z/VM
7.3

DFSMS/VM Customization

IBM

Note:

Before you use this information and the product it supports, read the information in "Notices" on page 93.

Contents

Figures

Tables

x

About This Document

This document describes the customization concepts for Data Facility Storage Management Subsystem/ Virtual Machine (DFSMS/VM), an IBM® z/VM® feature that improves productivity by providing storage management for Shared File System (SFS) storage and Removable Media Services (RMS), and by easing the task of moving minidisks from one physical device to another. The topics covered include verifying the installation, authorizing and customizing DFSMS/VM operations, and improving DFSMS/VM performance.

Intended Audience

This document is designed for system programmers responsible for customizing DFSMS/VM.

The programmer should be familiar with the *Program Directory for function level 221* and the *z/VM: DFSMS/VM Planning Guide* before customizing DFSMS/VM.

Where to Find More Information

See "Bibliography" on page 97 at the back of this document.

Links to Other Documents and Websites

The PDF version of this document contains links to other documents and websites. A link from this document to another document works only when both documents are in the same directory or database, and a link to a website works only if you have access to the Internet. A document link is to a specific edition. If a new edition of a linked document has been published since the publication of this document, the linked document might not be the latest edition.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. See How to send feedback to IBM for additional information.

Summary of Changes for z/VM: DFSMS/VM Customization

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line (|) to the left of the change.

SC24-6274-73, z/VM 7.3 (December 2023)

This edition includes terminology, maintenance, and editorial changes.

SC24-6274-73, z/VM 7.3 (September 2022)

This edition supports the general availability of z/VM 7.3. Note that the publication number suffix (-73) indicates the z/VM release to which this edition applies.

SC24-6274-01, z/VM 7.2 (September 2020)

This edition supports the general availability of z/VM 7.2.

SC24-6274-00, z/VM 7.1 (September 2018)

This edition supports the general availability of z/VM 7.1.

Chapter 1. Introduction

This chapter is a brief overview of Data Facility Storage Management Subsystem for z/VM (DFSMS/VM) and summarizes the customization options.

DFSMS/VM Overview

DFSMS/VM is a feature of z/VM designed to (1) manage files in Shared File System (SFS) file pools, (2) help you move CMS minidisks between like and unlike direct access storage devices (DASD) quickly and efficiently, and (3) provide command and Callable Services Library (CSL) interfaces to removable media library support.

Figure 1 on page 1 shows the four major functional components of DFSMS/VM:

```
                    ┌─────────────────┐
                    │                 │
                    │  User Interface │
                    │                 │
                    └─────────────────┘
          ┌───────────────┼───────────────┐
┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐
│                 │ │                 │ │    Removable    │
│    Minidisk     │ │      Space      │ │      Media      │
│   Management    │ │   Management    │ │    Services     │
│                 │ │                 │ │                 │
└─────────────────┘ └─────────────────┘ └─────────────────┘
```
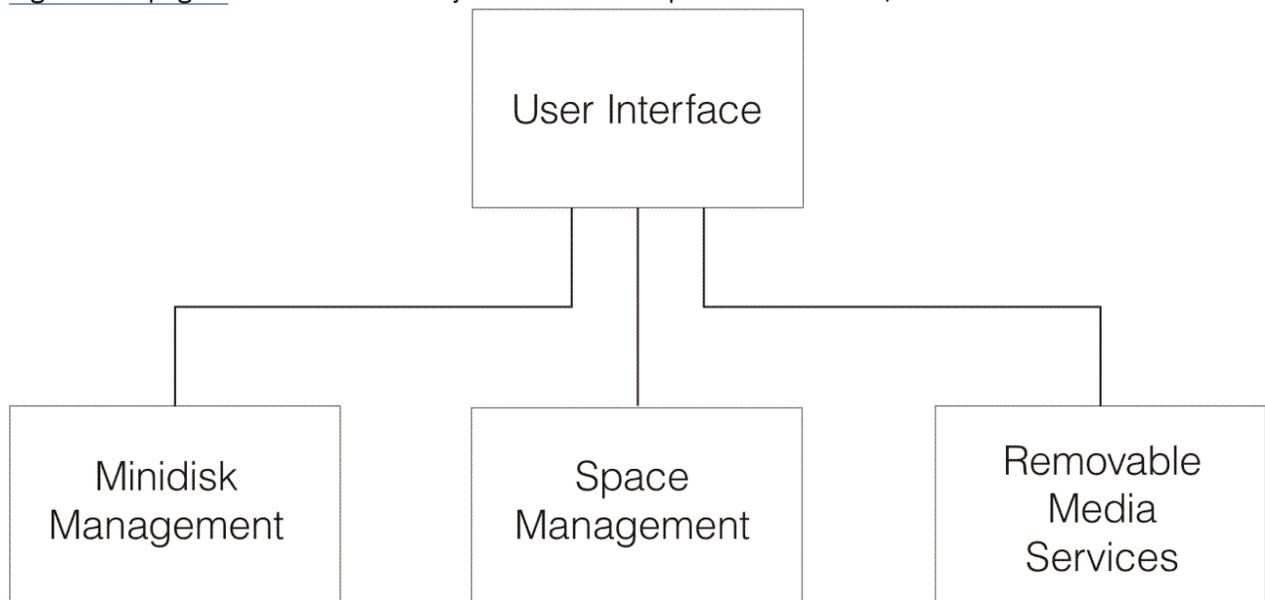
Figure 1. DFSMS/VM Major Components

- User interface consists of:
  - Interactive Storage Management Facility (ISMF), which offers a set of menus, data entry panels, and list panels that run on the user's virtual machine. Through ISMF, you can select applications to:
    - Perform storage management functions against the file and minidisk lists
    - Create file, minidisk, and management class lists based on your criteria
    - Define management classes
    - Edit and test automatic class selection (ACS) routines
    - Define the installation's space management policies in DFSMS configurations
    - Activate DFSMS/VM configurations.
  - DFSMS module, which provides a command interface to most minidisk and storage management functions
  - DFSMSRM module, which provides a command interface to removable media services functions
  - A set of CSL-routine programming interfaces to removable media services functions.
- Minidisk management provides:
  - A high-performance data mover for copying CMS-formatted minidisks between like and unlike DASD
  - The capability to do an integrity-evaluation of CMS minidisks

