041</td>
</tr>
<tr>
<td>EndEnumPrinters</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>EnumJobs</td>
<td></td>
<td>60</td>
<td>0</td>
<td>0.574</td>
<td>0.314</td>
</tr>
<tr>
<td>EnumPrinters</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>GetJobInfo</td>
<td></td>
<td>8</td>
<td>0</td>
<td>1.422</td>
<td>0.798</td>
</tr>
<tr>
<td>GetPrinterInfo</td>
<td></td>
<td>110</td>
<td>3</td>
<td>1.886</td>
<td>0.530</td>
</tr>
<tr>
<td>HoldJob</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>InitAPI</td>
<td></td>
<td>2</td>
<td>1</td>
<td>558.078</td>
<td>37.522</td>
</tr>
<tr>
<td>ReleaseJob</td>
<td></td>
<td>0</td>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>SetTerminationHandler</td>
<td></td>
<td>1</td>
<td>0</td>
<td>0.016</td>
<td>0.016</td>
</tr>
<tr>
<td>TermAPI</td>
<td></td>
<td>0</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>WritePrintFile</td>
<td></td>
<td>55</td>
<td>0</td>
<td>0.271</td>
<td>0.253</td>
</tr>
</tbody>
</table>

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 SMB server has two thread pools to service incoming 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.

### Tuning Options

The following environment variables control the size of the SMB file server main and secondary thread pools. 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 SMB 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 SMB 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 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        Ready : 0
Recvs : 386875   Recvfrms : 136
Writes : 0       Writev : 0
Sends : 0       Sendtos : 0
Cancels : 0     Cancel : 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.
Token Management
This section discusses token management.

Central SMB token manager
The SMB file server central token manager is a cache of tokens with a fixed maximum size. The maximum size is tailorable by the administrator. A large amount of file access activity increases the cache of tokens. When the central token manager runs out of tokens, it performs garbage collection to reclaim a large amount of tokens from clients. For more information, see _IOE_TKM_MAX_TOKENS in Table 7 on page 143. It is possible to ensure garbage collection does not happen at all.

Local User Glue
Local user access to file systems exported by SMB also obtain tokens to ensure remote client caches remain correct. The Distributed File Service glue code caches tokens in the UNIX System Services kernel address space. For all local user access to file systems exported by SMB, the 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 the SMB 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
To ensure proper file sharing semantics and client cache consistency, SMB uses token cache to obtain tokens for requests from the central token manager. Windows (SMB) clients use a form of caching called opportunistic locking. These opportunistic lock requests are mapped to SMB token requests. The server caches SMB tokens from the central token manager to reduce path lengths. 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 options. For defaults, see Table 7 on page 143.

_IOE_TKM_MAX_TOKENS
This is the maximum number of tokens that the central token manager can hold. 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 PFS user token requests. This is important if PFS file systems are updated concurrently from remote SMB clients and local PFS users.

_IOE_TKMGLUE_SERVER_THREADS
The number of threads available for processing requests from local PFS users to get tokens from the central token manager.

_IOE_TKCGLUE_CACHE_SIZE
The maximum number of files for the DFS glue token cache. When more files are cached, the need to obtain tokens from the central token manager decreases. The glue provides this cache in the z/OS UNIX Systems Services kernel address space.

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. 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.

```
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 Filesests 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 peak period.

**Note:** The TKN4 report is for DCE/DFS only.

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
```

Chapter 15. Tuning and debugging guidelines 131
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.

**Virtual Memory File Cache**

The SMB file server has a cache that is used to contain file data to reduce disk I/O 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 SMB server file systems: zFS, HFS, RFS, and LFS. The following are 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. 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 PFS file systems also caches data in the address space, but the VM cache is used for the PFS file systems to reduce calls and hence reduce pathlength. A production SMB file server should be run with a larger VM cache for SMB workloads than the current 1 M default.

When running large workloads from many SMB remote clients, especially if accessing (create, write, delete) the same files, it might be beneficial to change defaults for a number of environment variables. In a test environment for 50 SMB clients running up to 100 active sessions, generating over 2 billion SMB calls in 24 hours, the following environment variable settings showed a significant improvement:

- \_IOE\_SMB\_MAIN\_POOL=64 (This environment variable is related to the z/OS UNIX MAXTHREADS value.)
- \_IOE\_TKM\_MAX\_TOKENS=256000
- \_IOE\_VM\_CACHE\_SIZE=256M
- \_IOE\_VM\_MAX\_FILES=32768
- \_IOE\_VNODE\_CACHE\_SIZE=18432

You must ensure that you have sufficient real storage so that paging is not significantly increased.

**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. For default values, see Appendix A, “Environment variables in SMB,” on page 143.
_IOE_VM_MAX_FILES
This is the maximum number of files to cache.

_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 following example sets the cache size to 256 MB.

_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  Getatttrs  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 119018  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%.

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).

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 A, “Environment variables in SMB,” on page 143 for more information on these environment variables.

**File system caches**
Any z/OS UNIX System Services file system can 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 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 data sets, while zFS is accessed through z/OS UNIX System Services. This cache hides zFS/RFS implementations that do not use the same interfaces as the rest of SMB and provides pathlength reduction because it reduces calls substantially to MVS access methods or zFS. The performance of this cache is essential for RFS performance.

**Status**
File status is cached in an extension to the vnode. By caching this status calls are reduced to z/OS UNIX System Services or z/OS data set access methods. By caching vnode handles the number of calls are reduced to z/OS UNIX System Services to obtain and return handles. The size of this cache is controlled by the administrator.

**Permissions**
The file system cache stores user permissions to file system objects. This is done by querying the permissions from SAF or z/OS UNIX System Services. By saving the permissions, calls to z/OS UNIX System Services or SAF are reduced.

**Directory Contents**
The directory cache stores directory contents in SMB format. Use of this cache reduces lookup and directory read calls made to z/OS data set access methods or zFS.
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 SMB 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 a large vnode cache might be desirable.

_IOE_VNODE_CACHE_SIZE
This is the number of vnodes provided in the cache. The default is 6144. More vnodes are provided, if needed, resulting in less calls to 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. Also see, "VNODE report" on page 123.

_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. Use the ADAPTER command to determine the performance of the shared PFS and RFS cache and the performance of the PFS.

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
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

<table>
<thead>
<tr>
<th>Vnode Op</th>
<th>Count</th>
<th>Vnode Op</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>xoefs_hold</td>
<td>1158433</td>
<td>xoefs_readdir</td>
<td>143600</td>
</tr>
<tr>
<td>xoefs_rele</td>
<td>7651388</td>
<td>xoefs_create</td>
<td>41387</td>
</tr>
<tr>
<td>xoefs_inactive</td>
<td>0</td>
<td>xoefs_remove</td>
<td>41343</td>
</tr>
<tr>
<td>xoefs_getattr</td>
<td>2043702</td>
<td>xoefs_rename</td>
<td>8684</td>
</tr>
<tr>
<td>xoefs_setattr</td>
<td>361916</td>
<td>xoefs_mkdir</td>
<td>0</td>
</tr>
<tr>
<td>xoefs_access</td>
<td>519908</td>
<td>xoefs_rmdir</td>
<td>0</td>
</tr>
<tr>
<td>xoefs_lookup</td>
<td>5113507</td>
<td>xoefs_link</td>
<td>0</td>
</tr>
<tr>
<td>xoefs_getvolume</td>
<td>0</td>
<td>xoefs_smlink</td>
<td>0</td>
</tr>
<tr>
<td>xoefs_getlength</td>
<td>0</td>
<td>xoefs_readlink</td>
<td>0</td>
</tr>
<tr>
<td>xoefs_afsfid</td>
<td>8149223</td>
<td>xoefs_rdwr</td>
<td>0</td>
</tr>
<tr>
<td>xoefs_fid</td>
<td>0</td>
<td>xoefs_fsync</td>
<td>14362</td>
</tr>
<tr>
<td>xoefs_vmread</td>
<td>150510</td>
<td>xoefs_translate</td>
<td>0</td>
</tr>
<tr>
<td>xoefs_vwrite</td>
<td>101034</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total HFS Vnode Ops 25653277

VFS Call       | Average HFS Time (msecs) | CPU Time |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Response Time</td>
<td></td>
</tr>
<tr>
<td>v_access</td>
<td>82778</td>
<td>1.443</td>
</tr>
<tr>
<td>v_lookup</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>v_get</td>
<td>0</td>
<td>0.000</td>
</tr>
<tr>
<td>v_getattr</td>
<td>56325</td>
<td>18.862</td>
</tr>
<tr>
<td>v_setattr</td>
<td>356120</td>
<td>7.694</td>
</tr>
</tbody>
</table>
```
v_create 41387  41.396  0.196
v_link 0  0.000  0.000
v_mkdir 0  0.000  0.000
v_rdwr 265906  49.819  0.171
v_readdir 0  0.000  0.000
v_readlink 0  0.000  0.000
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

<table>
<thead>
<tr>
<th>Vnode Op</th>
<th>Count</th>
<th>Vnode Op</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>xrda_hold</td>
<td>0</td>
<td>xrda_readdir</td>
<td>40</td>
</tr>
<tr>
<td>xrda_rele</td>
<td>0</td>
<td>xrda_create</td>
<td>219</td>
</tr>
<tr>
<td>xrda_inactive</td>
<td>0</td>
<td>xrda_remove</td>
<td>197</td>
</tr>
<tr>
<td>xrda_getattr</td>
<td>17692</td>
<td>xrda_rename</td>
<td>10</td>
</tr>
<tr>
<td>xrda_setattr</td>
<td>1479</td>
<td>xrda_mkdir</td>
<td>20</td>
</tr>
<tr>
<td>xrda_access</td>
<td>21486</td>
<td>xrda_rmdir</td>
<td>20</td>
</tr>
<tr>
<td>xrda_lookup</td>
<td>7555</td>
<td>xrda_link</td>
<td>0</td>
</tr>
<tr>
<td>xrda_getvolume</td>
<td>0</td>
<td>xrda_symlink</td>
<td>0</td>
</tr>
<tr>
<td>xrda_getlength</td>
<td>0</td>
<td>xrda_readlink</td>
<td>0</td>
</tr>
<tr>
<td>xrda_afsfid</td>
<td>50687</td>
<td>xrda_rdwr</td>
<td>0</td>
</tr>
<tr>
<td>xrda_fid</td>
<td>0</td>
<td>xrda_fsync</td>
<td>0</td>
</tr>
<tr>
<td>xrda_vmmread</td>
<td>1591</td>
<td>xrda_translate</td>
<td>10059</td>
</tr>
<tr>
<td>xrda_vmmwrite</td>
<td>8468</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total RFS Vnode Ops 119493

RFS Adapter VFS Op Counts

<table>
<thead>
<tr>
<th>VFS Op</th>
<th>Count</th>
<th>VFS Op</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>r_get</td>
<td>0</td>
<td>r_readdir</td>
<td>20</td>
</tr>
<tr>
<td>r_relat</td>
<td>222</td>
<td>r_create</td>
<td>219</td>
</tr>
<tr>
<td>r_getattr</td>
<td>57</td>
<td>r_remove</td>
<td>197</td>
</tr>
<tr>
<td>r_setattr</td>
<td>1259</td>
<td>r_rename</td>
<td>10</td>
</tr>
<tr>
<td>r_access</td>
<td>478</td>
<td>r_mkdir</td>
<td>20</td>
</tr>
<tr>
<td>r_lookup</td>
<td>5</td>
<td>r_rmdir</td>
<td>20</td>
</tr>
<tr>
<td>r_export</td>
<td>0</td>
<td>r_link</td>
<td>0</td>
</tr>
<tr>
<td>r_fstatfs</td>
<td>2</td>
<td>r_symlink</td>
<td>0</td>
</tr>
<tr>
<td>r_rpn</td>
<td>0</td>
<td>r_readlink</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r_rdwr</td>
<td>10059</td>
</tr>
</tbody>
</table>

Total RFS Operations 12568

The following sections provide explanations of the ADAPTER report.

**Adapter caching statistics:** This section of the report 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 zFS/RFS cache. It also displays the size of an zFS/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 SMB protocol servers to the zFS adapter. Many of these calls do not result in a zFS call due to the presence of the shared cache.

Average zFS or HFS time: This section shows the average response time and CPU time in milliseconds for each HFS or zFS physical file system call. The administrator determines the portion of the overall SMB response time that is represented by zFS or 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 SMB server response time could indicate DASD I/O bottlenecks.

RFS adapter vnode op counts: This is the number of calls made to the 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 SMB RFS physical file system. Response times are not given for these calls because the SMB product provides the support for RFS. The RFS report shows the z/OS 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.

RFS
The RFS file system maps POSIX based file I/O and maps them to z/OS data set access method calls. It uses z/OS access methods to store file status information into z/OS catalogs and retrieves the information later. The important factors regarding RFS performance is the number of access method and z/OS dynamic allocation calls made and their corresponding response times. SMB attempts to perform I/O efficiently to and from the corresponding data sets. 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 z/OS data sets, 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 data sets 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 data set open/close requests. z/OS has a restriction where a data set must be closed by the same task that opened the data set. Because the 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 data set. SMB handles this restriction by making available a pool of threads used exclusively for data set 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 the local z/OS UNIX physical file system. An important option is the size of the cache used to store BSAM data set blocks. The VM cache caches the file contents in POSIX byte stream format. BSAM record data sets require additional control information imbedded in the disk blocks, therefore I/O to the data sets must use disk blocks properly formatted for the data set. RFS provides a buffer cache used to hold these blocks in raw data set 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` data set 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. z/OS allows applications to specify a block size of 0 which means z/OS will determine optimal block size, this is the recommended setting by SMB for BSAM data sets (sequential data sets and PDS or PDS/E members).

- **recordsize**
  
  The RFS processing option specified in the RFS attribute file. This sets the size of the records for VSAM data sets. SMB will read/write up to 64K of data at a time to a VSAM data set. 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 data set. Data set 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 data set 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

<table>
<thead>
<tr>
<th>Access Method</th>
<th>OPENs</th>
<th>CLOSEs</th>
<th>READs</th>
<th>WRITEs</th>
<th>CHECKs</th>
<th>WAITS</th>
<th><em>Total</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>BPAM</td>
<td>40</td>
<td>40</td>
<td>0</td>
<td>15</td>
<td>30</td>
<td>15</td>
<td>140</td>
</tr>
<tr>
<td>BSAM</td>
<td>263</td>
<td>263</td>
<td>509</td>
<td>704</td>
<td>1276</td>
<td>1213</td>
<td>4228</td>
</tr>
<tr>
<td>QSAM</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>60</td>
</tr>
<tr>
<td>VSAM</td>
<td>526</td>
<td>526</td>
<td>640</td>
<td>4062</td>
<td>903</td>
<td>0</td>
<td>6657</td>
</tr>
<tr>
<td><em>Total</em></td>
<td>849</td>
<td>849</td>
<td>1169</td>
<td>4781</td>
<td>2209</td>
<td>1228</td>
<td>11085</td>
</tr>
</tbody>
</table>

Average Time Per Call By Access Method
---------------------------------------

<table>
<thead>
<tr>
<th>Access Method</th>
<th>OPENs</th>
<th>CLOSEs</th>
<th>READs</th>
<th>WRITEs</th>
<th>CHECKs</th>
<th>WAITS</th>
<th><em>Avg</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>BPAM</td>
<td>4.901</td>
<td>7.276</td>
<td>0.000</td>
<td>0.163</td>
<td>0.015</td>
<td>2.708</td>
<td>3.790</td>
</tr>
<tr>
<td>BSAM</td>
<td>10.202</td>
<td>23.315</td>
<td>0.200</td>
<td>0.186</td>
<td>0.120</td>
<td>4.688</td>
<td>3.521</td>
</tr>
<tr>
<td>QSAM</td>
<td>11.428</td>
<td>2.705</td>
<td>0.008</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>4.714</td>
</tr>
<tr>
<td>VSAM</td>
<td>44.727</td>
<td>42.300</td>
<td>1.894</td>
<td>0.891</td>
<td>0.694</td>
<td>0.000</td>
<td>7.696</td>
</tr>
<tr>
<td><em>Avg</em></td>
<td>31.371</td>
<td>112.676</td>
<td>1.124</td>
<td>0.785</td>
<td>0.353</td>
<td>4.664</td>
<td>8.630</td>
</tr>
</tbody>
</table>

Average CPU Time Per Call By Access Method
-------------------------------------------

<table>
<thead>
<tr>
<th>Access Method</th>
<th>OPENs</th>
<th>CLOSEs</th>
<th>READs</th>
<th>WRITEs</th>
<th>CHECKs</th>
<th>WAITS</th>
<th><em>Avg</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>BPAM</td>
<td>1.785</td>
<td>1.621</td>
<td>0.000</td>
<td>0.100</td>
<td>0.015</td>
<td>0.031</td>
<td>0.991</td>
</tr>
<tr>
<td>BSAM</td>
<td>2.097</td>
<td>1.705</td>
<td>0.112</td>
<td>0.104</td>
<td>0.055</td>
<td>0.028</td>
<td>0.292</td>
</tr>
<tr>
<td>QSAM</td>
<td>2.488</td>
<td>1.206</td>
<td>0.008</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.234</td>
</tr>
<tr>
<td>VSAM</td>
<td>13.534</td>
<td>9.203</td>
<td>1.628</td>
<td>0.495</td>
<td>0.581</td>
<td>0.000</td>
<td>2.334</td>
</tr>
<tr>
<td><em>Avg</em></td>
<td>9.177</td>
<td>26.340</td>
<td>0.940</td>
<td>0.436</td>
<td>0.269</td>
<td>0.028</td>
<td>1.532</td>
</tr>
</tbody>
</table>

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 **xrda_vmread** and **xrda_vmmwrite** 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 z/OS access methods. SMB 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 SMB and determine what portion of the response time the access methods use.

**Locking and serialization**
The SMB server provides information on 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 SMB file server monitors. 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: 71331781 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
```

---

140 z/OS V1R11.0 Distributed File Service SMB Administration
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 an 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 SMB file server and client provides a command that reports storage usage of the runtime heap; storage that SMB allocates directly from base z/OS storage subpools. Therefore all SMB allocated storage is shown with the exception of storage allocated for thread stacks. There is no way for SMB to track that information, but there are LE runtime options that shows you that information. For more information, see [z/OS Language Environment Programming Guide for 64-bit Virtual Addressing Mode](https://pubs.opengroup.org/onlinepubs/009695399/basedocsRgb/lepg/).