- The capability to perform the move and integrity-evaluation operations against a list of minidisks.
- Space management provides:
  - Automatic assignment of management classes to SFS files and directories
  - Automatic migration (with compaction) or erasure of selected SFS files based on management class policies
  - Migration to ML1 and ML2 (lower cost tape and/or dasd)
  - Automatic recall of migrated files upon reference
  - Command-initiated migration and recall of files.
- Removable media services (RMS) provides:
  - Support of the IBM 3495 Tape Library Dataserver to extend automated mount support and other library capabilities to z/VM users and guests, as well as tape management applications.

Installation Verification Overview

Verifying the installation of DFSMS/VM includes:

- Verifying user machine status
- Verifying DFSMS
- Verifying removable media services functions and IBM 3495 Tape Library Dataserver support
- Minidisk management and space management verification.

Refer to Chapter 2, "Verifying the Installation Process," on page 5 for details on verifying the installation of DFSMS/VM.

Postinstallation and Optional Customization Overview

Once you have installed DFSMS/VM and verified the installation of DFSMS/VM, there are some postinstallation procedures and customization options to consider. Postinstallation and optional customization of DFSMS/VM includes:

- Performing optional customization
- Defining ML1 file space
- Verifying optional customization
- Verifying configuration information
- Verifying log files
- Verifying segment access
- Enabling SFS file pools for storage management by DFSMS/VM
- Defining an ML2 server.

Refer to Chapter 3, "Postinstallation and Optional Customization," on page 19 for postinstallation procedures and customization options.

Authorization Overview

Some functions of DFSMS/VM require proper authorization. This can be granted to a virtual machine by using one of the following methods:

- Resource Access Control Facility (RACF/VM)
- Another facility for security
- An authorization file.

Refer to Chapter 4, "Authorizing DFSMS/VM Users," on page 33 for details on authorizing DFSMS/VM users.

Customization Overview

DFSMS/VM includes three powerful customization capabilities.

DFSMS/VM Control File

This file contains keywords with values you can modify to control:

- DFSMS/VM virtual machine names
- Minidisk management operations
- Space management operations
- Message selection and disposition
- RMS resource names and operational options.

Installation-Wide Exits

You may want to perform various processes associated with DFSMS/VM functions or operations. You can customize a variety of DFSMS/VM functions with installation-wide exits, or you can use the default exits as shipped with the product.

Automatic Class Selection Processing

ACS processing automatically assigns management classes to SFS files and directories. You can customize this processing to meet installation storage management goals. ACS processing can be customized by writing an ACS routine or by writing an installation exit in either one of the two exits (REXX or assembler language) associated with ACS. Refer to *z/VM: DFSMS/VM Storage Administration* for more details on ACS processing and customization.

Refer to chapters 7 through 10 for detailed customization information.

Backout Overview

Also, after you have installed DFSMS/VM, you can remove it. The process of removing DFSMS/VM is called *backout*. Removing DFSMS/VM requires the following:

- Recalling **all** migrated files (if you used the space management function of DFSMS/VM)
- Identifying primary file pools managed by DFSMS/VM (if you used the space management function of DFSMS/VM)
- Performing miscellaneous cleanup.

Refer to Chapter 9, "Removing DFSMS/VM," on page 75 for details on removing DFSMS/VM.

Chapter 2. Verifying the Installation Process

Before attempting to verify your installation, ensure you have installed DFSMS/VM according to the steps outlined in the *Program Directory for function level 221*.

Verify the DFSMS/VM installation by doing the examples described in this section. These examples are designed to invoke prerequisite software, machine interfaces (IUCV/APPC), and the various product elements installed on the product disk.

Perform the functions in this section prior to performing any optional customization. During verification, if you receive any error messages, refer to the *z/VM: DFSMS/VM Diagnosis Guide* and *z/VM: DFSMS/VM Messages and Codes* manuals for problem resolution and error messages.

Important

It is recommended that verification be performed from the user ID DFSMS. However, any authorized user ID can perform verification. If you use a user ID other than DFSMS, make sure you have authorized this user. The user ID must have access to the product code (DFSMS 1B5). For information regarding user authorization, refer to Chapter 4 in the *Program Directory for function level 221*.

Note: For more information about using DFSMS/VM functions, see *z/VM: DFSMS/VM Storage Administration*.

Step 1: Verify Code in the User's Machine

Before using DFSMS/VM, you should verify several characteristics of the CMS virtual machine performing the verification. Spool the console to allow you to check your results.

1. If you are not already logged on to DFSMS or the virtual machine where verification is to be done, do so now and then do the following steps:

    a. Link to MAINT 5E5 (VMSES/E production disk) and access it as file mode B:

    ```
    CP LINK MAINT 5E5 5E5 RR
    ACC 5E5 B
    ```

    b. Access the VMSES/E software inventory disk as file mode D:

    ```
    ACC 51D D
    ```

2. Verify that the virtual machine is linked to the product code (the VMSES/E BUILD disk).

    To display your current links, issue the following command:

    ```
    VMFQMDA override    compname
    ```

    where,

    override
    is the value you specified when you installed DFSMS/VM. Refer to *Program Directory for function level 221* for more information.

    compname
    is the value you specified when you installed DFSMS/VM. Refer to *DFSMS/VM Program Directory for function level 221* for more information.

To test the DFSMS/VM commands, do the following:

1. Issue a HELP DFSMS command. The DFSMS/VM Storage Management Help menu is displayed.

2. Select help for the DFSMS CHECK command, and note the syntax (or refer to the *z/VM: DFSMS/VM Storage Administration*).
3. If you installed minidisk management, issue a CHECK command with the FILE option against any linked CMS minidisk.

The information is returned both to the screen and to a file on your A-mode SFS directory or minidisk. Compare the results; they should be similar. There should be no errors in the report. If there are errors, they need to be resolved before continuing with the install process. Refer to the following report:

```
Userid:  DFSMS   Date:  03/18/92   Time:  13:19:34
CHECK   191  ( FILE
Address:  0191  Device Type:  3380  Date Created:  90/09/27 08:11:13
Volid:  VMT191  Block size:  4096  Last Changed:  92/03/11 08:21:48
Cyls:    4  Usable Cyls:    4


Total number of CMS blocks:     600
Number of CMS blocks used:       23    (  4%)
CMS blocks counted:              23
Blocks in allocation map:        23

Lost CMS blocks:                  0
Invalid CMS blocks:               0
Overlapping CMS blocks:           0

Disk origin pointer:              4
Files reported in directory:     13 (Including DIRECTOR and ALLOCMAP)
Number of files found:           13

Command CHECK  gave return code = 0
```

Step 2: Verify DFSMS/VM

If you installed RMS only, go to .

This section will help you verify space and minidisk management functions. The purpose is to check that the machines come up and run correctly. Before starting the DFSMS/VM verification procedures, do the following:

1. If SMSMASTR is not running, log on to it and watch the master initialize. If you do not have RACF/VM or its equivalent installed, you will receive the following error messages:

```
RPICMS009W IUCV communication has been severed.
RPICMS011E You are not authorized to connect to RACF.
***
*** Error initiating connection to RACF/VM, return code = 8
*** DFSMS/VM authority checking will be done via the authorization file
***
```

Ignore these messages since you have not yet set up RACF/VM authorization. These messages indicate that you will be using the DFSMS/VM authorization file instead of RACF/VM.

Note: If you are installing DFSMS/VM Function Level 221 for the first time, message 8061W appears: **Active configuration file not found**.

If you are logging on to DFSMS/VM Function Level 221 and you have been running DFSMS/VM Function Level 220 previously, the following message (Message 8063W) appears: **Configuration file is not at the current version of DFSMS/VM**.

Either message is expected since you have not yet activated a DFSMS/VM Function Level 221 configuration.

If previously you have been running DFSMS/VM Function Level 220 and already have DFSMS/VM-managed file pools, activate a DFSMS/VM Function Level 221 configuration as soon as possible, because any DFSMS/VM-managed filepools will not have ACS processing and new and recalled files are not assigned management classes until you do.

During master initialization, any errors that may prevent the DFSMS master machine from initializing are written to the console. If you have received any errors, resolve the errors before continuing. Refer to *z/VM: DFSMS/VM Diagnosis Guide* and *z/VM: DFSMS/VM Messages and Codes* for problem resolution and error messages.

2. If you have not given the SMSMASTR virtual machine the proper authority to autolog servers, ensure that the DFSMS server and minidisk server virtual machines are logged on and running. Resolve any errors that may occur during server initialization.

3. If you are not already logged on to the installer virtual machine (DFSMS), log on to it now and verify that it is linked to the product code (the VMSES/E BUILD disk).

4. From the DFSMS virtual machine, issue the following command from the ready prompt:

```
DFSMS QUERY STATUS
```

Several messages will be displayed. The first is FSMDPS3161, which indicates that DFSMS is running. The next message, FSMDPS3162, indicates that there are no outstanding requests in the system. For each server logged on, you will see a message displayed. Message FSMDPS3200I is displayed for space management servers. Message FSMDPS3114I is displayed for minidisk management servers. These messages contain the server name and the number of tasks specified in the control file allowed for that server. The following is an example of returned messages:

```
FSMDPS3161I  DFSMS master is running FSMDPS3162I  DFSMS has 0 outstanding requests
FSMDPS3200I  DFSMS server SMSSRV01 has 2 tasks available for work and 0 tasks busy
FSMDPS3200I  DFSMS server SMSSRV02 has 6 tasks available for work and 0 tasks busy
FSMDPS3200I  DFSMS server SMSSRV03 has 8 tasks available for work and 0 tasks busy.
Ready; T=0.01/0.02 15:03:39
```

Note: If you have already customized your system, then additional messages regarding ML1 and ML2 may be displayed.

Step 3: Verify Minidisk Management

Before starting the minidisk management verification procedures, do the following:

1. This verification process gives you the choice of either creating a new minidisk or using an existing minidisk for the procedure.

   a. Because a minidisk CHECK request links to a minidisk in Exclusive Write (EW) mode, ensure no other virtual machine has a WRITE link to this minidisk. You might want to release and detach the DFSMS 1A4 disk and issue a check against it.

   b. If you create a new minidisk for this procedure, copy a few files to that minidisk to provide content to the output created in subsequent steps.

2. Ensure you can access the product code by checking that you are linked to the product disk (DFSMS 1B5).

3. You can issue the minidisk CHECK request from an ISMF storage administrator's Minidisk List panel on a virtual machine authorized for DFSMS/VM functions.

   a. Log on to the authorized virtual machine (DFSMS).