**Diagnostics**

The following sample output is from the STORAGE report.

```plaintext
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 SMB program use of heap storage. MVS and TCB owned storage statistics indicate the number of bytes allocated directly from z/OS storage subpools.
Appendix A. Environment variables in SMB

The following table lists the environment variables that affect the behavior of the SMB components. The environment variables that begin with _IOE are interpreted by Distributed File Service processes. The other environment variables are interpreted by other z/OS components (for example, z/OS Language Environment).

**Note:** 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 7. Environment variables in SMB**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| _EUV_AUTOLOG           | This environment variable controls whether DCE autologin processing is attempted. For the SMB server processes (dfsctl, dfskern, and dfsexport), this environment variable should always be set to NO. SMB Administrators who are not using DCE facilities should also specify this environment variable as NO in their OMVS environment. SMB users that are storing their SMB password by using the smbpw command should also specify this environment variable as NO in their OMVS environment. The valid value is:   
|                         | NO Do not enable single sign-on processing.                                                                                                        |
|                         | Any other value causes DCE autologin processing to be attempted.                                                                                   |
| _EUV_SVC_MSG_LOGGING   | This environment variable controls where SMB server messages are routed and displayed.                                                                                                                     |
| Default Value           | NO_LOGGING                                                                                                                                                                                                  |
| Expected Value          |                                                                                                                                                |
|                         | **NO_LOGGING**   
|                         | All messages are suppressed.                                                                                                                        |
|                         | **Note:** It is not recommended to set (or default) to this value.                                                                                   |
|                         | **CONSOLE_LOGGING**   
|                         | This is the recommended setting. All messages are routed to the operator console. All messages are in English (the NLSPATH environment variable is not used).                                                   |
|                         | **STDOUT_LOGGING**   
|                         | All messages are routed to the standard output file (the JES output). NLSPATH is used for messages if it exists.                                                                                           |
| Example                 | _EUV_SVC_MSG_LOGGING=CONSOLE_LOGGING                                                                                                                                                                        |
| Where Variable is Used  | All SMB server processes.                                                                                                                             |
### Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DCE_START_SOCKET_NAME</strong></td>
<td>The path name used as the well-known socket name by the SMB server processes during initialization.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>/opt/dfslocal/home/dfskern/ioepk.soc</td>
</tr>
<tr>
<td><strong>Expected Value</strong></td>
<td>Character string.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>DCE_START_SOCKET_NAME=/opt/dfslocal/home/dfscntl/ioepk.soc</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>If the default is not being used, this environment variable must be coded in the <code>envar</code> file for each process.</td>
</tr>
<tr>
<td><strong>LIBPATH</strong></td>
<td>The path names used to find DLLs.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Expected Value</strong></td>
<td>Character string.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>LIBPATH=/usr/lib:/usr/lpp/Printsrv/lib</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>DFSKERN</td>
</tr>
<tr>
<td><strong>NLSPATH</strong></td>
<td>A POSIX environment variable used by DCE that sets the search path for message catalogs.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>/usr/lib/nls/msg/En_US.IBM-1047/%N:</td>
</tr>
<tr>
<td></td>
<td>/usr/lib/nls/msg/%L/%N:</td>
</tr>
<tr>
<td></td>
<td>/usr/lib/nls/msg/prime/%N</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>All SMB server processes</td>
</tr>
<tr>
<td><strong>TZ</strong></td>
<td>Sets the time zone used by a process.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>localtime</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>All SMB server processes</td>
</tr>
<tr>
<td><strong>_IOE_DAEMONS_IN_AS</strong></td>
<td>This environment variable controls whether the DFSKERN process runs in its own address space or in the DFS Server Address Space.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>The default is &quot;.&quot; If &quot; is specified or the environment variable is not specified, <code>dfskern</code> runs in the DFS Server Address Space.</td>
</tr>
<tr>
<td><strong>Expected Value</strong></td>
<td>DFSKERN or &quot;</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>_IOE_DAEMONS_IN_AS=DFSKERN</td>
</tr>
<tr>
<td></td>
<td>In this case, DFSKERN runs in its own address space</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>DFSCNTL</td>
</tr>
</tbody>
</table>
### Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_DFS_MODIFY_PATH</td>
<td>The path used as the well-known socket name by the SMB processes when registering with DFSCNTL to receive modify commands (F DFS,QUERY DFSKERN).</td>
</tr>
<tr>
<td>Default Value</td>
<td>/opt/dfslocal/home/dfscntl/modify.rendezvous</td>
</tr>
<tr>
<td>Expected Value</td>
<td>Character string.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_DFS_MODIFY_PATH=/opt/dfslocal/home/dfscntl/test.rendezvous</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>All programs. If the default is not being used, this environment variable must be coded in the <code>envar</code> file for each process.</td>
</tr>
</tbody>
</table>

| _IOE_DIRECTORY_CACHE_SIZE   | The number of 512 byte blocks used to cache HFS directory entries.                                                                          |
| Default Value               | 2048                                                                                                                                       |
| Expected Value              | The value specified must be a numeric value greater than or equal to 768.                                                                     |
| Example                     | _IOE_DIRECTORY_CACHE_SIZE=4096                                                                                                               |
| Where Variable is Used      | DFSKERN                                                                                                                                    |

| _IOE_DYNAMIC_EXPORT         | If ON, when a client crosses a local mount point into a file system that is not known to the SMB server, the file system is dynamically assigned values and exported. |
| Default Value               | OFF                                                                                                                                        |
| Expected Value              | ON or OFF                                                                                                                                  |
| Example                     | _IOE_DYNAMIC_EXPORT=ON                                                                                                                     |
| Where Variable is Used      | DFSKERN                                                                                                                                    |
| Notes                       | This environment variable is turned off when DCE DFS processing is enabled (when _IOE_PROTOCOL_RPC=ON).                                      |

<p>| _IOE_EXPORT_TIMEOUT         | The number of minutes that a file system must be idle before it is dynamically unexported. The root of a share is never dynamically unexported.       |
| Default Value               | 15                                                                                                                                          |
| Expected Value              | OFF or a number of minutes greater than 0 and less than 480 (8 hours).                                                                     |
| Example                     | _IOE_EXPORT_TIMEOUT=OFF                                                                                                                     |
| Where Variable is Used      | DFSKERN                                                                                                                                    |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_HANG_DETECT_DUMP_LIMIT</td>
<td>Specifies the maximum number of dumps for the hang detector to take.</td>
</tr>
<tr>
<td><strong>Default Value</strong> 5</td>
<td></td>
</tr>
<tr>
<td><strong>Expected Value</strong> A number greater than or equal to five.</td>
<td></td>
</tr>
<tr>
<td><strong>Example</strong> _IOE_HANG_DETECT_DUMP_LIMIT=5</td>
<td></td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong> DFKERN</td>
<td></td>
</tr>
<tr>
<td>_IOE_HANG_DETECT_INTERVAL</td>
<td>Specifies the interval, in seconds, for which hangs are checked.</td>
</tr>
<tr>
<td><strong>Default Value</strong> 10</td>
<td></td>
</tr>
<tr>
<td><strong>Expected Value</strong> A number greater than or equal to zero.</td>
<td></td>
</tr>
<tr>
<td><strong>Example</strong> _IOE_HANG_DETECT_INTERVAL=10</td>
<td></td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong> DFKERN</td>
<td></td>
</tr>
<tr>
<td>_IOE_HANG_DETECT_OENODE_TIMEOUT</td>
<td>Specifies the period of time, in seconds, a thread must be hung before it is detected. When zero is specified, the hang detect function is off.</td>
</tr>
<tr>
<td><strong>Default Value</strong> 0</td>
<td></td>
</tr>
<tr>
<td><strong>Expected Value</strong> A number greater than or equal to zero.</td>
<td></td>
</tr>
<tr>
<td><strong>Example</strong> _IOE_HANG_DETECT_OENODE_TIMEOUT=30</td>
<td></td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong> DFKERN</td>
<td></td>
</tr>
<tr>
<td>_IOE_HANG_DETECT_THREAD_TIMEOUT</td>
<td>Specifies the period of time, in seconds, a thread must wait before it is detected as a hung thread. When zero is specified, the hang detect function is off.</td>
</tr>
<tr>
<td><strong>Default Value</strong> 0</td>
<td></td>
</tr>
<tr>
<td><strong>Expected Value</strong> A number greater than or equal to zero.</td>
<td></td>
</tr>
<tr>
<td><strong>Example</strong> _IOE_HANG_DETECT_THREAD_TIMEOUT=50</td>
<td></td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong> DFKERN</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| _IOE_HFS_ATTRIBUTES_FILE | Specifies the path name of the `hfsattr` file that contains the definition of file name suffixes that controls whether the data should be translated by the SMB 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 `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. It is recommended that the modified file reside in the `/opt/dfslocal/var/dfs` directory so that it is with the other customizable files. |
| _IOE_HFS_FILETAG    | The valid values are:  
  IGNORE: Do not query or set file tags.  
  QUERY: Honor tags when reading or writing but do not set tags when a file is created.  
  SET: QUERY and also set tags when files are created.  
  Default Value: IGNORE  
  Expected Value: IGNORE, QUERY, or SET.  
  Example: `_IOE_HFS_FILETAG=QUERY`  
  Where Variable is Used: DFSKERN |
Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_HFS_TRANSLATION</td>
<td>This environment variable controls the conversion of HFS data to the appropriate data format. Converts incoming data from ASCII ISO8859-1 (or the code page specified in the _IOE_WIRE_CODEPAGE environment variable of the dfskern process) to the local code page. Converts outgoing data from the local code page to ASCII ISO8859-1 (or the code page specified in the _IOE_WIRE_CODEPAGE environment variable of the dfskern process). Refer to &quot;File data translation for HFS&quot; on page 46, for information on data translation.</td>
</tr>
<tr>
<td>Default Value</td>
<td>OFF</td>
</tr>
<tr>
<td>Expected Value</td>
<td>ON, OFF, or AUTO</td>
</tr>
<tr>
<td></td>
<td><strong>ON</strong> means that all data is translated.</td>
</tr>
<tr>
<td></td>
<td><strong>OFF</strong> means that no data is translated.</td>
</tr>
<tr>
<td></td>
<td><strong>AUTO</strong> means that data that is received by (client write) or output from (client read) the SMB server is examined to determine if the data is text or not. When data is received by (client write) the SMB 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 characters, the file is marked for translation and all data is translated. Otherwise, the data is not translated. When data is output from (client read) the SMB server, the first 255 bytes of data are compared against a table of valid EBCDIC text characters. If all 255 characters are deemed to be valid characters, the file is marked for translation and all data is translated. Otherwise, the data is not translated. Refer to &quot;devtab&quot; on page 91 (auto parameter) for a list of valid code points.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_HFS_TRANSLATION=ON</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_MOVE_SHARED_FILESYSTEM</td>
<td>Specifies whether the SMB server should attempt to move the ownership of (Sysplex) Shared File Systems when it exports them.</td>
</tr>
<tr>
<td>Default Value</td>
<td>OFF</td>
</tr>
<tr>
<td>Expected Value</td>
<td>ON or OFF</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_MOVE_SHARED_FILESYSTEM=ON</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
</tbody>
</table>
### Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>_IOE_INHERIT_TRANSLATION</strong></td>
<td>When ON, for a dynamically exported file system, the translation option of the parent file system is inherited.</td>
</tr>
<tr>
<td>Default Value</td>
<td>ON</td>
</tr>
<tr>
<td>Expected Value</td>
<td>ON or OFF</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_INHERIT_TRANSLATION=OFF</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td><strong>_IOE_MVS_DFSDFLT</strong></td>
<td>The name of a RACF-defined user that is associated with unauthenticated users attempting to access shared directories or shared printers. This ID must be RACF-defined with a z/OS UNIX segment.</td>
</tr>
<tr>
<td>Default Value</td>
<td>None</td>
</tr>
<tr>
<td>Expected Value</td>
<td>A character string, eight characters or less.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_MVS_DFSDFLT=DFSDFLT</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td><strong>_IOE_PROTOCOL_RPC</strong></td>
<td>An environment variable that controls whether the DCE DFS protocol is supported (using DCE RPC).</td>
</tr>
<tr>
<td>Default Value</td>
<td>ON</td>
</tr>
<tr>
<td>Expected Value</td>
<td>ON or OFF</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_PROTOCOL_RPC=OFF</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Notes</td>
<td>This environment variable also affects DCE applications and commands. It should not be used in any other processes including shell user processes.</td>
</tr>
<tr>
<td><strong>_IOE_PROTOCOL_SMB</strong></td>
<td>An environment variable that controls whether the SMB protocol is supported (using TCP/IP).</td>
</tr>
<tr>
<td>Default Value</td>
<td>OFF</td>
</tr>
<tr>
<td>Expected Value</td>
<td>ON or OFF</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_PROTOCOL_SMB=ON</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| _IOE_RFS_ALLOC_TIMEOUT | Specifies the time period (in seconds) that an RFS data set remains allocated after there has been no access to the data set through the DFS/SMB server. After this time period, the data set is deallocated and is available to other applications such as ISPF.  
Default Value 300 (5 minutes)  
Expected Value A number greater than or equal to 30.  
Example  
Where Variable is Used DFSKERN |
| _IOE_RFS_ATTRIBUTES_FILE | Specifies the name of the (rfstab) file that contains the table for describing the attributes used to manipulate RFS files. The value can be the name of an 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  
Where Variable is Used DFSKERN |
| _IOE_RFS_STATUS_REFRESH_TIME | Specifies the time interval (in seconds) that the SMB server uses to refresh its cache of exported record data set names and attributes.  
Default Value 600 (10 minutes)  
Expected Value A number greater than zero.  
Example  
Where Variable is Used DFSKERN |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_RFS_TRANSLATION</td>
<td>Specifies whether RFS data should be translated. If conversion is on, incoming data is translated from ASCII ISO8859-1 (or the code page specified in the _IOE_WIRE_CODEPAGE environment variable of the dfskern process) to the local code page. Outgoing data is converted from the local code page to ISO8859-1 (or the code page specified in the _IOE_WIRE_CODEPAGE environment variable of the dfskern process).</td>
</tr>
<tr>
<td>Default Value</td>
<td>OFF</td>
</tr>
<tr>
<td>Expected Value</td>
<td>ON or OFF</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_RFS_TRANSLATION=ON</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Notes</td>
<td>This environment variable can be overridden for specific file systems in the devtab file. Refer to &quot;devtab&quot; on page 91.</td>
</tr>
<tr>
<td>_IOE_RFS_WORKER_THREADS</td>
<td>Specifies the number of threads to be started in dfskern to service open and close requests for RFS files.</td>
</tr>
<tr>
<td>Default Value</td>
<td>1</td>
</tr>
<tr>
<td>Expected Value</td>
<td>A number greater than zero.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_RFS_WORKER_THREADS=3</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Note:</td>
<td>This specification is related to the z/OS UNIX MAXTHREADS value. See step 6 on page 12 for additional information.</td>
</tr>
<tr>
<td>_IOE_SMB_ABS_SYMLINK</td>
<td>Specifies whether the SMB server can use absolute symbolic links.</td>
</tr>
<tr>
<td>Default Value</td>
<td>OFF</td>
</tr>
<tr>
<td>Expected Value</td>
<td>ON or OFF</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_ABS_SYMLINK=OFF</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_SMB_AUTH_DOMAIN_NAME</td>
<td>Specifies the domain name of the Domain Controller.</td>
</tr>
<tr>
<td>Default Value</td>
<td>None</td>
</tr>
<tr>
<td>Expected Value</td>
<td>Domain name, 15 characters or less.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_AUTH_DOMAIN_NAME=DOMAIN1</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>_IOE_SMB_AUTH_SERVER</td>
<td>Specifies the IP address of the Domain Controller that will be used to authenticate PC users.</td>
</tr>
<tr>
<td>_IOE_SMB_AUTH_SERVER_COMPUTER_NAME</td>
<td>Specifies the computer name of the Domain Controller that will be used to authenticate PC users. This name is used to pass the NetBIOS computer name of the Windows Domain Controller to the Windows Domain Controller. This name is not used to find a Windows Domain Controller. _IOE_SMB_AUTH_SERVER and _IOESMB_BACKUP_AUTH_SERVER are used for that purpose.</td>
</tr>
<tr>
<td>_IOE_SMB_BACKUP_AUTH_SERVER</td>
<td>Specifies the IP address of the Domain Controller that will be the backup for PC user authentication.</td>
</tr>
<tr>
<td>_IOE_SMB_BACKUP_AUTH_SERVER_COMPUTER_NAME</td>
<td>Specifies the computer name of the Backup Domain Controller that will be used to authenticate PC users. This name is used to pass the NetBIOS computer name of the Windows Domain Controller to the Windows Domain Controller. This name is not used to find a Windows Domain Controller. _IOE_SMB_AUTH_SERVER and _IOESMB_BACKUP_AUTH_SERVER are used for that purpose.</td>
</tr>
</tbody>
</table>
### Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default Value</th>
<th>Expected Value</th>
<th>Example</th>
<th>Where Variable is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_SMB_BLKSIZE</td>
<td>Specifies the block size value used by SMB clients.</td>
<td>32K</td>
<td>0K, 8K, 16K, or 32K (the 'K' is required). If you specify zero, the SMB server returns the block size assigned by the PFS.</td>
<td>_IOE_SMB_BLKSIZE=32K</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_SMB_BROWSE_INTERVAL</td>
<td>Specifies the maximum Browser announcement interval (in milliseconds).</td>
<td>720000 (12 minutes)</td>
<td>A number in the range of 600000 (10 minutes) - 720000 (12 minutes).</td>
<td>_IOE_SMB_BROWSE_INTERVAL=600000</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_SMB_CALLBACK_POOL</td>
<td>Specifies the number of secondary pool threads for processing SMB callback requests.</td>
<td>2</td>
<td>A number greater than 0.</td>
<td>_IOE_SMB_CALLBACK_POOL=3</td>
<td>DFSKERN</td>
</tr>
</tbody>
</table>

**Note:** This specification is related to the z/OS UNIX MAXTHREADS value. See step [5 on page 12](#) for additional information.
### Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_SMB_CLEAR_PW</td>
<td>Specifies the SMB server policy for flowing the SMB password in the clear.</td>
</tr>
<tr>
<td></td>
<td><strong>REQUIRED</strong> Specifies that passwords in the clear are required.</td>
</tr>
<tr>
<td></td>
<td><strong>ALLOWED</strong> 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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> ALLOWED was originally provided for clients that had no support for encrypted passwords. We have subsequently determined that all supported clients include encrypted password support, so the ALLOWED setting is not useful. We suggest that you use REQUIRED for clear passwords and NOTALLOWED for encrypted passwords.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTALLOWED</strong> Specifies that passwords in the clear are not allowed. Authentication is attempted using encrypted passwords only.</td>
</tr>
<tr>
<td>Default Value</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Expected Value</td>
<td>REQUIRED, ALLOWED, NOTALLOWED</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_CLEAR_PW=NOTALLOWED</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Notes</td>
<td>If passthrough authentication is used, see <a href="#">Using passthrough authentication</a> on page 55 for additional information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>_IOE_SMB_CONNECT_MSGS</th>
<th>This environment variable is for use in debugging or determining problems with connection issues. When in use, messages are issued for session connects, session configuration, and session cancels.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Value</td>
<td>0 (zero)</td>
</tr>
<tr>
<td>Expected Value</td>
<td>• 0 = No connection messages are issued.</td>
</tr>
<tr>
<td></td>
<td>• 1 = Only error connection messages are issued.</td>
</tr>
<tr>
<td></td>
<td>• 2 = All error and informational connection messages are issued.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_CONNECT_MSGS=1</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>_IOE_SMB_COMPUTER_NAME</td>
<td>Specifies the name being used by SMB redirectors (that is, clients) to contact this server. If a Windows Internet Naming Service (WINS) server is available (refer to the _IOE_SMB_PRIMARY_WINS environment variable <a href="#">on page 158</a>), the SMB computer name is used to identify this server to the WINS server.</td>