   b. Type ISMF or DFSMS on the CMS ready line. The user ISMF Primary Option menu is displayed:

```
DGTSMMD1                    ISMF PRIMARY OPTION MENU
 ENTER SELECTION OR COMMAND ===>

   0  ISMF PROFILE    - Change ISMF user profile
                       - Control how ISMF logs and traces errors

   1  FILE            - Access and view attributes of SFS files and
                         directories
                       - Browse, edit, migrate, or recall SFS files
                       - Create and modify lists of SFS files and
                         directories

   3  MANAGEMENT CLASS - Display the attributes of a management class
                       - Create and modify lists of management classes

+----------------------------------------------+ rt, and erase
| 5684-112 (C) Copyright IBM Corp. 1989, 1991. |
| All Rights Reserved.                         | og and list defaults
| US Government Users Restricted Rights -      |
| Use, duplication or disclosure restricted by |
| GSA ADP Schedule Contract with IBM Corp.     |
| Licensed Materials - Property of IBM.        |
+----------------------------------------------+
 USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 2. ISMF Primary Option Menu for End User (DGTSMMD1) with POP-UP Window

Pop-up windows appear on the ISMF Primary Option Menu and each of the list panels. The first time you access any of these panels, the pop-up window is displayed.

The pop-up on the Primary Option Menu contains the copyright statement. The pop-up on the list panels contains task oriented information that will help users determine what tasks they can perform. Press Enter to clear the pop-up window from the panel.

If you find that PF key definitions have replaced the directional line provided by the ISMF interface, then type PFSHOW OFF to erase the PFkeys and view the directional line. At any time, you can type PFSHOW ON to view the PF keys and PFSHOW OFF to view the directional line.

Though a pop-up window is shown in , we will not show the pop-up window on subsequent ISMF panels so that all of the information on the panel can be displayed.

shows the user ISMF Primary Option menu without the pop-up window.

```
DGTSMMD1                    ISMF PRIMARY OPTION MENU
 ENTER SELECTION OR COMMAND ===> 0

   0  ISMF PROFILE    - Change ISMF user profile
                       - Control how ISMF logs and traces errors

   1  FILE            - Access and view attributes of SFS files and
                         directories
                       - Browse, edit, migrate, or recall SFS files
                       - Create and modify lists of SFS files and
                         directories

   3  MANAGEMENT CLASS - Display the attributes of a management class
                       - Create and modify lists of management classes

   L  LIST            - Access, print, filter, sort, and erase
                         previously saved lists

   X  EXIT            - End ISMF using console, log and list defaults






 USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 3. ISMF Primary Option Menu for End User (DGTSMMD1)

c. Type **PANELID** on the command line of the ISMF Primary Option menu to display the panel ID of the ISMF panels and press Enter. The panel ID for the user ISMF Primary Option Menu is DGTSMMD1. The panel ID for the storage administrator ISMF Primary Option Menu is DGTSMMD2.

Bridge to rest of list.

In order to issue the minidisk CHECK request that will verify minidisk management, continue with the following steps:

4. Type **0** on the command line of the ISMF Primary Option menu and press Enter. The ISMF Profile Option menu (DGTSPPF1) is displayed:

```
DGTSPPF1                    ISMF PROFILE OPTION MENU
ENTER SELECTION OR COMMAND ===> 0


SELECT ONE OF THE FOLLOWING:

  0  USER MODE          - Change ISMF user mode

  1  LOGGING CONTROL    - Control how ISMF logs and traces errors

  X  EXIT               - Return to ISMF Primary Option Menu







USE ENTER TO SELECT OPTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 4. ISMF Profile Option Menu (DGTSPPF1)

5. Type **0** on the command line of the ISMF Profile Option menu and press Enter. The User Mode Entry panel (DGTDPPF5) is displayed:

```
DGTDPPF5                      USER MODE ENTRY PANEL
COMMAND ===>

SPECIFY THE FOLLOWING:

  USER MODE  ===> 2    (1 End User or 2 Storage Administrator)







USE ENTER TO PERFORM VERIFICATION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 5. User Mode Entry Panel (DGTDPPF5)

6. The User Mode field on the User Mode Entry panel has a default value of 1 when the panel is displayed.

   Change the User Mode field value to **2** and press Enter to save the field value.

7. Press PF3 to return to the ISMF Profile Option menu.

8. Press PF3 again to return to the ISMF Primary Option menu.

```
DGTSMMD1                 ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===>

  0  ISMF PROFILE    - Change ISMF user profile
                     - Control how ISMF logs and traces errors

  1  FILE            - Access and view attributes of SFS files and
                       directories
                     - Browse, edit, migrate, or recall SFS files
                     - Create and modify lists of SFS files and
                       directories

  3  MANAGEMENT CLASS - Display the attributes of a management class
                     - Create and modify lists of management classes

  L  LIST            - Access, print, filter, sort, and erase
                       previously saved lists

  X  EXIT            - End ISMF using console, log and list defaults









USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 6. ISMF Primary Option Menu for End User (DGTSMMD1)

9. Press PF3 again. When PF3 is pressed this time, a disposition panel (either ISPPFT01, 02, 03, or 04) will be displayed depending on whether or not ISPF logging has been enabled. The Disposition of Console and List File panel is shown in :

```
ISPPFT02 DISPOSITION OF CONSOLE AND LIST FILE --------------------------------
COMMAND ===>

CONSOLE PROCESS OPTION ===> D   (K or D)
LOG FILE WAS NOT USED
LIST PROCESS OPTION    ===> PD  (PD, K, or D)

VALID PROCESS OPTIONS:
  PD - Print file and delete
  K  - Keep file (without printing)
  D  - Delete (erase) file (without printing)

LOG/LIST SPOOL OPTIONS:
  NUMBER OF COPIES ===> 1              SPOOL CLASS ===> A
  BIN NUMBER       ===>                'FOR' USER  ===>
  3800 KEYWORDS    ===>

FOR SPOOLING LOG/LIST TO ANOTHER PERSON OR MACHINE:
  USER/MACHINE ID  ===>
  NODE/LINK ID     ===>
  TAG TEXT         ===>

Press ENTER key to complete ISPF termination.
Enter END command to return to the primary option menu.
```

Figure 7. Specify Disposition of Console and List File Panel (ISPPFT02)

10. Press PF3 again to return to the ISMF Primary Option menu. The Primary Option menu in now configured for a storage administrator. (Note: The panel ID is DGTSMMD2).

```
DGTSMMD2                 ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 2

SELECT ONE OF THE FOLLOWING:

  0  ISMF PROFILE    - Change ISMF user profile
                     - Control how ISMF logs and traces errors
  1  FILE            - Access and view attributes of SFS files and directories
                     - Create and modify lists of SFS files and directories
                     - Browse, edit, migrate, or recall SFS files
  2  MINIDISK        - Create and modify lists of minidisks
                     - Move and check minidisks
  3  MANAGEMENT CLASS - Define, alter, and display management class attributes
                     - Create and modify lists of management classes
  7  ACS             - Create, translate, validate, test, display, and
                       delete Automatic Class Selection routines
  8  CONFIGURATION   - Display or alter configurations
                     - Define, validate and activate a base configuration
  L  LIST            - Access, print, filter, sort, and erase previously
                       saved lists
  X  EXIT            - End ISMF using console, log and list defaults







USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 8. ISMF Primary Option Menu for Storage Administrator (DGTSMMD2)

11. To perform a minidisk CHECK request, type **2** (MINIDISK) on the command line and press Enter. The Minidisk Selection panel is displayed:

```
DGTDNSD1                    MINIDISK SELECTION PANEL              Page 1 of 2
COMMAND ===>

SELECT SOURCE OF MINIDISK LIST  ===> 2     (1 - saved list, 2 - new list)

   1   REUSE A SAVED LIST
          LIST NAME  ===> SAMPLE

   2   GENERATE A NEW LIST FROM CRITERIA BELOW

                              REL OP   VALUE
                              ------ ---------
       DEVICE TYPE        ===>
       DEVICE VOLSER      ===>
       MDISK ADDRESS      ===> EQ     1A4
       MDISK OWNERID      ===> EQ     DFSMS
       MDISK SIZE         ===>
       MDISK START        ===>



USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND FOR NEXT SELECTION PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 9. Minidisk Selection Panel (DGTDNSD1)

12. In order to do a successful check, you will want to check a minidisk that no one is linked to. If you have not already done so in Step "1.a" on page 7, you can release and detach the DFSMS 1A4 disk and issue a check against it. Specify the selection criteria as described in Figure 9 on page 11 shown above. Type **2** in the Select Source Of Minidisk List field and press Enter. The Minidisk List panel (DGTLNML1) is displayed:

```
DGTLNML1                    MINIDISK LIST PANEL
COMMAND ===>                                          SCROLL ===>

ENTER LINE OPERATORS BELOW:                      Data Columns 4-9 of 13

    LINE     MDISK   MDISK MDISK   MDISK   MDISK   MDISK DEVICE  DEVICE
    OPERATOR OWNERID ADDR  START    END    SIZE    GAP   VOLSER  TYPE
   ---(1)---- --(2)--- -(3)- --(4)-- --(5)-- --(6)-- --(7)-- --(8)- --(9)
    CHECK    DFSMS    1A4    393     394      2 ------- SMS90B 3390
   ---------- -------- -------- BOTTOM  OF  DATA  ------- -------- -----



USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 10. Minidisk List Panel (DGTLNML1)

13. Type the CHECK line operator next to the minidisk you want to check and press Enter. The Check Entry panel (DGTDCK01) is displayed:

```
DGTDCK01                       CHECK ENTRY PANEL

 COMMAND ===>

 MINIDISK:  DFSMS     1A4
 FILENAME AND FILETYPE FOR RESULT:  TESTID    0191CHCK


 OPTIONALLY SPECIFY ONE OR MORE:

   OUTPUT DESTINATION USERID  ===> DFSMS     (user ID, 1-8 characters)

   OUTPUT DESTINATION SYSTEM  ===> TUCVM3    (system ID, 1-8 characters)

   OUTPUT RESULTS LEVEL       ===> 1         (1=any result, 2=fail only)

   NUMBER OF RETRIES FOR LINK  ===> 2        (0 to 99)

   LINK RETRY INTERVAL    HOURS ===> 00    MIN ===> 00    SEC ===> 10



 USE ENTER TO CONTINUE.
 USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 11. Check Entry Panel (DGTDCK01)

14. Specify that you want output from CHECK sent to your reader by typing in your user ID (DFSMS in the example above) and system name (TUCVM3 in the example above) on the Check Entry panel. Press Enter. The Minidisk List panel is displayed again. You should receive short messages in the upper-right corner of the screen and on the line below the Command field indicating that the CHECK operation was successful. If desired, press PF1 to verify the message "CHECK issued successfully".

```
DGTLNML1                      MINIDISK LIST PANEL              CHECK ISSUED OK
COMMAND ===>                                              SCROLL ===>
CHECK  issued successfully with request ID: 3
ENTER LINE OPERATORS BELOW:                            Data Columns 4-9 of 13

    LINE     MDISK    MDISK MDISK  MDISK   MDISK   MDISK DEVICE  DEVICE
  OPERATOR  OWNERID   ADDR  START   END     SIZE    GAP  VOLSER  TYPE
  ---(1)---- --(2)--- -(3)- --(4)-- --(5)-- --(6)-- --(7)-- --(8)- --(9)
  *CHECK     DFSMS     1A4   393    394      2 ------- SMS90B  3390
  ---------- -------- -------- BOTTOM  OF  DATA  ------- -------- -----




USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 12. Minidisk List Panel (DGTLNML1)

15. If desired, you can use HELP (PF1) to see the long message associated with the short CHECK message you received. Note the request ID specified as part of the message.

16. When the checking is complete, you can issue the CMS RDRLIST command on the command line of the Minidisk List panel to view the CHECK output. You will receive two reader files if you filled in the panel selection criteria as shown in step "14" on page 11. One of the files is the result of the CHECK operation and the other file is a DFSMS command status file. Review both of these reader files.

The reader files are identified as follows:

```
response filename = global resource ID specified in the DFSMS control file

response filetype = the letter "T" followed by the operation requestor ID

check filename = ownerid of the minidisk being checked

check filetype = vdevCHCK
```

You will always receive a response file. You will receive a CHECK results file if there were CHECK failures or if you requested to see the results of all CHECK output whether there were failures or not.