</tr>
<tr>
<td>Default Value</td>
<td>The TCP/IP hostname of this system.</td>
</tr>
<tr>
<td>Expected Value</td>
<td>Character string, 15 characters or less</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_COMPUTER_NAME=OS390DATA1</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Notes</td>
<td>If you are using Windows clients, this environment variable must specify (or be defaulted to) the TCP/IP hostname to allow Search Computername functionality to work.</td>
</tr>
<tr>
<td>_IOE_SMB_CROSS_MOUNTS</td>
<td>When ON, SMB clients cross local mount points (as in prior releases). When OFF, SMB clients go under local mount points.</td>
</tr>
<tr>
<td>Default Value</td>
<td>ON</td>
</tr>
<tr>
<td>Expected Value</td>
<td>ON or OFF</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_CROSS_MOUNTS=OFF</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_SMB_DESCRIPTION</td>
<td>Specifies the description of this server that appears on the PC.</td>
</tr>
<tr>
<td>Default Value</td>
<td>DFS/MVS CIFS Server</td>
</tr>
<tr>
<td>Expected Value</td>
<td>Character string, 50 characters or less</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_DESCRIPTION=z/OS File Server</td>
</tr>
<tr>
<td>Note:</td>
<td>In this example there are embedded blanks.</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_SMB_DIR_PERMS</td>
<td>The permissions for a directory created by the SMB server at the request of an SMB client. (This can also be specified on a shared directory basis. Refer to <a href="#">smbtab on page 106</a>.)</td>
</tr>
<tr>
<td>Default Value</td>
<td>777</td>
</tr>
<tr>
<td>Expected Value</td>
<td>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.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_DIR_PERMS=755</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>_IOE_SMB_DOMAIN_NAME</td>
<td>Specifies the name being used as the domain name for this server. If possible, specify the same domain or workgroup as your client PCs.</td>
</tr>
<tr>
<td>Default Value</td>
<td>WORKGROUP</td>
</tr>
<tr>
<td>Expected Value</td>
<td>Character string, 15 characters or less</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_DOMAIN_NAME=0S/390DOMAIN1</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFKERN</td>
</tr>
<tr>
<td>_IOE_SMB_FILE_PERMS</td>
<td>The permissions for a file created by the SMB server at the request of an SMB client. (This can also be specified on a shared directory basis. Refer to &quot;smbtab&quot; on page 106.) It also limits which write permission bits can be turned on by an SMB client.</td>
</tr>
<tr>
<td>Default Value</td>
<td>777</td>
</tr>
<tr>
<td>Expected Value</td>
<td>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.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_FILE_PERMS=750</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFKERN</td>
</tr>
<tr>
<td>Notes</td>
<td>If Read-only is set, the write permissions (222) are turned off. In this case, a file with permissions of 750 would become 550. If Read-only is cleared, the write permissions that intersect with this specification are turned on. In the example, the intersection of 750 and 222 would cause 200 to be or’d into the permissions of the file.</td>
</tr>
<tr>
<td>_IOE_SMB_IDMAP</td>
<td>Specifies the location of the <code>smbidmap</code> file. The <code>smbidmap</code> file contains the mapping of SMB user IDs to z/OS user IDs.</td>
</tr>
<tr>
<td>Default Value</td>
<td>None</td>
</tr>
<tr>
<td>Expected Value</td>
<td>Character string specifying the path name of the <code>smbidmap</code> file.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_IDMAP=/opt/dfslocal/home/dfskern/smbidmap</td>
</tr>
<tr>
<td></td>
<td>_IOE_SMB_IDMAP='USERID.SMB_OPTIONS(SMBIDMAP)'</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFKERN</td>
</tr>
<tr>
<td>_IOE_SMB_IDLE_TIMEOUT</td>
<td>Specifies how long (in seconds) an SMB session can remain inactive before it is terminated.</td>
</tr>
<tr>
<td>Default Value</td>
<td>400</td>
</tr>
<tr>
<td>Expected Value</td>
<td>A number in the range of 0 - 4294967295.</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_IDLE_TIMEOUT=4000</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFKERN</td>
</tr>
</tbody>
</table>
## Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default Value</th>
<th>Expected Value</th>
<th>Example</th>
<th>Where Variable is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_SMB_MAIN_POOL</td>
<td>Specifies the number of primary pool threads for processing SMB requests. This is the number of SMB requests that can be handled by the SMB server at the same time without queuing them. If you have a large number of concurrently active PC clients, consider increasing this number. Note that we are referring to concurrently active users, not just logged on users. Concurrently active users are sending SMB requests to the SMB server at the same time. This number should be set to your best estimate of the maximum number of concurrent requests that might be received at the SMB server at the same time.</td>
<td>14</td>
<td>A number greater than 0.</td>
<td>_IOE_SMB_MAIN_POOL=20</td>
<td>DFSKERN</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This specification is related to the z/OS UNIX MAXTHREADS value. See step 5 on page 12 for additional information.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_IOE_SMB_MAXXMT</td>
<td>Specifies the maximum server buffer size that is returned on the SMB Server Negotiate response.</td>
<td>65535 bytes</td>
<td>A number in the range of 1024 - 65535</td>
<td>_IOE_SMB_MAXXMT=8192</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_SMB_NT_SMBS</td>
<td>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.</td>
<td>ON</td>
<td>ON or OFF</td>
<td>_IOE_SMB_NT_SMBS=OFF</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_SMB_OCSF</td>
<td>Specifies whether OCSF is used for encrypted passwords. In order to use encryption hardware, OCSF (and therefore ON) is required.</td>
<td>ON</td>
<td>ON or OFF</td>
<td>_IOE_SMB_OCSF=OFF</td>
<td>DFSKERN</td>
</tr>
</tbody>
</table>
Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_SMB_OPLOCK_TIMEOUT</td>
<td>Specifies the Opportunistic Lock Timeout period (in seconds).</td>
</tr>
<tr>
<td>Default Value</td>
<td>35</td>
</tr>
<tr>
<td>Expected Value</td>
<td>A number in the range of 1 - 4294967295</td>
</tr>
<tr>
<td>Example</td>
<td>_IOE_SMB_OPLOCK_TIMEOUT=60</td>
</tr>
<tr>
<td>Where Variable is Used</td>
<td>DFSKERN</td>
</tr>
</tbody>
</table>

| _IOE_SMB_OPLOCKS            | Specifies whether the server permits clients to use opportunistic locks     |
|                             | on files and let some SMB clients 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                                                                     |

| _IOE_SMB_PRIMARY_WINS       | Specifies the IP address of the Windows Internet Naming Service             |
|                             | (WINS) server that this server announces itself to and forwards proxy      |
|                             | WINS requests to.                                                          |
| Default Value               | None                                                                       |
| Expected Value              | An IP address (n.n.n.n, where n is a number in the range of 0 - 255).      |
| Example                     | _IOE_SMB_PRIMARY_WINS=9.120.44.55                                          |
| Where Variable is Used      | DFSKERN                                                                     |

| _IOE_SMB_PROTOCOL_LEVEL     | Specifies the highest level of SMB protocol dialect that is being          |
|                             | negotiated with the PC client. NTLMv2 is the highest dialect that the      |
|                             | SMB server supports when you specify NT. LANMAN is the next lower          |
|                             | dialect. It is recommended that you run with the default value NT.        |
|                             | Running with the value LANMAN can restrict the capabilities and function-  |
|                             | ality of the SMB server. For example, if LANMAN is specified,             |
|                             | than _IOE_SMB_ABS_SYMLINK must be OFF.                                    |
| Default Value               | NT (which is the default setting for NTLMv2)                              |
| Expected Value              | NT or LANMAN                                                               |
| Example                     | _IOE_SMB_PROTOCOL_LEVEL=NT                                                 |
| Where Variable is Used      | DFSKERN                                                                     |
Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_SMB_RAW</td>
<td>Specifies whether raw mode is supported in the SMB Server Negotiate response. Raw mode is used for large data transfers.</td>
</tr>
<tr>
<td></td>
<td><strong>Default Value</strong> ON</td>
</tr>
<tr>
<td></td>
<td><strong>Expected Value</strong> ON or OFF</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> _IOE_SMB_RAW=OFF</td>
</tr>
<tr>
<td></td>
<td><strong>Where Variable is Used</strong> DFKERN</td>
</tr>
<tr>
<td>_IOE_SMB_SETATTR_OVERRIDE</td>
<td>Specifies whether the SMB server should appear to allow a user who is not the owner of a file to set the modtime of the file to an explicit value. z/OS UNIX does not allow a user who is not the owner of a file to explicitly set the modtime to a specific value. However, when any user updates a file, the modtime is updated implicitly to the current time. So normally, when a file is updated, there is no need to explicitly set the modtime to a specific time value. Some Windows applications attempt to set the modtime explicitly when a file is saved. When the user is not the owner of the file, this attempt will fail and this failure is reflected back to the PC user. As stated above, this attempt to update the modtime is unnecessary (because the modtime will be updated implicitly to the current time). When this environment variable is set to ON, the SMB server will ignore the failure (to set the modtime to an explicit value) and avoid the failure of saving the file. The modtime will be set to the current time.</td>
</tr>
<tr>
<td></td>
<td><strong>Default Value</strong> OFF</td>
</tr>
<tr>
<td></td>
<td><strong>Expected Value</strong> ON or OFF</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> _IOE_SMB_SETATTR_OVERRIDE=ON</td>
</tr>
<tr>
<td></td>
<td><strong>Where Variable is Used</strong> DFKERN</td>
</tr>
<tr>
<td>_IOE_SMB_SCOPE</td>
<td>Specifies the Scope ID for the Windows Internet Naming Service (WINS) server. The Scope ID defines a group of computers that recognize a registered NetBIOS name. (This should normally be omitted.)</td>
</tr>
<tr>
<td></td>
<td><strong>Default Value</strong> None</td>
</tr>
<tr>
<td></td>
<td><strong>Expected Value</strong> Character string, 224 characters or less</td>
</tr>
<tr>
<td></td>
<td><strong>Example</strong> _IOE_SMB_SCOPE=MYDEPARTMENTSCOPE</td>
</tr>
<tr>
<td></td>
<td><strong>Where Variable is Used</strong> DFKERN</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| _IOE_SMB_SECONDARY_WINS      | Specifies the IP address of the Windows Internet Naming Service (WINS) server that this server announces itself to and forwards proxy WINS requests to if the Primary WINS server does not respond.  
  Default Value: None  
  Expected Value: An IP address (n.n.n.n, where n is a number in the range of 0 - 255).  
  Example:  
  _IOE_SMB_SECONDARY_WINS=9.120.66.77  
  Where Variable is Used: DFSKERN                                                                                                                                                                                                                           |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Default Value</th>
<th>Expected Value</th>
<th>Example</th>
<th>Where Variable is Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_TKM_MAX_TOKENS</td>
<td>Specifies the maximum number of tokens that can be held in the SMB server memory.</td>
<td>10240</td>
<td>A positive number. The minimum value is 10240.</td>
<td>_IOE_TKM_MAX_TOKENS=20480</td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_TKMGLUE_SERVER_THREADS</td>
<td>The number of threads to be started in DFSKERN to service token requests from the glue layer. The glue layer makes token requests during local HFS access.</td>
<td>5</td>
<td>The value specified must be a numeric value greater than or equal to 5.</td>
<td>_IOE_TKMGLUE_SERVER_THREADS=8</td>
<td>DFSKERN</td>
</tr>
<tr>
<td></td>
<td>Note: This specification is related to the z/OS UNIX MAXTHREADS value. See step 5 on page 12 for additional information.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_IOE_TRACE_TABLE_SIZE</td>
<td>Specifies the size, in bytes, of the DFSKERN trace table</td>
<td>64 MB</td>
<td>A positive number between 4 MB and 256 MB. 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).</td>
<td>_IOE_TRACE_TABLE_SIZE=64M</td>
<td>DFSKERN</td>
</tr>
<tr>
<td></td>
<td>Notes: Do not change this setting unless you are instructed to do so by IBM Service.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>_IOE_USE_PTHREAD_SECURITY</td>
<td>When this environment variable is ON, pthread_security_np() is used to establish the user security environment. This results in security updates becoming active without restarting DFSKERN.</td>
<td>OFF</td>
<td>ON or OFF</td>
<td>_IOE_USE_PTHREAD_SECURITY=ON</td>
<td>DFSKERN</td>
</tr>
</tbody>
</table>
### Table 7. Environment variables in SMB (continued)

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_IOE_VM_CACHE_SIZE</td>
<td>Specifies the size, in bytes, of the virtual memory used by DFSKERN for all file systems that are exported.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>10M</td>
</tr>
<tr>
<td><strong>Expected Value</strong></td>
<td>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).</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>_IOE_VM_CACHE_SIZE=2M</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>DFSKERN</td>
</tr>
<tr>
<td><strong>Notes</strong></td>
<td>The virtual memory is used for data in all file systems that are exported.</td>
</tr>
<tr>
<td>_IOE_VM_MAX_FILES</td>
<td>Specifies the maximum number of files that can be contained in the virtual memory used by DFSKERN for all file systems that are exported.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>4096</td>
</tr>
<tr>
<td><strong>Expected Value</strong></td>
<td>A positive number. The minimum value is 4096.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>_IOE_VM_MAX_FILES=6144</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_VNODE_CACHE_SIZE</td>
<td>The size of the HFS vnode cache, that is, the number of vnodes.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>6144</td>
</tr>
<tr>
<td><strong>Expected Value</strong></td>
<td>The value specified must be a numeric value greater than or equal to 6144.</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>_IOE_VNODE_CACHE_SIZE=6144</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>DFSKERN</td>
</tr>
<tr>
<td>_IOE_WIRE_CODEPAGE</td>
<td>Specifies the codepage to be used for text data on the wire for clients.</td>
</tr>
<tr>
<td><strong>Default Value</strong></td>
<td>ISO8859-1</td>
</tr>
<tr>
<td><strong>Expected Value</strong></td>
<td>ISO8859-1 or codepage</td>
</tr>
<tr>
<td><strong>Example</strong></td>
<td>_IOE_WIRE_CODEPAGE=ISO8859-1</td>
</tr>
<tr>
<td><strong>Where Variable is Used</strong></td>
<td>DFSKERN</td>
</tr>
</tbody>
</table>
Appendix B. Additional information about using the SMB server

Important Note to Users
This information is outside the scope of the z/OS Distributed File Service SMB server. However, it is included here because it might be helpful to the SMB administrator or the PC user.

Client does not communicate
If the client does not communicate with the SMB server, it may be because a service pack has been applied or is present on the client that requires the challenge/response type of logon. For the service pack noted below (or a later service pack), the registry needs to be updated to permit unencrypted passwords.

Attention: Using Registry Editor incorrectly can cause serious, system-wide problems that may require you to reinstall Windows to correct them. Microsoft cannot guarantee that any problems resulting from the use of Registry Editor can be solved. Use this tool at your own risk.

Note: If you are using the encrypted password support and your users have entered their SMB password in RACF using the OMVS smbpw command, this registry change is not necessary. Users that have made this change to the registry do not need to remove it to use encrypted passwords.

Windows Vista, Windows XP and Windows 2003
This section shows you how to update the registry for Windows Vista, Windows XP and Windows 2003.

1. Click Start, point to Settings, click on Control Panel.
2. Double-click on Administrative Tools.
4. On the left pane, click on the plus (+) to expand Local Policies (under Security Settings).
5. On the left pane, click on Security Options.
6. Double-click on Microsoft network client: Send unencrypted password to third-party SMB servers.
7. Click the Enabled radio button, and then click OK.

If the registry still does not permit unencrypted passwords, do the following:
1. Windows XP: Execute regedit: Click Start, and then click Run. Type Regedit, and then click OK. For Windows Vista: Type Regedit in the search box and the Registry editor will open in a new window.
2. On the left pane, click on the plus (+) to expand HKEY_LOCAL_MACHINE
3. On the left pane, click on the plus (+) to expand SYSTEM
4. On the left pane, click on the plus (+) to expand CurrentControlSet
5. On the left pane, click on the plus (+) to expand Services
6. On the left pane, click on the plus (+) to expand Lanmanworkstation
7. On the left pane, click on parameters
8. On the right pane, select EnablePlainTextPassword
9. Click on Edit and select Modify
10. Update value data to hex “1”
11. Restart client machine for registry change to take effect
Insufficient space errors

Before saving certain files, some client applications attempt to determine if there is sufficient disk space available. An SMB request is sent to the server to query the space available in that file system. The request is against the file system at the share name; the request is not in the file system where the saved file will reside. As a result, you might receive a message similar to the following indicating that insufficient space is available to save the file.

The disk you were saving to or the disk used for temporary files is full.
Free some space on this disk and try again, or save to a different disk.

When crossing through multiple file systems or mount points, the SMB query only looks for space available at the share name and not in the file system below that point. This behavior is consistent with SMB implementations on other platforms (for example, Microsoft Windows, Redhat Linux with Samba).

Use one of the following scenarios to resolve this issue:

1. Statically exported file systems (_IOE_DYNAMIC_EXPORT=OFF)
   Verify that sufficient space exists in the file system at the exported share name mount point. Increase the size of the file system defined by the entry in the devtab configuration file.

2. Dynamically exported file systems (_IOE_DYNAMIC_EXPORT=ON)
   Verify that sufficient space exists in the file system at the exported share name mount point that represents the parent file system to any dynamically mounted file systems. Dynamically mounted file systems under the parent file system have no bearing on the amount of space returned from the query.

3. Automounted file systems scenarios (_IOE_DYNAMIC_EXPORT=ON and automount)
   a. Accessing an SMB share using the smbtab &USERID keyword
      Verify sufficient space is available for the automounted file system reference.
      Example: Using the sample configuration files in Figure 15 on page 165 if the user “smbuser” performed a net use to myhomedir and entered the dir command, the file system “smbuser.home.auto” is automounted and exported by the SMB server. Insure sufficient space in that file system exists when saving in this file system or in any file systems mounted below this point.
   b. Accessing an SMB share at the automount file system
      z/OS UNIX System Services reserves this file system for automount processing. The response to a file system query at this point will return zero space available. In this scenario, add a static entry in you devtab, dfstab, and smbtab for that file system and access through that exported share name mount point to save the file.
      Example: Using the sample configuration files in Figure 15 on page 165 if the user “smbuser” performed a net use to homedir, and then proceeds to traverse into a directory called smbuser, that triggers the mount of the file system “smbuser.home.auto”. If the user attempts a save there, the query file response would return zero bytes available.
Editor or word processor changes the owner/permissions of the HFS file

Many Editors and Word Processors, when saving a file, do not simply update the file. Rather, in order to avoid losing data if the system should fail during the save operation, they first create a new file with a temporary name and save the data there, then they erase the original file, and then they rename the file with the temporary name to the original file name. Since a new file was created, it is owned by the user that created it. This may be different than the original file's owner. Also, the file permissions are changed to the default permissions which may be different from the original permissions.

Editor or word processor cannot save a file to HFS

Many Editors and Word Processors, when saving a file, do not simply update the file. Rather, in order to avoid losing data if the system should fail during the save operation, they first create a new file with a temporary name and save the data there, then they erase the original file, and then they rename the file with the temporary name to the original file name. Since a new file is being created, the user must have write and execute permission to the directory that the file is contained in and read and write permission to the file. If the user does not have the required permissions on both the directory and the file, the save is denied.

Different end of line characters in text files

The PC environment and the z/OS UNIX environment normally use different end of line characters in text files. The end of line characters are placed into the file as the text lines are written by the application (for example, an editor) based on the environment the application is running on (Windows versus z/OS UNIX). If you are sharing text files between the PC environment and the z/OS UNIX environment, one of the environments sees end of line characters at the end of each line of text that it does not normally expect. Some applications may not process a text file if the wrong end of line characters are in the text file. For example, the C/C++ compiler does not compile source code with PC end of line characters. You should try to use applications on the PC that support z/OS UNIX end of line characters (see note below). Microsoft WordPad correctly displays text files that have z/OS UNIX end of line characters. However, when the file is saved, the end of line characters are changed to Windows end of line characters. There are also PC applications that optionally create a text file with z/OS UNIX end of line characters. An example of an
application that creates a text file with z/OS UNIX end of line characters is MicroEdge Visual SlickEdit. Many PC applications tolerate z/OS UNIX end of line characters.

If you want to determine which end of line characters a text file contains, you can use the following OMVS command to display the hexadecimal values of the characters in a file:

```
od -cx textfile.txt
```

where `textfile.txt` is the name of the text file you want to display.

When a text file has PC end of line characters, you can create another file with the same data but with z/OS UNIX end of line characters with the following OMVS command:

```
tr -d '\r' <textfile.txt >newfile.txt
```

where `textfile.txt` is the text file with PC end of line characters and `newfile.txt` is a new text file with z/OS UNIX end of line characters.

**Note:** In general, z/OS UNIX text files contain a newline character at the end of each line. In ASCII, newline is X'0A'. In EBCDIC, newline is X'15'. (For example, ASCII code page ISO8859-1 and EBCDIC code page IBM-1047 translate back and forth between these characters.) Windows programs normally use a carriage return followed by a line feed character at the end of each line of a text file. In ASCII, carriage return/line feed is X'0D'/X'0A'. In EBCDIC, carriage return/line feed is X'0D'/X'15'. The `tr` command above simply deletes all of the carriage return characters. (Line feed and newline characters have the same hexadecimal value.) The SMB server can translate end of line characters from ASCII to EBCDIC and back but it does not change the type of delimiter (PC vs. z/OS UNIX) nor the number of characters in the file.

### PC clients disconnect during high DASD I/O activity

When there is high DASD I/O activity on an HFS file system, there may be an undue delay for HFS requests. This may cause the PC clients to disconnect from the SMB server because they think the SMB server is down. A possible bypass to this situation is to mount the HFS file system with a sync interval that is less than 60 seconds. For example,