17. Exit from the reader list by pressing PF3. You are returned to the Minidisk List panel.

18. You can save the minidisk list display by issuing a SAVE command and a list name (for example, TESTINST) from the Minidisk List panel. After you have saved the minidisk list, exit the minidisk list display.

19. To redisplay the saved minidisk list, select Option 1 on the Minidisk Selection panel with the list name you used in the previous step (TESTINST).

20. If desired, you can use the help facility to get an overview of the Minidisk Selection panel. Use HELP (PF1) to invoke the help facility. Navigate through the help hierarchy, if desired, and return to the ISMF Primary Option menu by exiting the help facility and the Minidisk Selection panel.

21. Exit ISMF.

22. If you created a new minidisk for the verification procedure ("1.b" on page 7), you may choose to erase the files and delete the minidisk that was created during verification.

Step 4: Verify Space Management Configuration

In order to verify the configuration of space management, you will need to ensure that you can create and activate configurations.

If you have not previously used space management, creating and activating a configuration file will cause no impact to your system until you specify a file pool as DFSMS/VM-managed. At that time, DFSMS/VM will use the configuration you activate to assign management classes to any new files and directories that are created.

Attention

If you choose the samples we provide, some samples run ACS processing for file create, defer, or inherit. Care should be taken to select a routine that best meets your installation's needs. If you choose to use ACS at the file level, there is a greater performance impact than using ACS at the directory level.

To assist you in this verification and in setting up ACS processing, we have provided you with several sample configurations, ACS routines and a REXX Exit. Refer to Table 1 on page 13 for names of the sample configurations, ACS routines, and REXX exit. These files are located on the LOCALSMP disk (DFSMS 1C2). See Appendix B, "Sample ACS Routines, REXX Exit and Source Configuration Files," on page 89 for details on using the sample ACS routines, REXX Exit, and source configuration files.

Note: If you want to know what is in a sample configuration file before choosing it, use ISMF to display the base configuration information and then to display the management classes. To see what is in an ACS routine, just browse or edit the particular ACS routine file.

Table 1. Sample files for configuration and ACS routines

Configuration		ACS Routine		
File Name	File Type	File Name	File Type	Invoked as
SMPCNFGL	CONFIG	SMPACSL	ACS	INHERIT
SMPCNFG2	CONFIG	SMPACS2	ACS	DEFER
SMPCNFGS	CONFIG	SMPACSS	ACS	YES
SMPCNFGR	CONFIG	SMPREXX	EXEC	INHERIT

1. If you are using a sample configuration, make sure it resides on an SFS file pool and is accessible by this ID. Copy it to an SFS file pool if necessary.

   Copy the REXX exit, or the compiled version if you chose to compile the REXX Exit, into file name IGDACSMC, file type DFSMS, directory name VMSYS:DFSMS.ACSEXITS.

   Note: Because the minidisk verification procedure validated ISMF, you may choose to activate a configuration by specifying the CMS DFSMS ACTIVATE command rather than stepping through the ISMF panels as described below. If you do choose to bypass ISMF, select the configuration file you want to use and ensure that that file is in an SFS directory before issuing the CMS DFSMS ACTIVATE command.

   For more information on the DFSMS ACTIVATE command, issue the HELP DFSMS command.

2. Type ISMF to enter into the ISMF application. If you get the ISMF Primary Option menu for storage administrators (panel DGTSMMD2), proceed to step 3. If you get the ISMF Primary Option menu for end users (DGTSMMD1) instead, then you will need to change your session to storage administrator as discussed in "Step 3: Verify Minidisk Management" on page 7.

3. You are now on the ISMF Primary Option menu for storage administrators, DGTSMMD2 panel. Select Configuration (Option 8), and press ENTER. See Figure 13 on page 14.

```
DGTSMMD2                    ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 8

SELECT ONE OF THE FOLLOWING:

  0  ISMF PROFILE    - Change ISMF user profile
                     - Control how ISMF logs and traces errors
  1  FILE            - Access and view attributes of SFS files and directories
                     - Create and modify lists of SFS files and directories
                     - Browse, edit, migrate, or recall SFS files
  2  MINIDISK        - Create and modify lists of minidisks

                     - Move and check minidisks
  3  MANAGEMENT CLASS - Define, alter, and display management class attributes
                     - Create and modify lists of management classes
  7  ACS             - Create, translate, validate, test, display, and
                       delete Automatic Class Selection routines
  8  CONFIGURATION   - Display or alter configurations
                     - Define, validate and activate a base configuration
  L  LIST            - Access, print, filter, sort, and erase previously
                       saved lists
  X  EXIT            - End ISMF using console, log and list defaults







USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 13. ISMF Primary Option Menu for storage administrators (DGTSMMD2)

4. Input the file name, file type, and file mode of the sample configuration you wish to activate. Refer to Table 1 on page 13, for examples of sample configuration file names. Choose Option 5 to perform the activation. See Figure 14 on page 14.

```
DGTSBSA1           CONFIGURATION APPLICATION SELECTION PANEL
COMMAND ===>

TO PERFORM CONFIGURATION OPERATIONS, SPECIFY:

CONFIGURATION FN FT ===> SMPCNFGL CONFIG     (File name and type or 'ACTIVE')
FM/DIRID ===> B
        ===>
        ===>                                 (File mode letter or directory name)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 5

  1  DISPLAY  - Display base configuration information
  2  DEFINE   - Define a base configuration
  3  ALTER    - Alter a base configuration
  4  VALIDATE - Validate a source configuration
  5  ACTIVATE - Activate a configuration


USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 14. ISMF Configuration Application Selection Panel (DGTSBSA1)

5. If you are using an ACS routine only (no REXX and no module exit), when the next panel comes up, you will see EXITS NOT FOUND in the upper right corner. This is just a warning to you and can be ignored. Select *Y* to confirm. See Figure 15 on page 14.

```
DGTDCAC1              ACTIVATE REQUEST CONFIRMATION PANEL      EXITS NOT FOUND
COMMAND ===>

  TO CONFIRM ACTIVATION ON THE FOLLOWING CONFIGURATION:

CONFIGURATION FN FT : SMPCNFGL CONFIG
  FM: A DIRID:  VMSYSU:VMTEST0.


SPECIFY THE FOLLOWING:
     PERFORM ACTIVATION  ===>  Y          (Y OR N)




USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 15. ISMF Activate Request Confirmation Panel (DGTDCAC1)

6. You should see **Activation successful** in the upper right corner of your screen.

```
DGTSBSA1      CONFIGURATION APPLICATION SELECTION PANEL     ACTIVATION SUCCESSFUL
COMMAND ===>
TO PERFORM CONFIGURATION OPERATIONS, SPECIFY:

CONFIGURATION FN FT ===> SMPCNFGL CONFIG        (File name and type or 'ACTIVE')
FM/DIRID ===>
         ===>
         ===>                        (File mode letter or directory name)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 1

  1  DISPLAY  - Display base configuration information
  2  DEFINE   - Define a base configuration
  3  ALTER    - Alter a base configuration
  4  VALIDATE - Validate a source configuration
  5  ACTIVATE - Activate a configuration


USE ENTER TO PERFORM ACTIVATION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 16. ISMF Configuration Application Selection Panel (DGTSBSA1)

7. Type 'ACTIVE' in the CONFIGURATION FN FT field. Select Option 1, and press ENTER to verify the information. The Base Configuration Display panel reflects the configuration you activated.

This completes the configuration verification process. If desired, now that you have this active configuration, you may want to look at management classes, study the management class application, look at the ACS routine to see how management classes are assigned, and become familiarized with panels.

Step 5: Verify Removable Media Services (RMS)

Verify the removable media services (RMS) function by doing the procedures described below.

Start the RMS master Virtual Machine

This section demonstrates the following:

- DFSMS/VM software has been installed correctly
- Library tape drives are installed properly
- Data files (DGTVCNTL DATA and RMCONFIG DATA) have been edited appropriately.

Do the following steps to start the RMS virtual machine:

1. Preset the RM_LOG_TO_CONSOLE Control File parameter message level to 4 to ensure that sufficient information will be displayed to identify any problems that occur.

2. Ensure that library tape drives are free (not attached to any virtual machine) during the verification test. RMS master virtual machine initialization can help you demonstrate that all library tape drives are installed properly. Messages are written to the console if a problem is encountered with any drive.

3. Log on to the RMS master virtual machine and watch it initialize.

   If you do not have RACF/VM or its equivalent installed, you will receive the following error messages:

```
RPICMS009W IUCV communication has been severed
   RPICMS011E You are not authorized to connect to RACF.
   ***
   *** Error initiating connection to RACF/VM, return code = 8
   *** DFSMS/VM authority checking will be done via the authorization file
   ***
```

   Ignore these messages since you have not yet set up RACF/VM authorization. These messages indicate that you will be using the DFSMS/VM authorization file instead of RACF/VM.

   During the initialization of the RMS master virtual machine, any errors that might stop the initialization are written to the console. In particular, watch for messages indicating that individual drives specified in the RMS configuration file can not be initialized. If you receive any errors, resolve them before continuing. See *z/VM: DFSMS/VM Messages and Codes*, *z/VM: DFSMS/VM Diagnosis Guide*, and "Appendix E. Common Problems and Solutions" in *z/VM: DFSMS/VM Removable Media Services* for error messages and problem resolution.

Issue Requests to the RMS Master Virtual Machine

This section demonstrates the following:

- Authorization control is operational
- User interface software is functional and accessible
- RMS master can perform library I/O functions.

Do the following steps to issue a request to the RMS virtual machine:

1. Ensure that the user ID used to issue test requests is authorized for the Query Library function. The user ID must link and access the DFSMS 1B5 minidisk and be authorized to issue RMS commands, as defined in Chapter 4, "Authorizing DFSMS/VM Users," on page 33.

2. Issue the command:

   ```
   DFSMSRM QUERY LIBRARY OPSTATE
   ```

   You should receive a response indicating the status of library operations. The IBM 3495 Tape Library Dataserver does not need to have any volumes in it, but library tape drives must be installed and functional. See *z/VM: DFSMS/VM Removable Media Services* for more details.

3. Ensure that the CSL routine FSMRMQLB is loaded. Do this by issuing the following CMS command:

   ```
   RTNLOAD * (FROM FSMPPSI
   ```

   Note: The RTNLOAD command is only required for applications which will be using the CSL routines.

4. Issue the DFSMSRM QUERY LIBRARY OPSTATE request by means of the CSL-routine interface. The REXX EXEC given below shows a sample invocation of the FSMRMQLB routine.