```
MOUNT FILESYSTEM('OMVS.ABC.DEF') MOUNTPOINT('/abc/def') TYPE(HFS) MODE(RDWR) PARM('SYNC(30)')
```

sets the sync interval to 30 seconds for the HFS file system specified. Refer to the TSO/E MOUNT command in the [z/OS UNIX System Services Command Reference](https://www.ibm.com/support/docview.pl?uid=ssg1s7003921), SA22-7802, for information on the sync interval parameter.

**Note:** This does not apply to ZFS file systems.

### SMB server does not show up in My Network Places

The My Network Places function is not required in order to use the SMB server. The SMB server can be contacted if you know its computer name by using the Find Computer function. It can also be contacted from Windows clients by using the DNS domain name of the SMB server in the `net use` command or the Map Network Drive function. Some Windows systems may require an LMHOSTS file.

If you want to use the My Network Places function, you must meet the following networking requirements:

- There must be a Windows 2003 domain controller on the same subnet as the SMB server.
- The Windows 2003 domain controller must be acting as a Domain Master Browser.

You can determine the roles that a Windows machine is running by using the `browstat view` command.
Writing of data appears successful but data isn’t there

If a write failure occurs due to the file system being full, the error will be returned on close. If the application / client ignores the error, the operation will appear to be successful. This can also occur if the end of line character(s) cannot be found in text data.

Sharing an Excel file editing user inaccurate

In V1R9 and above, if a user (user A) opens a Microsoft Excel spreadsheet that is stored on z/OS, and then a second user (user B) opens the same spreadsheet, a popup window appears that gives user B three choices:

1. Open the file read-only
2. Open the file read-only and be notified when the file has been saved/closed
3. Cancel - do not open the file.

In the popup window, the user that is indicated as having the file locked for editing might be inaccurate. If the current user that has the file open for write is using the z/OS SMB server and has not saved any updates, the editing user displayed is the last user that updated the file (possibly a while ago). See also "Editor or word processor changes the owner/permissions of the HFS file" on page 165. Excel apparently causes the owner to change when the spreadsheet is saved. This may deny a user (who is mapped to a different z/OS user) access to the file. The error message displayed did not indicate an authorization problem but that the file was read-only or encrypted.
**Appendix C. Using both SMB and DCE DFS**

This information must be considered if you plan to use the Distributed File Service DCE DFS protocols (enabled by using the `_IOE_PROTOCOL_RPC=ON` environment variable) along with the SMB protocols (enabled by using the `_IOE_PROTOCOL_SMB=ON` environment variable).

**Note:** Dynamic export is not supported when `_IOE_PROTOCOL_RPC=ON`.

### Fileset IDs in dfstab

The `dfstab` and `devtab` files specify HFS and RFS file systems that are being exported. Those files are used for the SMB protocol and the DCE DFS protocol.

When you use the DFS protocol, fileset IDs must be assigned by the `flserver`. (This is required whether SMB protocols are being used or not.) For an HFS fileset, a fileset ID is assigned by the `flserver` by using the `fts crldbentry` command. The administrator enters the assigned fileset ID in the fileset's `dfstab` entry.

If you have been using DCE DFS protocols and you are now adding SMB protocols, the fileset IDs for any exported HFS or RFS filesets have already been assigned by the `flserver` and are already specified in the `dfstab`. The Distributed File Service SMB Server exports these same filesets for the SMB protocol. You can then create your `smbtab` entries to specify which shared directories should be available to PC clients.

If you need to export any additional HFS or RFS filesets, the fileset IDs must be assigned by the `flserver` (even if they are only going to be accessed by SMB protocols).

Or, if you have been using SMB protocols and you are now adding DCE DFS protocols, the HFS and RFS fileset IDs specified in the `dfstab` must be changed to numbers that have been assigned by the `flserver`. This means that you must enable and start the Distributed File Service Server for DCE DFS protocols, assign the HFS and RFS fileset IDs by using the `fts crldbentry` command, update your `dfstab` entries for the HFS and RFS fileset IDs, and then export the filesets (use the `modify dfs,start export` system command or the `dfsexport` command). If you need to add any HFS or RFS filesets, their fileset IDs must also be assigned by the `flserver`.

### Crossing local mount points

When the SMB protocol is used (without the DCE DFS protocol), PC users cross local mount points. That is, when a PC user changes the current directory (for example, `dirA`) to a subdirectory (for example, `dirB`) by using the `cd` command and that subdirectory (`dirB`) is a mount point, the PC user “crosses” into the root directory of the file system that is mounted on `dirB`. A `dir` command against `dirB` shows the files and directories contained in the root directory of the file system mounted on `dirB` (as opposed to showing the files and directories that might actually be contained in `dirB`). This assumes that the file system has been exported and that the user is authorized. In fact, any files or subdirectories that are actually contained in `dirB` are inaccessible. This is normal behavior for a z/OS UNIX file system.

When the DCE DFS protocol is used (regardless of whether the SMB protocol is used), clients do not cross mount points. Rather, clients see the files and directories that are actually contained in the mount point directory. This is how DFS clients expect the server to perform, and is how the server has performed in previous releases.

It is possible for SMB clients to cross local mount points and for DCE DFS clients to not cross local mount points. In fact, this is the default behavior. Whether SMB clients cross local mount points is controlled by the `_IOE_SMB_CROSS_MOUNTS` environment variable of `dfskern`. © Copyright IBM Corp. 1999, 2009
SMB encrypted passwords and DCE single sign-on

If you are using SMB encrypted passwords and DCE single sign-on, your SMB password and your DCE password must be the same since both of these come from the RACF DCE segment. However, if the DCE Security Server is running on the same system as the DCE user, then the DCE single sign-on processing does not require the DCE password to be stored in RACF. So, in this case, the SMB password can be stored in the RACF DCE segment and the RACF DCE segment is not used for DCE single sign-on.
Appendix D. Customizable files

Notes:
1. The symbolic link /opt/dfsglobal refers to the directory /usr/lpp/dfs/global.
2. The symbolic link /opt/dfslocal refers to the directory /etc/dfs.

The following table summarizes the Distributed File Service example files that are supplied by IBM and where they are placed so that you can customize them to meet your local DCE DFS or SMB support requirements. The table indicates whether the file applies to (DCE) DFS, SMB, or both. Files that apply to support not being used are created but do not need to be modified.

Most of the IBM supplied files are copied from the /opt/dfsglobal/examples directory to the target subdirectory in /opt/dfslocal (/etc/dfs) by the /opt/dfsglobal/scripts/dfs_cpfiles program run as part of the Distributed File Service post installation processing described in Chapter 3, “SMB post installation processing,” on page 11.

The dfs_cpfiles program does not replace a file in the target directory if it already exists. When installing a new Distributed File Service release, you can compare the contents of the IBM supplied files with your current customized files to determine if there are any new optional parameters that you may want to consider specifying.

The customizable files used for a previous release can be used for a new release if the same IP address, RACF user definitions and exported file definitions apply to the target image for the new release. If any of these definitions are not the same for the target image, the customizable files should be reviewed and modified as required. Refer to the [z/OS Distributed File Service Customization](#) document for more information.

Table 8. DFS customizable files

<table>
<thead>
<tr>
<th>IBM Supplied File</th>
<th>Customizable File</th>
<th>Applies To</th>
</tr>
</thead>
<tbody>
<tr>
<td>/opt/dfsglobal/examples/bakserver.envar</td>
<td>/opt/dfslocal/home/bakserver/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/boserver.envar</td>
<td>/opt/dfslocal/home/boserver/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/butc01.envar</td>
<td>/opt/dfslocal/home/butc01/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/butc02.envar</td>
<td>/opt/dfslocal/home/butc02/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/butc03.envar</td>
<td>/opt/dfslocal/home/butc03/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/butc04.envar</td>
<td>/opt/dfslocal/home/butc04/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/butc05.envar</td>
<td>/opt/dfslocal/home/butc05/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/butc06.envar</td>
<td>/opt/dfslocal/home/butc06/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/butc07.envar</td>
<td>/opt/dfslocal/home/butc07/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/butc08.envar</td>
<td>/opt/dfslocal/home/butc08/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/cmattr</td>
<td>CMATTR (not created by dfs_cpfiles; cmattr file defined by envar _IOE_CM_ATTRIBUTES_FILE)</td>
<td>DFS</td>
</tr>
<tr>
<td>(applies to both cmattr and hfsattr)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/daemonct.envar</td>
<td>/opt/dfslocal/home/daemonct/envar</td>
<td>both</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/devtab</td>
<td>/opt/dfslocal/var/dfs/devtab</td>
<td>both</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/dfs.ioepdcf</td>
<td>/opt/dfslocal/etc/ioepdcf</td>
<td>both</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/dfs.CachInfo</td>
<td>/opt/dfslocal/etc/CachInfo</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/dfs_envar</td>
<td>/opt/dfslocal/home/dfs/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/dfsCntl.envar</td>
<td>/opt/dfslocal/home/dfsCntl/envar</td>
<td>both</td>
</tr>
<tr>
<td>/opt/dfsglobal/examples/dvsexport.envar</td>
<td>/opt/dfslocal/home/dvsexport/envar</td>
<td>both</td>
</tr>
<tr>
<td>IBM Supplied File</td>
<td>Customizable File</td>
<td>Applies To</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>(None)</td>
<td>DFSIDMAP</td>
<td>DFS</td>
</tr>
<tr>
<td></td>
<td>(not created by dfs_cpfiles; dfsidmap file defined by envar _IOE_MVS_IDMAP)</td>
<td></td>
</tr>
<tr>
<td>/opt/dfs/global/examples/dfskern.envar</td>
<td>/opt/dfslocal/home/dfskern/envar</td>
<td>both</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/dfstab</td>
<td>/opt/dfslocal/var/dfs/dfstab</td>
<td>both</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/flserver.envar</td>
<td>/opt/dfslocal/home/flserver/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/ftserver.envar</td>
<td>/opt/dfslocal/home/ftserver/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/growaggr.envar</td>
<td>/opt/dfslocal/home/growaggr/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/cmattr</td>
<td>HFSATTR</td>
<td>both</td>
</tr>
<tr>
<td>(applies to both cmattr and hfsattr)</td>
<td>(not created by dfs_cpfiles; hfsattr file defined by envar _IOE_HFS_ATTRIBUTES_FILE)</td>
<td></td>
</tr>
<tr>
<td>/opt/dfs/global/examples/newaggr.envar</td>
<td>/opt/dfslocal/home/newaggr/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/repserver.envar</td>
<td>/opt/dfslocal/home/repserver/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/rfstab</td>
<td>/opt/dfslocal/var/dfs/rfstab</td>
<td>both</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/salvage.envar</td>
<td>/opt/dfslocal/home/salvage/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>(None)</td>
<td>SMBIDMAP</td>
<td>SMB</td>
</tr>
<tr>
<td></td>
<td>(not created by dfs_cpfiles; smbidmap file defined by envar _IOE_SMB_IDMAP)</td>
<td></td>
</tr>
<tr>
<td>/opt/dfs/global/examples/smbtab</td>
<td>/opt/dfslocal/var/dfs/smbtab</td>
<td>SMB</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/upclient.envar</td>
<td>/opt/dfslocal/home/upclient/envar</td>
<td>DFS</td>
</tr>
<tr>
<td>/opt/dfs/global/examples/upserver.envar</td>
<td>/opt/dfslocal/home/upserver/envar</td>
<td>DFS</td>
</tr>
</tbody>
</table>
Appendix E. Using data sets

This appendix explains what you need to know when you use data sets from a PC client. The SMB server can export record files (sequential, PDS, PDSE, and Virtual Storage Access Method (VSAM)). This makes record data available to PC client applications. The data is presented as byte stream data.

**Note:** Generation Data Groups (GDG) and multi-volume data sets (except for striped) are not supported.

Mapping between the PC client’s view and record data

In z/OS, a file is called a data set. These two terms are used interchangeably.

The files for a computer system are organized into a file system. The PC client environments use a byte file system which is a hierarchy of directories that can contain other directories or files. Record data, however, uses a nonhierarchical file system in which groups of data sets are referred to by specifying a high-level qualifier.

The high-level qualifier can include the first (leftmost) qualifier of data sets, or the first and second qualifiers, or the first, second, and third qualifiers, and so on. For example, SMITH is the high-level qualifier for the files named SMITH.TEST.DATA and SMITH.PROJ7.SCHED, while SMITH.TEST is the high-level qualifier of SMITH.TEST.DATA and SMITH.TEST.Docs.

**Note:** Only Integrated Catalog Facility (ICF) cataloged data sets are supported by the SMB server. Tape data sets are not supported.

The SMB server exports a set of files with a specified prefix as a single file system. All files with the specified prefix are shown as one level of hierarchy. If the data sets specified include partitioned data sets, a second level of hierarchy is shown.

Since RACF requires a data set profile that covers the prefix specified on the devtab, you must specify two levels of prefix to have it covered by a profile. For example, prefix USERA.PCDSNS is covered by data set profile 'USERA.*'. Otherwise, you must give the user the OPERATIONS attribute.

Mapping data sets onto an RFS file system

If the following data sets exist:

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERA.PCDSNS.B()</td>
<td>with members M1 and M2</td>
</tr>
<tr>
<td>USERA.PCDSNS.B.C</td>
<td></td>
</tr>
<tr>
<td>USERA.PCDSNS.B.C.D</td>
<td></td>
</tr>
<tr>
<td>USERA.PCDSNS.X.Y()</td>
<td>with members M1 and M2</td>
</tr>
<tr>
<td>USERA.PCDSNS.X.Y.Z</td>
<td></td>
</tr>
</tbody>
</table>

The following is an example of SMB server commands and operations to export an RFS file system that contains data sets that begin with the prefix USERA.PCDSNS.

1. Add an **rfs** file system to the **devtab** file.
   ```bash
   * RFS devices
   define ufs 3 rfs
   USERA.PCDSNS
   ```

2. Add the **rfs** file system to the **dfstab** file using the same minor device number (in this example, 3) in the device parameter (so in this example, it would be /dev/ufs3). Use a unique file system name (for example, rfs3), a file system type of **ufs** and a unique file system ID (for example, 102). Finally, use a unique filesystem ID (for example, 0.,1718). An example **dfstab** entry might look like this:
   ```bash
   /dev/ufs3 rfs3 ufs 102 0.,1718
   ```

© Copyright IBM Corp. 1999, 2009
Note: If you are using only SMB protocols (and not DCE DFS protocols), you can use the same number for all the numeric values in a dfstab entry as long as the number used is unique. For example, the dfstab entry might look like this:

/dev/ufs3  rfs3  ufs 3 0,,3

If you are using both SMB protocols and DCE DFS protocols, the file system ID must be assigned by the fts crfldbentry command. See Appendix C, “Using both SMB and DCE DFS,” on page 169 for more information.

3. Add an entry in smbtab for the directory you want to share. Use the same minor device number (in this example, 3) in the device name parameter (so in this example, it would be /dev/ufs3). Choose a unique share name (for example, mvsusera), a file system type of ufs, a description, share permission (for example, r/w), maximum users (for example, 100) and the directory name (in this example /). The directory name is relative to the root of the file system that the device name refers to.

   dev/ufs3  mvsusera  ufs  "mvsusera data sets"  r/w  100  /

Note: RFS has many restrictions when writing to or creating files. If it is appropriate, you may want to specify read-only access to avoid these restrictions by using r/o in the share permission field of the smbtab entry.

4. Issue the following dfsshare command to cause the new share to be made available to PC users.

   # dfsshare -share mvsusera

At this point, a PC user can connect to this share.

5. The PC user can now connect to the shared directory using the following net use command.

   C:\>net use r: \computer1.abc.com\mvsusera password

   The following figure represents a view of the data set hierarchy.

```
   r:|
   |  b  b.c  b.c.d  x.y  x.y.z
   |    m1   m2   m1   m2
```

You can define one level of directory under the shared directory (at r:). Thus, you can issue mkdir to create a directory (stored as a PDS or PDSE) and then create files (stored as PDS or PDSE members) within that directory.

Alternatively, if the prefix specified in the devtab is an actual data set, the SMB server attempts to export the data set. If the data set is a PDS (or PDSE), it is handled as a directory and the members are exported. However, in this case, sharing with local z/OS record applications is not supported. That is, the data set remains allocated the entire time that the data set is exported. Refer to "Sharing data" on page 173 for more information. If it is any other type of data set, the export for that data set fails. (The name that is specified for export must be able to be handled as a directory.)

Reading, writing, and creating data sets

You can use the SMB server to read data stored in a data set from your PC client system. You can also write data to a data set (stored on DASD) from your PC client system. The data sets appear as files on the drive letter on your PC client system.
You can create data sets from a PC client using the SMB server. The default record data set creation attributes specified by the system administrator are used to create data sets, unless the user overrides them. These attributes determine how the data set is structured and where it is stored. You can override the data set creation attributes at file creation time.

Sharing data

The sharing semantics that local z/OS applications accessing record data expect and the semantics that PC clients expect are very different. PC clients can potentially allow multiple users to be able to open a file for write. The file integrity is protected through the use of byte range locks. Local z/OS applications, in general, do not expect other users to be writing to the file (data set) while their application is writing.

Also, SMB clients request opportunistic locks when opening a file. Possession of an exclusive opportunistic lock means that the client can act on the file without communicating to the SMB server. The SMB server keeps track of opportunistic locks it has given out. When a PC client requests an opportunistic lock that conflicts with another opportunistic lock that the SMB server has given out, the SMB server issues a callback (requests that it be given back to the server) for that opportunistic lock. PC clients that receive callbacks give back the opportunistic lock as soon as they can. PC clients expect that all requesters are properly requesting and returning opportunistic locks. Local z/OS applications using record data sets do not request opportunistic locks.

As a result, the SMB server must provide one set of semantics and opportunistic lock management to the PC clients and another set of semantics with no opportunistic lock management to record data applications. In order to provide this, the SMB server must construct a “wall” between the PC clients and the record data applications that are attempting to access the same record file. PC clients or record applications can write the data, but not both. In some cases, they can both read the data.

Note: None of the following techniques for freeing the data set are supported when an actual data set (as opposed to a data set prefix) is specified in the devtab. This means that the RFS file system must be unexported to permit a batch job that needs access to the data set to continue. For example, the z/OS UNIX command `dfsexport /dev/ufs4 -detach` would unexport the data set (assuming that the devtab entry for the data set begins with `/dev/ufs4`).

The general technique that the SMB server uses to accomplish this is to do a dynamic allocate on any record data set that a PC client is attempting to access.

1. If the PC client is attempting to read record data, and the data set is not a PDS, the SMB server attempts a DISP=SHR allocation for the data set.
   a. If the data set is free, then the dynamic allocate issued by the SMB server is successful and the SMB server provides the record data set to its PC clients for reading.
      A batch job that subsequently attempts to access that same record data set with an allocation of DISP=SHR successfully accesses the data set.
      A batch job that subsequently attempts to access that same record data set with an allocation of DISP=OLD waits for the unallocate to occur. The SMB server is notified that a batch job is waiting for the unallocate. (The SMB server uses the Event Notification Facility (ENF) and an event supported by Global Resource Serialization (GRS) for resource contention. Refer to [z/OS MVS Programming: Authorized Assembler Services Reference ALE-DYN](SA22-7608) for information on the Event Notification Facility.) In this case, the SMB server frees up the record data set as soon as it can.
      b. If the data set is not free, the SMB server denies access to the data set. It appears to the PC clients that the data set is unavailable. (PC clients cannot wait an indefinite amount of time for the batch job to complete.)

2. If the PC client is attempting to write record data, or the data set is a PDS, the SMB server attempts a DISP=OLD allocation for the data set.
a. If the data set is free, then the dynamic allocate issued by the SMB server is successful and the SMB server provides the record data set to its PC clients for writing.

A batch job that subsequently attempts to access that same record data set with an allocation of DISP=SHR or DISP=OLD waits for the unallocate to occur. The SMB server is notified that a batch job is waiting for the unallocate. In this case, the SMB server frees up the record data set as soon as it can.

b. If the data set is not free, the SMB server denies access to the data set. It appears to the PC clients that the data set is unavailable. (PC clients cannot wait an indefinite amount of time for the batch job to complete.)

ISPF users attempting to access a record data set that has been accessed by PC clients may be denied access because the SMB server may still have the data set allocated. ISPF (and TSO users) use dynamic allocation and this does not cause resource contention (and therefore no resource contention event occurs). The allocation request is simply denied. The SMB server does not recognize that the data set is being requested for use. However, the SMB server deallocates data sets that have had no activity for a period of time. This makes the data sets available to other applications such as ISPF. The time period is controlled by the _IOE_RFS_ALLOC_TIMEOUT environment variable. The default time period is 5 minutes. Refer to page 150 for more information on this environment variable.

In order to free up a data set right away, the user can submit a simple batch job that allocates the data set. This causes resource contention and therefore causes the SMB server to free up the data set as soon as it can. The batch job can be as simple as:

```plaintext
//JOBNAME JOB
//STEP1 EXEC PGM=IEFBR14
//DD1 DD DSN=USERA.PCDSNS.B.C,DISP=OLD
```

**Note:** If you are using JES3 to protect the data sets that are being shared with PC clients, resource contention does not occur (and therefore no resource contention event occurs). The SMB server does not recognize that the data set is being requested for use. In this case you could request that JES3 bypass these data sets (by using the DYNAALDSN statement). Refer to the z/OS JES3 Initialization and Tuning Reference for information on the DYNAALDSN statement. Alternatively, you can use one of the other techniques that follow to free up the data set.

Another technique can be used by PC client users with the proper authority to force the SMB server to free up a data set. A user with RACF ALTER authority to the data set can issue the `mkdir "name,release"` command from a PC client machine. For example, `mkdir "b.c,release"` would cause the SMB server to free up the USERA,PCDSNS,B.C data set as soon as it can. (If `name` is a member of a PDS or PDSE data set, the entire PDS or PDSE data set is released.)

When you issue the `mkdir release` command, a `mkdir` error message is expected. If the release was successful, the message indicates that the system attempted to access the file but the system could not because the object already exists. For example,

```
E:\>mkdir "r:\b.c,release"
A subdirectory or file r:\b.c,release already exists.
```

This is because the PC client sends this request as a make directory request. Since we are not really making a directory at the server (rather, we are doing release processing against a file or set of files), we must send an error code indicating that the object already exists back to the PC client when the release processing is completed successfully to keep the PC client from continuing with its create directory processing.

If the user does not have RACF ALTER authority to the file(s), the SMB server sends back EACCES (the user does not have the required permission) to the PC client. For a Linux Samba client, the `mkdir release` command can also be used. For example:

```
linux001:/mntfrs # mkdir "b.c,release"
mkdir: cannot create directory 'b.c,release': File exists
```

176  z/OS V1R11.0 Distributed File Service SMB Administration
An operator force command (**modify dfs**) is provided to force the SMB server to free up a data set if an important batch job has been waiting too long.

**Forcing a data set to be freed by SMB**

The following operator command may be issued to force the SMB server to free a data set for access by a batch job:

```
modify dfs,send dfskern,release,data-set-name
```

where *data-set-name* is the name of the data set that the SMB server should make available.

---

**Refreshing RFS file names**

The SMB server uses the Event Notification Facility to determine when a batch job is attempting to allocate a data set that is currently being accessed by PC clients. When the SMB server detects this, it issues callbacks for all opportunistic locks and unallocates the data set (thus making it available to the batch job).

Because there is no event to indicate when data sets are created, deleted, or renamed by a batch job, the SMB server refreshes its cache of exported data set names and attributes at a regular interval. This interval is controlled by the **_IOE_RFS_STATUS_REFRESH_TIME** environment variable in the dfskern process. The time is specified in seconds and the default is 600 (ten minutes). In this case, a PC client listing file names when positioned at an *rfs* root directory may not see a new file created by a z/OS job for up to ten minutes after it is created.

This only affects commands that list file names and attributes. If a PC client creates an RFS file, that file can be directly referenced immediately after creation. Data sets or files created by a TSO user, a batch job or any z/OS application are not visible or immediately accessible to the PC client, until the deallocation time period (**_IOE_RFS_ALLOC_TIMEOUT** and the refresh time interval (**_IOE_RFS_STATUS_REFRESH_TIME**) expire. In addition, if the data set or file remains opened by the z/OS user, batch job, or application after it is created, it is not visible or accessible to the PC client, until it is closed and both the deallocation time period and the refresh time interval expire. For details about the **_IOE_RFS_ALLOC_TIMEOUT** and **_IOE_RFS_STATUS_REFRESH_TIME** environment variables, see Appendix A, “Environment variables in SMB,” on page 143.

---

**Special considerations for record data**

In addition to mapping between the PC client’s view and z/OS file systems, you should be aware of the other ways in which the record data might differ from hierarchical byte data. These differences include:

- Selecting a data storage format
- File size determination and time stamps
- Client caching
- Record file names.

**Selecting a data storage format for record data**

PC clients can access data sets. These data sets are record oriented and can be sequential, direct, VSAM, partitioned, and so forth, and also can contain variable or fixed-length records. PC client environments, however, are byte-oriented and may write or read at certain byte offsets in the file.

The SMB server can map PC client requests to most types of data set organizations. However, how the time stamps and file size value are handled depends on the type of data set used, and the file size processing can affect performance.

Direct reads with the data set attributes **recfm(fbs)** or **recfm(f)** can be fast because in some cases, the SMB server can determine the physical block addresses from the record offsets. The z/OS sequential file
organization with recfm(f) or recfm(fbs) on DASD allows for efficient updating or reading at any offset in the file. Other supported z/OS access methods (for example, VSAM) may not perform as efficiently.

**File size determination and time stamps**
The SMB server determines how to handle the file size value and time stamps depending on the type of data set used and the attributes used to access the data set. Refer to “Handling of the file size value” on page 185 for more information of file size determination and to “Handling of the time stamps” on page 186 for more information on handling time stamps.

**PC client caching**
Record file data is cached at PC clients in the normal manner. PC clients request and return opportunistic locks as usual. However, if the SMB server fails and restarts, and you are currently positioned within an RFS file system, then you need to change directory (cd) back to the root of that RFS file system before you can access files within that RFS file system.

**Record file names**
As explained in “Mapping between the PC client’s view and record data” on page 173, record file names consist of segments separated by a dot with a maximum length of 44 characters. Each segment can be from one to eight alphanumeric characters. (Refer to z/OS DFSMS Using Data Sets, SC26-7410, for information on data set naming.) Each record file appears as a byte file to PC clients. A PDS appears as a directory with the members appearing as files within the directory. PDS member names are limited to eight characters.

Data set names are always in upper case characters. This means that file names would be displayed to PC client users in upper case and would need to be entered in upper case. There is, however, a processing attribute in the attributes file (refer to “Attributes file (rfstab)” on page 84 called maplower. maplower is the default. When this attribute is specified or defaulted in the attributes file, users can enter lower case letters for file names and they are mapped to upper case by the SMB server. Also, when the (upper case) file names are displayed to the user, they are mapped to lower case by the SMB server.

You are cautioned, however, that when the maplower processing attribute is in effect, you should only use lower case letters for file names. If you create a file with the name AbCd, it becomes a data set named ABCD. When the file name is displayed to the (PC client) user, it appears as abcd. The file name would be displayed as a different name than the user entered when it was created.

Also, many byte file systems typically have file names that begin with a dot. This is an invalid name for a data set. However, a processing attribute in the attributes file called mapleaddot causes a leading dot in a file name to be mapped to a dollar sign in the data set name and back to a dot for the PC client. mapleaddot is the default.

**Creating z/OS files**
This section describes how to create the various types of data sets (files) supported by the SMB server.

The examples shown are for an MS-DOS session. Any examples for other platforms have been indicated.

**Overriding data set creating attributes**
When you create a z/OS file, default file creation attributes are applied, unless you override them. The attributes are passed to the z/OS host.

Data set creation attributes are controlled in the following ways, in increasing order of priority:
- Default server data set creation attributes (refer to “Attributes file (rfstab)” on page 84
Default installation data set creation attributes, specified by the system administrator in the attributes file (refer to “Attributes file (rfstab)” on page 84)
DFSMS data class attributes
Data set creation attributes specified for the fileset in the devtab file (refer to “devtab” on page 91)
Data set creation attributes specified at file creation, for example, in the mkdir, notepad (edit), or copy commands (highest priority).

Note: These data class attributes are not supported by the SMB server:
- Retention period/Expiration date
- Number of volumes
- VSAM imbed index option
- VSAM replicate index option
- CI size of data component
- Percentage of CI or CA free space.

Preparing to create a z/OS file
When you create a z/OS file, you may want to specify what type of file to create.

These following types of files are supported by the SMB server:
- Physical sequential (PS)
- Direct access (DA)
- Partitioned data sets (PDS)
- Partitioned data sets extended (PDSE)
- VSAM KSDS
- VSAM ESDS
- VSAM RRDS
- Sequential Access Method (SAM) extended format data sets.

Notes:
1. Keyed access to files is not supported by PC clients.
2. You can only share a PDSE between systems in the same parallel sysplex; z/OS does not support sharing PDSE between sysplexes.

Naming z/OS files:
When naming conventional z/OS files, you must follow the file naming conventions, as described in Z/OS DFSMS Using Data Sets.

A file name (or data set name) can consist of one or several simple names joined so that each represents a level of qualification. For example, the file name DEPT58.SMITH.DATA3 is composed of three qualifiers.

The following characteristics apply to the file name:
- Each qualifier consists of 1 to 8 alphanumeric characters, national characters (@, #, $), or a hyphen (-).
- Each qualifier must start with an alphabetical or national character.
- The period (.) separates simple names from each other.
- Including all simple names and periods, the length of the file name must not exceed 44 characters. Note that the high-level qualifier that was exported (in the previous example, USERA.PCDSNS) is added as a prefix to the file name. So, if you created file r:\b.c.d.e, the SMB server would create data set USERA.PCDSNS.B.C.D.E.
- PDS and PDSE member names can be up to 8 characters long.

Restrictions using alias names for z/OS files: For PDSs and PDSEs, alias names (for member names) are not supported. They are not displayed when you list the names in a directory (that is really a PDS or PDSE). You cannot access a file (member) by its alias name. If you try to create a file in the directory that has the same name as an alias, it is denied.
Creating physical sequential files

Note that the following examples assume that an rfs fileset is mapped to the r: drive and the data set prefix in devtab is SMITH. It is also assumed that these files are translated between ASCII and EBCDIC characters by an administrator specification of:

- _IOE_RFS_TRANSLATION=ON, or
- Text in the devtab entry for rfs fileset, or
- Text in the attributes file for the rfs fileset.

When creating a physical sequential (PS) file, specify the dsorg(ps) attribute (if it is not the default already) with a file creation command, such as the notepad command.

C:\>notepad "r:\new.txt,dsorg(ps)"

When you save the file using notepad, you have just created a new PS file named SMITH.NEW.TXT.

You must specify the .txt suffix on the file name. If you do not, notepad will automatically append it and will try to create "new,dsorg(ps).txt", which will not be successful. You must also type a carriage return after the last line (or the only line) of the file. Notepad does not do this automatically.

An example for Linux Samba follows:

linux001:/mntrfs # vi "new.txt,dsorg(ps)"

When reading or changing data in a Physical Sequential file with fixed-length records in text mode, the blankstrip processing attribute in the attributes file controls how trailing blanks are handled. If blankstrip is specified in the attributes file, trailing blanks are removed from the end of each record when it is read, and blanks are padded to the end of each record when it is written. blankstrip is the default. If noblankstrip is specified in the attributes file, trailing blanks are not removed from the end of each record when it is read, and blanks are not padded to the end of each record when it is written. In this case, each record written must be the correct size or an I/O error is reported.

Note: When copying a file to an RFS file system, if you are overriding the data set creation attributes, you must specify the target file name in the copy command. For example:

C:\>copy file1 "r:\newfile1,space(2,5),cyls"

The previous example is correct.

C:\>copy file1 "r:\,space(2,5),cyls"

The previous example is incorrect.

Examples for Linux Samba are:

linux001:/mntrfs # cp file1 "newfile1,space(2,5),cyls"

The previous example is correct.

linux001:/mntrfs # cp file1 ",space(2,5),cyls"

cp: cannot create regular file ",space(2,5),cyls": Invalid argument

The previous example is incorrect.

Creating direct access files

When creating a direct access (DA) file, specify the dsorg(da) attribute (if it is not the default already) with a file creation command, such as the notepad command.

C:\>notepad "r:\new.txt,dsorg(da),blksize(80),lrecl(80),recfm(f)"

You have just created a new DA file named SMITH.NEW.TXT
You must specify the .txt suffix on the file name. If you do not, notepad will automatically append it and will try to create "new,dsorg(da),blksize(80),lrecl(80),recfm(f).txt", which will not be successful. You must also type a carriage return after the last line (or the only line) of the file. Notepad does not do this automatically.

An example of Linux Samba follows:

```
linux001:/mntrfs # vi "new.txt,dsorg(da),blksize(80),lrecl(80),recfm(f)"
```

When reading or changing data in a Direct Access file with fixed-length records in text mode, the blankstrip processing attribute (in the attributes file) controls how trailing blanks are handled. If blankstrip is specified or defaulted to in the attributes file, trailing blanks are removed from the end of each record when it is read, and blanks are padded to the end of each record when it is written. If noblankstrip is specified in the attributes file, trailing blanks are not removed from the end of each record when it is read, and blanks are not padded to the end of each record when it is written. In this case, each record written must be the correct size or an I/O error is reported.

### Creating PDSs and PDSEs

Partitioned data sets (PDSs) and partitioned data sets extended (PDSEs) can be used as directories, and their members are files within those directories. An illustration of the use of PDSs to act as directories is shown on page. For general information on PDSs and PDSEs, refer to [z/OS DFSMS Using Data Sets](#).

**Notes:**
1. You cannot create new directories within a PDS or PDSE, due to the nature of these data structures.
2. You can only share a PDSE between systems in the same parallel sysplex; z/OS does not support sharing PDSE between sysplexes.

#### Creating a PDS or PDSE -- mkdir dsntype(pds), dsntype(library):

To create a PDS or PDSE, perform the following steps:

1. If creating a PDSE, use the `mkdir` (make directory) command specifying the `dsntype(library)` attribute to create a PDSE named `smith.datalib`:
   ```
   C:\>mkdir "r:\datalib,dsntype(library)"
   ```
   If creating a PDS, use the `mkdir` (make directory) command specifying the `dsntype(pds)` attribute as follows:
   ```
   C:\>mkdir "r:\datalib,dsntype(pds),dir(20)"
   ```

   **Note:** You can omit specifying the `dsntype(pds)` attribute, if `pds` has been specified for the `dsntype` attribute in the attributes file on page 85.

   Examples for Linux Samba follows:
   ```
   linux001:/mntrfs # mkdir "datalib,dsntype(library)"
   linux001:/mntrfs # mkdir "datalib,dsntype(pds),dir(20)"
   ```

2. You can use the `copy` command to create a PDS or PDSE member named `smith.datalib(member1)`:
   ```
   C:\>copy c:\otherfile.txt r:\datalib\member1
   ```
   An example for Linux Samba follows:
   ```
   linux001:/mntrfs # cp otherfile.txt datalib/member1
   ```

You have now created a PDS or PDSE member. You can use the `type` command (the `cat` command for Linux) to view the contents of your PDS or PDSE member.

**Note:** The SMB server supports a maximum of 14,562 members in a PDS or PDSE data set. When a PC read-directory request on a PDS or PDSE is processed, the SMB server returns up to 14,562 member names. Other requests, such as read and write, to individual members are not affected.
Removing a PDS or PDSE -- erase, rmdir:
To remove a PDS or PDSE, first make sure that the PDS or PDSE is empty. You can delete all members under the directory using the `erase` command. Then use the `rmdir` (remove directory) command. This example removes the `datalib` directory, and confirms its removal by a failed try to query it (dir is the list files command):

```
C:\>dir r:\\DATALIB
Volume in drive R has no label.
Volume Serial Number is 0000-0000
Directory of r:\datalib
06/29/00 09:04p <DIR>  .
06/29/00 10:51a <DIR>  ..
08/01/96 12:19p 298 butcvsmp
02/17/95 02:13p 1,019 copyunld
09/20/96 09:20a 298 butcvsmp
09/20/96 07:50a 547 testaloc
07/02/99 06:49a 0 testtext
07/02/99 07:56a 38 testtxt3
09/20/99 07:50a 547 testaloc
9 File(s) 3,000 bytes
61,440,000 bytes free
C:\>erase r:\datalib\*
C:\>rmdir r:\datalib
C:\>dir r:\datalib
File Not Found
```

An example of Linux Samba follows:

```
linux001:/mntrfs # ls datalib
. .. butcvsmp copyunld su2aloc test testtext testtxt3 testaloc
linux001:/mntrfs # rm datalib/*
linux001:/mntrfs # rmdir datalib
linux001:/mntrfs # ls datalib
/bin/ls: datalib: No such file or directory
```

Accessing PDS or PDSE members:
There is more than one way to access PDS and PDSE members. For example, you could display the existing PDS member `smith.source(bigblue)` by entering either of these command sequences:

```
C:\>type r:\source\bigblue

or

C:\>cd r:\source
C:\>type r:bigblue
```

These two approaches are equivalent.

Examples for Linux Samba follows:

```
linux001:/mntfs # cat source/bigblue

or

linux001:/mntfs # cd source
linux001:/mntfs/source # cat bigblue
```

Updating or extending a PDS or PDSE member
The SMB server does not generally support updating or extending a PDS or PDSE member directly. To update or extend a PDS or PDSE member, a client program must follow these steps:

1. Copy the file to the client machine
2. Update or extend the copied version on the local system
3. Truncate the original file to zero size by sending a SETATTR request with zero file size
4. Copy the updated version on the local host to z/OS by writing request.

Some client editors follow the above steps, for example, the AIX and z/OS UNIX vi editor. Other editors do not follow the above steps, for example, the Notepad editor. In the latter case the user must save the updated version into a new file. You cannot save the updated version into the PDS or PDSE, because Notepad will attempt to append .txt to the member name. You must save it somewhere else (for example, c:saved_updates.txt) and then copy it into a new member in the PDS. For example:

C:\>copy c:saved_updates.txt r:\datalib\member2

When reading or changing data in a PDS member or PDSE member with fixed-length records in text mode, the blankstrip processing attribute in the attributes file controls how trailing blanks are handled. If blankstrip is specified in the attributes file, trailing blanks are removed from the end of each record when it is read, and blanks are padded to the end of each record when it is written. blankstrip is the default. If noblankstrip is specified in the attributes file, trailing blanks are not removed from the end of each record when it is read, and blanks are not padded to the end of each record when it is written. In this case, each record written must be the correct size or an I/O error is reported.

Renaming or moving a PDS or PDSE member:
Record file system (RFS) files and directories can only be renamed or moved in a way that does not cause them to be moved from one directory to another.

If you are writing to a PDS or PDSE member and a timeout occurs, the timeout causes the member to close. The remaining write requests appear to append to a PDS or PDSE member. This operation is not supported and causes an I/O error. To avoid timing out, increase the timeout setting.

Wildcard copy to a PDS or PDSE:
To ensure that a wildcard copy, copy c:\mydir\* r:\source, to a PDS or PDSE can be completed successfully, a prior PDS member is closed and dequeued (if necessary) to allow the creation of a new member.

Limitations of writing to a PDS:
The PDS support in the SMB server adheres to the conventions used in z/OS. For example, you cannot have more than one member of a PDS open for writing at a time. If you try to write to a member of a PDS while another member is open for write by a different user, you get a “*** Permission denied” message.

A PDS member stays open for the timeout period specified in the appropriate timeout processing attribute, filetimeout, or until you try to create or write to another member.

Concurrent writes to a PDSE:
The SMB server supports concurrent writes to a PDSE. If you are writing to one member of a PDSE, another client can write to any other member in the same PDSE.

Creating VSAM files
The SMB server supports three types of VSAM files:
- key-sequenced (KSDS)
- entry-sequenced (ESDS)
- relative record (RRDS).

However, keyed access and relative-number access to the files are not supported.

If you plan to update a VSAM data set (for example, with the notepad editor or with the copy command), the data set must have been defined with the REUSE option. Trying to write back a VSAM data set that was not defined as reusable results in an “I/O error”, “failure to open”, or similar error message. When you create a VSAM file through the SMB server, the REUSE option is always specified by the server.

For more information on VSAM files, refer to [z/OS DFSMS Using Data Sets](#).
In the following example, the attributes indicate that:
- Spanned records are allowed
- Organization is key-sequenced
- Keys are 8 bytes long and start in position 0 of each record
- Average record size is 1024
- Maximum record size is 4096
- Space is allocated for 50 records with a secondary allocation of 10
- Cross-region and cross-system share options are provided
- The file is to be created on a volume named D80CAT.

```
notepad "r:\ksds.new2.txt,spanned,dsorg(indexed),keys(8,0),recordsize(1k,4k),
space(50,10),shareoptions(1,3),vol(d80cat)"
```

An example of Linux Samba follows:

```
linux001:/mntrfs # vi "ksds.new2.txt,spanned,dsorg(indexed),keys(8,0),recordsize(1k,4k),
space(50,10),shareoptions(1,3),vol(d80cat)"
```

In the following example for creating a VSAM ESDS file, the attributes indicate that:
- Spanned records are allowed
- Organization is entry-sequenced
- Average record size is 1024
- Maximum record size is 4096
- Space is allocated for 50 records with a secondary allocation of 10
- Cross-region and cross-system share options are provided
- The file is to be created on a volume named D80CAT.

```
notepad "r:\esds.new3.txt,spanned,dsorg(nonindexed),recordsize(1k,4k),
space(50,10),shareoptions(1,3),vol(d80cat)"
```

An example of Linux Samba follows:

```
linux001:/mntrfs # vi "esds.new3.txt,spanned,dsorg(nonindexed),recordsize(1k,4k),
space(50,10),shareoptions(1,3),vol(d80cat)"
```

In the following example, the attributes indicate that:
- Spanned records are not allowed
- Organization is relative record, numbered in ascending order
- Average record size is 1024
- Maximum record size is 1024
- Space is allocated for 50 records with a secondary allocation of 10
- Cross-region and cross-system share options are provided
- The file is to be created on a volume named D80CAT.

```
notepad "r:\rrds.new4.txt,nonspanned,dsorg(numbered),recordsize(1k,1k),
space(50,10),shareoptions(1,3),vol(d80cat)"
```

An example of Linux Samba follows:

```
linux001:/mntrfs # vi "rrds.new4.txt,nonspanned,dsorg(numbered),recordsize(1k,1k),
space(50,10),shareoptions(1,3),vol(d80cat)"
```

**Specifying attributes multiple times**

Specifying an attribute several times on a line does not cause an error. The line is read from left to right, and the last of any duplicate attribute is used. For example:

```
C:\>notepad "r:\file,recfm(vb),recfm(fb)"
```

An example of Linux Samba follows:

```
linux001:/mntrfs # vi "file,recfm(vb),recfm(fb)"
```

This results in a file created with a fixed-blocked format.
Exploiting SAM striped files
With SAM striping, data I/O is done in parallel to improve performance. For a file with 16 stripes, data is processed on the first track of the allocated space on the first volume (that is, the first stripe), then on the first track of the second volume, and so on for all 16 volumes. Then, processing continues with the second track of all the volumes, then the third, and so on.

The SMB server can support data set striping through the use of data class and storage class attributes that define extended format data sets. The SMB server can exploit the performance of extended format data sets by reading multiple blocks at a time when reading ahead.

For more information on striped files, refer to [z/OS DFSMS Using Data Sets](#).

Handling of the file size value
Many file system commands (such as: `dir`) require the file size to be returned. This appendix explains some performance and accuracy considerations in obtaining the file size value.

The meaning of the file size value returned by the SMB server and how fast the file size is returned depends on:
- Whether you use text or binary processing mode
- The type of data set being accessed
- If the data set is system-managed
- Whether you use `fastfilesize` or `nofastfilesize` processing.

Storage of the file size value
Whether or not the file size value is already stored on your z/OS system, affects how quickly files are accessed and depends on the type of z/OS data set used.

System-managed PS, VSAM, and PDSE data sets:
For system-managed data sets, text and binary file size are saved on non-volatile storage (DASD) and maintained by the SMB server for the following data set types:
- Physical sequential (including striped)
- VSAM ESDS
- VSAM KSDS
- VSAM RRDS
- PDSE members.

When the SMB server accesses a data set for the first time, it performs a read-for-size to get the text or binary file size and stores this value on DASD. Subsequent file size requests from clients do not cause the server to read for size, thus improving performance. However, when the data set is modified outside the server by a non-PC application (for example, by the TSO editor), the stored file size could be incorrect. When the data set is accessed again by the server, read-for-size is done to determine the correct file size.

Migrated data sets
A data set that is migrated cannot be accessed. It must be recalled before it can be accessed. Whether a data set is recalled or not is based on the retrieve / noretrieve specification in the attributes file (`rfstab`). When a migrated data set is referenced, the request fails and the data set is recalled if retrieve (no wait) is specified in `rfstab`. Later, when the data set has been recalled, the data set can be accessed. If noretrieve is specified in `rfstab`, the request fails and the data set is not recalled.

Non-system managed, PDS, and DA data sets:
The file size value for non-system managed data sets, PDS members, and DA data sets is cached in virtual storage until released but not written to DASD. Therefore, for these types of data sets, the file size value is regenerated after the file is released or after the server is restarted.
How the file size is generated

When a file is first accessed (for example with `dir`), usually the entire file is read to determine its size, except for when specifying the `recfm(f)` or `recfm(fbs)` attributes where the binary size can be computed without reading the file. If the file is a system-managed PS, VSAM, or PDSE member, both binary and text file sizes are stored on DASD, so that subsequent file size requests do not require the file to be read.

Binary file size can be quickly generated by using `recfm(f)` or `recfm(fbs)` to specify a fixed-length record format for the data set. With this format type, the server pads the last logical record with binary zeros in binary mode processing, because z/OS always expects complete logical records. If the application tolerates these zeros, using `recfm(f)` or `recfm(fbs)` allows the binary size to be computed quickly because the number of bytes can be computed from the number of blocks, which is stored by z/OS.

If you need the exact file size and are using binary mode processing, map it to a variable-format, sequential data set on DASD so that the SMB server does not need to pad a partially filled last logical record to a record boundary.

For reading small files or the beginning of files, the read-for-size might not add any processing time. As the file is being read for size, the beginning of the file is stored in the buffers set aside by the `maxrdforzleft` site attribute, until the buffers are full. When the application reads the beginning of the file, this read is fast because it reads directly from the buffer.

z/OS stores the number of blocks (rather than the number of bytes) in a z/OS file. For most files, therefore, without reading the entire file, the SMB server can only give an estimate of the number of bytes in the file, not the exact number of bytes in the file.

Using fastfilesize to avoid read-for-size:

If you can use an approximate file size for a PDS, PDSE, DA, or non-system managed data set, you can specify the `fastfilesize` attribute to improve performance. With this attribute, the server estimates the size without opening and reading the entire file.

<table>
<thead>
<tr>
<th>PDS members</th>
<th>For PDS and PDSE members, the <code>fastfilesize</code> attribute gets the file size from ISPF statistics if they exist; otherwise, the size of the entire PDS or PDSE data set is returned as the member size.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS or DA data sets</td>
<td>For PS or DA data sets, an approximate file size is calculated based on the device characteristics, the number of disk tracks in use, and the block size of the data set.</td>
</tr>
<tr>
<td>VSAM</td>
<td>For non-system-managed VSAM data sets, the estimated size using <code>fastfilesize</code> is the size of the data set.</td>
</tr>
</tbody>
</table>

The `fastfilesize` attribute speeds up data set access by calculating approximate file sizes during data set access. Use this only when you are displaying file names and sizes (using the `dir` Windows command, for example) because many commands (such as `copy` and editors) and many applications might not work correctly if `fastfilesize` is set. When modifying or copying a data set, the `nofastfilesize` attribute should be used to ensure accurate results.

`nofastfilesize:`
When you use the default, `nofastfilesize` attribute, the SMB server reads the entire file or member to get the file size. It stores the file size value in cache until release. Using this attribute might cause a delay when first accessing very large data sets.

Handling of the time stamps

z/OS UNIX file attributes define the following time stamps:

- `atime` The last time the file was accessed (read).
- `mtime` The last time the file was modified (write).
The SMB server handles time stamps differently for these types of data sets:
- System-managed PS data sets and system-managed VSAM data sets
- Direct Access data sets and non-system managed PS data sets
- Non-system managed VSAM data sets
- PDS and PDSE members.

**Time stamps for system-managed VSAM and PS data sets**
For system-managed PS data sets and system-managed VSAM data sets, \textit{atime} and \textit{mtime} are fully maintained, and the \textit{ctime} is set to the \textit{mtime}.

**Time stamps for non-system managed PS and DS data sets**
For non-system managed PS and DA data sets, consider the following:
- How time stamps are stored
- The requirements of your workstation programs
- The type of data set used to store the file.

**Storing time stamps:**
For non-system managed PS and DA data sets, the SMB server temporarily stores the time stamps in virtual storage, but not on DASD. These cached attributes are purged when the file is released or when the server is restarted. When the file is accessed again, the time stamps are regenerated.

**Client program requirements:**
Some workstation-based utilities (such as \texttt{make}) rely on date and time stamps. For example, \texttt{make} checks the update time of the object file with the source file and recompiles if the source has been updated. Before storing these types of files using the server, examine them before moving them to ensure that these attributes are unimportant. In an environment which relies on such utilities, use HFS.

**Generating time stamps:**
This is how the SMB server generates \textit{atime} and \textit{mtime} for non-system managed PS and DA data sets from the dates:

\[
\text{atime} = \text{mtime} = \text{reference\_date} + \text{time\_increment} \\
\text{ctime} = \text{creation\_date} + \text{time\_increment}
\]

\textit{time\_increment} is either the server local time or 23:59 hours. If \textit{reference\_date} or \textit{creation\_date} is equal to the server local date, the server local time is added. Otherwise, a fixed value of 23 hours and 59 minutes is added.

If \textit{reference\_date} = 0 (that is, the file has not yet been referenced), \textit{atime} and \textit{mtime} are set equal to \textit{ctime}.

**Time stamps for non-system managed VSAM data sets**
The time stamps for these types of data sets are set to the current time.

**Time stamps for PDSs and PDSEs**
A PDS data set can act as a directory. Members of the PDS are files within the directory. When the directory is accessed by the client, the z/OS UNIX times are expected for each file.

Ordinarily, z/OS does not maintain time stamps for members of a PDS. The z/OS UNIX time stamps here are generated from the z/OS creation and reference dates of the PDS data set containing the members. This is how the time stamps for PDS members are generated:
atime = mtime = reference_date + time_increment
ctime = creation_date + time_increment

ISPF is a z/OS base element that does maintain some additional statistics for each member. They include the creation date and the last modification date and time.

If the ISPF time stamps are present for a PDS member, this is how the server generates the time stamps and initializes the z/OS UNIX times:

atime = mtime = modification_date + modification_time
ctime = ISPF_creation_date + time_increment

time_increment is either server local time or 23:59 hours as described for non-system managed PS and DA data sets.

The server also creates new ISPF statistics for PDS members created by the clients. The ISPF statistics are created even if existing members do not have statistics.

The time stamp information is saved in the PDS directory according to ISPF conventions. If STATS=ON was specified when the member was created, the server uses them to get more accurate attributes. Even if STATS=ON was not specified originally, the server writes back new time stamp information if the member is modified from the workstation.

Time stamp generation for a PDSE member is identical to that of a PDS with one exception. Accurate mtime of a PDSE member is returned to a client as the result of a file attribute request for that PDSE member.

**Setting time stamps**

PC clients can issue commands that result in SETATTR requests (such as, Windows Explorer, right-click on file, and Choose Properties) to set the atime and mtime for a system-managed PS or VSAM data set. For PDSE members, setting mtime is allowed, but setting atime is not supported. PDSE member mtime is also maintained by PDSE access methods, so it is modified when a TSO user modifies the PDSE member.
Appendix F. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to z/OS TSO/E Primer, z/OS TSO/E User’s Guide, and z/OS ISPF User’s Guide Vol I for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

http://www.ibm.com/systems/z/os/zos/bkserv/
Notices

This information was developed for products and services offered in the USA.

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
USA

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
USA
Notices

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.

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

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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS™, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Trademarks

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

- AIX
- BookManager
- DFS
- DFSMSShsm
- IBM
- IBMLink
- Language Environment
- Library Reader
- MVS
- OS/390
- RACF
- Resource Link
- System z
- System z9
Notices

- WebSphere
- z/OS
- zSeries

These terms are trademarks or registered trademarks of Ricoh Co., Ltd., in the United States, other countries, or both:
- AFP
- Infoprint

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

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Special characters

_COMPUTER_NAME 152
_EUV_AUTLOG 143
_EUV_SVC_MSG_LOGGING 143
_IOE_DAEMONS_IN_AS 21, 144
_IOE_DFS_MODIFY_PATH 145
_IOE_DIRECTORY_CACHE_SIZE 135, 145
_IOE_DYNAMIC_EXPORT 14, 145
_IOE_EXPORT_TIMEOUT 44, 145
_IOE_HANG_DETECT_DUMP_LIMIT 146
_IOE_HANG_DETECT_OENODE_TIMEOUT 146
_IOE_HANG_DETECT_THREAD_TIMEOUT 146
_IOE_HFS_ATTRIBUTES 28
_IOE_HFS_ATTRIBUTES_FILE 147
_IOE_HFS_FILETAG 46, 147
_IOE_HFS_TRANSLATION 47, 98, 148
_IOE_INHERIT_TRANSLATION 46, 149
_IOE_MOVE_SHARED_FILESYSTEM 43, 148
_IOE_MVS_DFSDFLT 14, 35, 149
_IOE_PROTOCOL_RPC 149
_IOE_PROTOCOL_SMB 149
_IOE_RFS_ALLOC_TIMEOUT 150, 176
_IOE_RFS_ATTRIBUTES 28
_IOE_RFS_ATTRIBUTES_FILE 150
_IOE_RFS_STATUS_REFRESH_TIME 150
_IOE_RFS_TRANSLATION 52, 151
_IOE_RFS_WORKER_THREADS 120, 138, 151
_IOE_SMB_ABS_SYMLINK 37, 68, 151
_IOE_SMB_AUTH_DOMAIN_NAME 151
_IOE_SMB_AUTH_SERVER 152
_IOE_SMB_BACKUP_AUTH_SERVER 152
_IOE_SMB_BLKSIZE 153
_IOE_SMB_BROWSE_INTERVAL 32, 153
_IOE_SMB_CALLBACK_POOL 129, 153
_IOE_SMB_CLEAR 14
_IOE_SMB_CLEAR_PW 117, 154
_IOE_SMB_COMPUTER_NAME 31, 155
_IOE_SMB_CONNECT_MSGS 154
_IOE_SMB_CROSS_MOUNTS 43, 155, 169
_IOE_SMB_DESCRIPTION 155
_IOE_SMB_DIR_PERMS 155
_IOE_SMB_DOMAIN_NAME 31, 156
_IOE_SMB_FILE_PERMS 156
_IOE_SMB_IDLE_TIMEOUT 46, 156
_IOE_SMB_IDMAP 28, 33, 34, 104, 156
_IOE_SMB_MAIN_POOL 120, 129, 157
_IOE_SMB_MAXXMT 127, 157
_IOE_SMB_NT_SMBS 127, 157
_IOE_SMB_OCSF 157
_IOE_SMB_OLOCK_TIMEOUT 158
_IOE_SMB_OPLOCKS 127, 158
_IOE_SMB_PRIMARY_WINS 31, 158
_IOE_SMB_PROTOCOL_LEVEL 127, 158
_IOE_SMB_RAW 126, 159
_IOE_SMB_SCOPE 159
_IOE_SMB_SECONDARY_WINS 31, 160
_IOE_SMB_SETATTR_OVERRIDE 159
_IOE_SMB_TOKEN_FILE_MAX 130, 160
_IOE_SMB_TRANSPORTS 31
_IOE_SMB_WINS_PROXY 31, 160
_IOE_TKCGLUE_CACHE_SIZE 130, 160
_IOE_TKMGGLUE_SERVER_THREADS 130, 161
_IOE_TRACE_TABLE_SIZE 161
_IOE_USE_PTHREAD_SECURITY 161
dynamically recognize changes 33
_IOE_VM_CACHE_SIZE 120, 133, 162
_IOE_VM_MAX_FILES 133, 162
_IOE_VNODE_CACHE_SIZE 135, 162
_IOE_WIRE_CODEPAGE 47, 52, 60, 162
# (pound sign) xiv

A

Access Control List (ACL) 12, 37, 48
FSSEC 14
accessibility 189
accessing
data 65
HFS 67
RFS 68
files
PC 37
PDS 182
PDSE 182
print drives
PC clients 74
printers 71
shared directories
Linux 66
Windows Vista 66
Windows XP 65
shared printers
Linux 71
Windows XP 72
ACL (Access Control List) 12, 37, 48
FSSEC 14
ADAPTERS report 121
adding
printers
Windows Vista 72
Windows XP 72
administrators
defining 15
AFS4INT report 121
ALL report 121
attributes
fastfilesize 186
nofastfilesize 186
UNIX 186
audience
(PC) users 1
system administrators 1
authorization 12
HFS 48
authorization (continued)
  print data 60
  RFS 52
  sharing files 48
automount 14, 44
  accessing share 164
  example 46
home directories
  PC user access 46

B
backslash xiv
BPX.DAEMON
  BPX.SERVER 7
bufhigh 120, 138

C
caching
  _IOE_VM_CACHE_SIZE 120
  bufhigh 120
  PC clients 178
callable services 47
case sensitivity
  considerations 67
directory name 67
  file name 67
changing
  environment variables 28
  hfsattr 28
  Infoprint Server DLL 29
  mappings 28
owner
  HFS file 165
rfstab 28
shared
  directories 28
  printers 28
command structure 4
commands (continued)
  z/OS system 77
configuration 11
configuration file
  considerations 9
  sample 165
  updating 8
considerations
  case sensitivity 67
configuration file
  exported data 8
logon 53
migration 7
networking 31
printers 8
RACF database 8
record data 177
RFS
  directory and file name 68
  SMB file and print server 9
  symbolic links 7
  tuning 120
  using both SMB and DCE DFS 169
conventions
  this documentation xiii
creating 4
  configuration files 15
data sets 174
files
  direct access 180
  physical sequential 180
  VSAM 183
  z/OS 178
PDS 181
PDSE 181
shared directories 4, 41
  steps 41, 50
shared printers 59
smbidmap file 33
UID(0) 17
crossing
  local mount points 169
customizable files 171

D
daemon configuration file 22, 26
examples 26
data
  accessing 65
  RFS 68
  sharing 175
data normalization 126
data set
  attributes 84
data sets
  creating 174
  mapping
    RFS 173
PDS 49
PDSE 49
data sets (continued)
  reading 174
  using 173
  VSAM 49
  writing 174
DCE_START_SOCKET_NAME 144
defining
  SMB administrators 15
definitions
  root file system 37
deleting
  identity mapping entries 34
determining
  file size 178
  SMB user ID 35
devtab 40, 91
  examples 93
df command 41
DFS
  server daemons
    dfscntl 27
    starting 26
    viewing 25
DFS control 4
DFS server
  stopping 8
DFS server address
  stopping 24
DFS server address space 21
DFS server daemon
  viewing status 25
DFS server daemons
  starting 23, 24
  steps 24
  stopping 23, 24
DFS server tuning 126
dfs_cpfiles 15
  examples 16
  using 15
dfs_cpfiles program 7
dfscntl 22, 26
  file permissions 19
  starting DFS server daemons 27
  using -nodfs option 28
dfsexport 110
  examples 112
dfs kern
  examples 79
  file permissions 19
dfsshare 6, 113
dfssyntax 116
dfstab 40, 95
  examples 96
DHCP (dynamic host configuration protocol) 61
diagnostics (continued)
  SMBTKC 131
  STORAGE 141
  TKM 131
  VM 133
direct access
  files
    creating 180
directories
  home 13, 27, 97
  shared
    changing 28
    creating 4
directory name
  case sensitivity 67
disability 189
disk full
  error 164
displaying
  printer queue
    steps 73
Distributed File Service
  installation 7
  SMB
    commands 109
    files 83
DNS
  steps
    Windows XP 61
domain controller
    Windows 32
domain name service (DNS) 31
dynamic export 14, 40, 44, 92
  HFS 42
dynamic host configuration protocol (DHCP) 61
E
encrypted passwords 11, 170
  RACF DCE segments for SMB 57
  steps 57
end of line characters
  text files 165
envar 97
  examples 97
environment variables 143
  _EUV_AUTOLOG 143
  _EUV_SVC_MSG_LOGGING 143
  _IOE_DAEMONS_IN_AS 21, 144
  _IOE_DFS_MODIFY_PATH 145
  _IOE_DIRECTORY_CACHE_SIZE 135, 145
  _IOE_DYNAMIC_EXPORT 14, 145
  _IOE_EXPORT_TIMEOUT 44, 145
  _IOE_HANG_DETECT_DUMP_LIMIT 146
  _IOE_HANG_DETECT_INTERVAL 146
  _IOE_HANG_DETECT_OENODE_TIMEOUT 146
  _IOE_HANG_DETECT_THREAD_TIMEOUT 146
  _IOE_HFS_ATTRIBUTES_FILE 28, 147
  _IOE_HFS_FILETAG 46, 147
  _IOE_HFS_TRANSLATION 47, 98, 148
  _IOE_INHERIT_TRANSLATION 46, 149
examples

automount 46
creating a PDS or PDSE 181
creating VSAM files 184
daemon configuration file 26
devtab 93
dfs_cpf 16
dfsexport 112
diskern process 79
dsshare 6
dfstab 96
eenv 97
fsattr 98
ioepdcf 101
Linux

creating a PDS 181
creating a PDSE 181
creating physical sequential files 180
creating VSAM files 183
direct access 180
mount 66
specifying attributes 184
rfstab 89
smbidmap 105
smbpw 118
smbtab 107
start dfs 81
stop dfs 82
exploiting
SAM striped files 185
export process 22
exported data
considerations 8
exporting
file systems
HFS 37
RFS 49
extending
PDS 182
PDSE 182

F

gensize attribute 186
features
Server Message Block (SMB) 3
file
hierarchy
HFS 37
permissions 21
size
determining 178
generating 186
smbidmap 33
creating 33
systems 37
exporting 111
root 37
file data translation
HFS 46
RFS 52

environment variables (continued)

_IOE_MOVE_SHARED_FILESYSTEM 43, 148
_IOE_MVS_DFSDFLT 14, 35, 149
_IOE_PROTOCOL_RPC 149
_IOE_PROTOCOL_SMB 149
_IOE_RFS_ALLOC_TIMEOUT 150, 176
_IOE_RFS_ATTRIBUTES_FILE 28, 150
_IOE_RFS_STATUS_REFRESH_TIME 150
_IOE_RFS_TRANSLATION 52, 151
_IOE_RFS_WORKER_THREADS 120, 138, 151
_IOE_SMB_ABS_SYMLINK 37, 68, 151
_IOE_SMB_AUTH_DOMAIN 151
_IOE_SMB_AUTH_SERVER 152
_COMPUTER_NAME 152
_IOE_SMB_BACKUP_AUTH_SERVER 152
_COMPUTER_NAME 152
_IOE_SMB_BLKSIZE 153
_IOE_SMB_BROWSE_INTERVAL 32, 153
_IOE_SMB_CALLBACK_POOL 129, 153
_IOE_SMB_CLEAR_PW 14, 117, 154
_IOE_SMB_COMPUTER_NAME 31, 155
_IOE_SMB_CONNECT_MSGS 154
_IOE_SMB_CROSS_MOUNTS 43, 155, 169
_IOE_SMB_DESCRIPTION 155
_IOE_SMB_DIR_PERMS 155
_IOE_SMB_DOMAIN_NAME 31, 156
_IOE_SMB_FILE_PERMS 156
_IOE_SMB_IDLE_TIMEOUT 46, 156
_IOE_SMB_IDMAP 28, 33, 34, 104, 156
_IOE_SMB_MAIN_POOL 120, 129, 157
_IOE_SMB_MAXXMT 127, 157
_IOE_SMB_NT_SMBS 127, 157
_IOE_SMB_OCSF 157
_IOE_SMB_OPLCK_TIMEOUT 158
_IOE_SMB_OPLCKS 127, 157
_IOE_SMB_PRIMARY_WINS 31, 158
_IOE_SMB_PROTOCOL_LEVEL 127, 158
_IOE_SMB_RAW 126, 159
_IOE_SMB_SCOPE 159
_IOE_SMB_SECONDARY_WINS 31, 160
_IOE_SMB_SETATTR_OVERRIDE 159
_IOE_SMB_TOKEN_FILE_MAX 130, 160
_IOE_SMB_TRANSPORTS 31
_IOE_SMB_WINS_PROXY 31, 160
_IOE_TKCGLUE_CACHE_SIZE 130, 160
_IOE_TKMG_GLUE_TOKENS 130, 161
_IOE_TKMG_GLUE_SERVER_THREADS 130, 161
_IOE_TRACE_TABLE_SIZE 161
_IOE_USE_PTHREAD_SECURITY 33, 161
_IOE_VM_CACHE_SIZE 133, 162
_IOE_VM_MAX_FILES 133, 162
_IOE_VNODE_CACHE_SIZE 135, 162
_IOE_WIRE_CODEPAGE 47, 52, 60, 162
changing 28
DCE_START_SOCKET_NAME 144
LIBPATH 144
NLSPATH 144
TZ 144
error
space 164
K
keyboard 189

L
LAN manager
settings 56
LFS report 121
LIBPATH 144
Linux
accessing
shared directories 66
clients
connect to SMB 61
example
accessing PDS or PDSE 182
copy 181
creating a DA file 180
creating a PDS 181
creating a PDSE 181
creating physical sequential files 180
creating VSAM files 183
mkdir 181
mkdir release 177
mount 66
removing a PDS or PDSE 182
printing 71
shared directories 65
steps
accessing shared printers 71
LMHOSTS 32
LMHOSTS file
steps
finding SMB 62
local file system 121
local mount points
crossing 169
locate
SMB 63
DNS 61
LMHOSTS 62
Windows XP 62
SMB server
Windows Vista 63
Windows XP 63
locating
SMB server 61
LOCKING report 121
logon
considerations 53

M
mainframe
education xv
managing
SMB processes 21
mapping
data sets
RFS 173
mapping (continued)
PC view 173
record data 173
shared directories 65, 66
Universal Naming Convention 4
user IDs
SMB to z/OS 33
mappings
changing 28
metacharacter 118
migration
considerations 7
new SMB release 7
modify
to stop DFS server daemon 25
modify command 24
modify dfs 23, 24
modify dfs processes 78
modifying
identity mapping entries 34
mount points
multiple 164
moving
PDS member 183
PDSE member 183
moving files 68
MS-DOS move command 68
multiple mount points
crossing 164
multipliers 84
my network places
using 63
My Network Places
not in SMB server 166

N
naming
files
z/OS 179
net use command 35
NETBIOS
ports 31
Network
domain controller 32
Network and Sharing Center
using 63
Network security
LAN manager 56
networking
considerations 31
NLSPATH 144
nofastfilesize attribute 186
non-system managed
time stamps 187
nonzero UID
changing 19
creating 18
Notices 191
removing
- PDS 182
- PDSE 182
  - shared directories 42, 51
  - shared printers 59
renaming
- PDS member 183
- PDSE member 183
reports
  - SMB server
    - ADAPTERS 121
    - AFS4INT 121
    - ALL 121
    - LFS 121
    - LOCKING 121
    - RFS 121
    - SMBCOMM 122
    - SMBIN 122
    - SMBPRT 122
    - SMBTKC 123
    - STORAGE 123
    - TKM 123
    - VM 123
- RESET command 123
restriction
- zFS sysplex 7
restrictions
- RFS 69
  - using alias names
    - z/OS files 179
RFS
  - accessing
    - data 68
    - authorization 52
  - creating
    - shared directory 50
  - directory and file name considerations 68
  - file data translation 52
  - file names
    - refreshing 177
  - file systems
    - exporting 49
files
  - sharing 48
  - free space 53
hierarchy 49
mapping
  - data sets 173
removing
  - shared directory 51
  - restrictions 69
- smbtab, dfstab, devstab 50
RFS (record file system) 48
RFS report 121
rfstab 28, 84, 103
  - changing 28
  - examples 89
root file system 37

S
saving
- files to HFS 165
search
- find SMB
  - Windows Vista 63
Server Message Block (SMB) xiii, 3
  - clients 3
  - features 3
  - overview 3
  - PC connection 3
  - See SMB (Server Message Block) 126
server process 4
shared directories 4
setting
  - _IOE_SMB_IDMAP 34
  - time stamps 188
share
  - automount 164
shared
  - directories changing 28
  - printers changing 28
shared directories
  - accessing
    - Linux 66
    - Windows Vista 66
    - Windows XP 65
  - creating 41
  - mapping 65, 66
  - removing 42, 51
  - using 65
shared printers 4
  - creating 4, 59
  - removing 59
sharing
  - data 175
  - files 37
    - HFS 37
    - RFS 48
printers 59
shell metacharacter 118
shortcut keys 189
shortcuts
  - commands 5
  - single sign-on 170
SMB
  - administrators defining 15
  - commands 109
encrypted passwords
  - RACF DCE segments 57
environment variables 143
file and print server 11
  - considerations 9
  - functions
    - file and print server 11
  - installation 11
  - managing
    - processes 21
SMB (continued)
migration
   new release 7
server
   finding 63
   locating 61
   not in My Network Places 166
shared directories
   using 65
user IDs
   determining 35
   mapping to z/OS 33
SMB (Server Message Block) xiii, 3, 126
SMB files 83
SMBCOMM report 122
smbidmap 33, 104
   examples 105
SMBINT report 122, 127
SMBPRT report 122, 128
smbpw 117
   examples 118
smbtab 40, 106
   accessing share 164
   examples 107
SMBTKC report 123, 132
space
   error 164
start command 23
start dfs 81
   examples 81
starting
   DFS server daemons 23, 24, 26
   dfscntl 27
   steps 24
steps
   configuration
      file and print server 11
   creating shared directories 41, 50
dfs_cpfiles 15
displaying
   printer queue 73
DNS
   Windows XP 61
   encrypted passwords 57
   export file systems 111
Linux
   accessing shared printers 71
LMHOSTS file
   locating 62
mapping
   shared printer to logical printer 71
shared printers 59
sharing printers 59
starting DFS server daemons 24
translating file data 47
Windows 2003
   update registry 163
Windows Vista
   adding printers 72
Windows XP
   accessing shared printers 72
steps (continued)
Windows XP (continued)
   adding printers 72
   update registry 163
WINS
   Windows XP 62
stop command 24
stop dfs 82
   examples 82
stopping
   DFS server 8
   DFS server address 24
   DFS server daemons 23, 24
storage
   file size value 185
   STORAGE report 123
storing
   time stamps 187
   symbolic links
      considerations 7
      HFS 67
syntax
   command for checking 116
system administrators 1
system-managed
   time stamps 187

T
TCP/IP
   calling ports 31
text files
   end of line characters 165
   threading 120
   tuning options 120
time stamps 178
   generating 187
   handling 186
   non-system managed 187
   PDS 187
   PDSE 187
   setting 188
   storing 187
   system-managed 187
TKM report 123
TRACE command 124
tuning
   considerations 120
   DFS server 126
   system-managed 187
tuning (continued)
guidelines 119
threading 120
tuning options 126
_IOE_VM_CACHE_SIZE 120
bufhigh 120
types
printers 74
users 15
TZ 144

U
UID(0)
changing 19
creating 17
UNC (Universal Naming Convention) 4
Universal Naming Convention (UNC) mapping 4
UNIX system services
df command 41
file attributes 186
update registry
Windows 2003 steps 163
Windows XP steps 163
updating
configuration file 8
PDS member 182
PDSE member 182
user credentials
changing behavior 33
user ID
a new SMB installation 17, 18
how to determine the z/OS user ID 34
modifying an existing SMB installation 19
z/OS 34
users
PC 1
types 15
using
-nodfs option 28
data sets 173
dfs_cpfies 15
modify dfs 24
to stop DFS server daemon 25
my network places 63
Network and Sharing Center 63
passthrough authentication 55
SMB
shared directories 65
SMB Server
Windows XP 61
this documentation xiii

V
variables, environment (continued)
_IOE_SMB_CALLBACK_POOL 129
_IOE_SMB_MAIN_POOL 120, 129
_IOE_SMB_MAXXMT 127
_IOE_SMB_NT_SMBS 127
_IOE_SMB_OPLOCKS 127
_IOE_SMB_PROTOCOL_LEVEL 127
_IOE_SMB_RAW 126
_IOE_SMB_TOKEN_FILE_MAX 130
_IOE_TKCLU_E_CACHE_SIZE 130
_IOE_TKM_MAX_TOKENS 130
_IOE_TKMGLUE_SERVER_THREADS 130
_IOE_VM_CACHE_SIZE 133
_IOE_VM_MAX_FILES 133
_IOE_VNODE_CACHE_SIZE 135
viewing
DFS server daemons 25
status
DFS server daemon 25
virtual storage access method (VSAM) 49
VM report 123
VSAM
data sets
time stamps for non-system managed 187
files
creating 183
VSAM (virtual storage access method) 49

W
Windows
connect to SMB 61
windows internet naming service (WINS) 31
Windows Vista
accessing
shared directories 66
authentication levels 56
mapping
shared directories to logical drives 66
Network and Sharing Center 63
Network Sharing Center 61
steps
adding printers 72
Windows XP
accessing
shared directories 65
authentication levels 56
mapping
shared directories to logical drives 65
steps
accessing shared printers 72
adding printers 72
UNC
mapping 66
using
SMB Server 61
WINS
steps
Windows XP 62
WINS (Windows Internet Naming Service) 31
workloads 126
writing
data sets 174

Z

z/OS
files
  creating 178
  naming 179
user IDs
  mapping 33
z/OS Basic Skills information center xv
zero space
correct 164
zFS
  sysplex-aware
  restriction 7
zSeries File System (zFS)
support 3
Readers’ Comments — We’d Like to Hear from You

z/OS
Distributed File Service
SMB Administration

Publication No. SC24-5918-09

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.
Submit your comments using one of these channels:
• Send your comments to the address on the reverse side of this form.
• Send your comments via e-mail to: mhvrdfs@us.ibm.com

If you would like a response from IBM, please fill in the following information:

Name..................................................................................................................

Address...........................................................................................................

Company or Organization..............................................................................

Phone No.........................................................................................................

E-mail address................................................................................................