Sample FSMRMQLB Routine Invocation

The following example shows a sample FSMRMQLB routine invocation.

```
/**********************************************************************/
/* Program to query the operational status of the 3495 and          */
/*           verify the CSL interface installation                  */
/**********************************************************************/
signal on novalue
/***********************************/
/* Set up the FSMRMQLB parameters    */
/***********************************/

cslrc = 0                          /* Initial return and reason codes */
cslreas = 0
qryrc = 0
qryreas = 0
reqtoken = 1                       /* Request asynchronous processing  */
qrytyp = 'OPS'                     /* Ask for 3495 operational state   */
libname = ' '                      /* Have library name returned       */
libname = left(libname,32,' ')
length1 = 32
vlabel = ''                        /* n/a for OPS invocation           */
length2 = 0
status = '0000'X                   /* Initialize status field          */
volcateg = ''                      /* n/a for OPS invocation           */
length3 = 0
volclass = ''                      /* n/a for OPS invocation           */
voltype = ''                       /* n/a for OPS invocation           */
volmntd = ''                       /* n/a for OPS invocation           */
asscatg = ''                       /* n/a for OPS invocation           */
length4 = 0
count = 0                          /* n/a for OPS invocation           */

/***************************************/
/* Call FSMRMQLB until the             */
/* request completes.                  */
/***************************************/

do forever

  call CSL 'FSMRMQLB cslrc cslreas qryrc qryreas reqtoken',
```

```
            'qrytyp libname length1 vlabel length2 status',
            'volcateg length3 volclass voltype volmntd',
            'asscatg length4 count'

   if cslrc ¬= 2 then leave

end
```

```
if cslrc > 0 & cslreas = 0 then do      /* Error in RMS processing  */
  say 'Return code is 'qryrc
  say 'Reason code is 'qryreas
end
else do                                 /* Error in CSL processing  */
  say 'Return code is 'cslrc
  say 'Reason code is 'cslreas
end
if (cslrc = 0) | (cslrc = 4) then do    /* Success or warning cond  */
  status = B2X(status)
  say "Library operational status is X'"status"'"
  select
    when status = '0000'  then say 'Library is in auto mode.'
    when status = '0100'  then say 'Library is in paused mode.'
    when status = '0200'  then say 'Library is in manual mode.'
    otherwise say 'Library is in an unknown state.'
  end
end
else do                                 /* Unsuccessful             */
  say 'See RMS User Guide and Reference for',
      'meaning of a non-0 reason code.'
  say 'See CMS Application Development Reference',
      'for meaning of a negative return code.'
  say 'CSL call was not completed successfully.'
end

exit

novalue:
say 'Signal-on-novalue occurred at line' sigl
exit (99)
```

Chapter 3. Postinstallation and Optional Customization

This chapter discusses optional customization and postinstallation procedures for DFSMS/VM. These functions should be performed after the verification process.

Note: If you installed only the minidisk management or RMS functions, you will only need to do the first sections of this chapter:

> "Step 1: Perform Optional Customization" on page 19,
> "Step 2: Verify Log File" on page 19, and
> "Step 3: Enable the DFSMS/VM Accounting Function" on page 20.

If you installed space management, do all of the sections described in this chapter.

Step 1: Perform Optional Customization

Once you have verified the correct installation of DFSMS/VM, you can consider the various customization options.

Authorization

Based on the authorization scheme you have selected for your installation, make the necessary authorization updates. See Chapter 4, "Authorizing DFSMS/VM Users," on page 33.

DFSMS/VM Master/Server Operation

The DFSMS/VM control file provides an extensive set of keywords that you can use to modify operational parameters associated with the master and server virtual machines. Refer to Chapter 5, "Customizing the DFSMS/VM Control File," on page 41 for details on modifying the DFSMS/VM control file keyword values.

Installation-Wide Exits

An extensive set of installation-wide exits are provided by DFSMS/VM. Refer to Chapter 6, "Using DFSMS/VM Installation-Wide Exits," on page 57 for descriptions of them.

Step 2: Verify Log File

If you have specified that messages are to be placed in a log file by putting the DFSMS_LOG_TO_FILE keyword in the control file, then in order to view the log files you must access the file space that was specified on this keyword. The file name of a log file is the name of the master or server machine that created it. The file type is the date that the log file was created. A new log file is started with the first request after midnight of each day and is the active file for that day.

The following are examples of log file IDs:

```
SMSMASTR 19930331
SMSSRV01 19930330
DGTSRV01 19930401
```

XEDIT each log file with the NOLOCK option. Enter *VERIFY 1 ** to see the entire length of the records. Browse each file and look for warnings or error messages. Refer to *z/VM: DFSMS/VM Messages and Codes* or online help messages for details on and format of messages.

If you are not using log files, then browse the spooled console for warnings or error messages (if the consoles are spooled.)

Step 3: Enable the DFSMS/VM Accounting Function

The DFSMS/VM accounting functions generate user and system accounting records which provide information such as CPU utilization and storage resources used. See *z/VM: DFSMS/VM Storage Administration*, for more information on the accounting interface.

To enable the accounting function, your system must have been set up for CP accounting. See *z/VM: CP Planning and Administration*. Once you have determined that your system is set up for CP accounting:

- Update the accounting fields in the DFSMS/VM Control File to 'Y'. See "Control File Keywords Required by Both Space and Minidisk Management" on page 43 for a description of the DFSMS_ACCOUNTING control file keyword and "Control File Keywords Required by Removable Media Services (RMS)" on page 52 for a description of the RM_ACCOUNTING control file keyword.
- Ensure the directory entries for the DFSMS/VM virtual machines include the ACCT parameter on the OPTION statement. This is necessary in order to give a virtual machine the authority to generate CP accounting records. See the sample product CP directory entries listed in Appendix A, "DFSMS/VM Sample CP Directory Entries," on page 81.

Step 4: Define a File Space for Migration Level 1 (ML1)

If you choose to use the space management migration capability of DFSMS/VM, you need to set up a storage area for ML1. This ML1 storage area is an SFS file space defined in an SFS file pool.

Important

It is best if ML1 is defined in its own file pool. However, if ML1 resides on a file pool that has user data that is DFSMS/VM-managed, then the ML1 data must reside in a distinct storage group.

Select a file pool and file space into which ML1 data will be stored. The first five characters of the file space name must be set to DFSMS. The three remaining characters (positions 6-8) may be anything you choose, or left blank. Enroll the file space in the file pool, ensuring that there is sufficient space to contain all migrated data.

The amount of space needed for ML1 is contingent on each installation's data reference patterns and the management classes established. However, the following formula can be used as a guideline:

$$\left[\text{Number of 4K blocks in all DFSMS/VM-managed local and remote file pools}\right] * \left[\text{Percentage of data that you expect to migrate to ML1}\right] * \left[\frac{1}{3}\right]$$

Enroll this file space by issuing the following command:

```
ENROLL USER DFSMSxxx myfp (BLOCKS blks STO storgrp
```

where,

DFSMSxxx
    is the name of the file space selected to be enrolled in the file pool

myfp
    is the file pool in which the ML1 file space is to be enrolled

blks
    is the number of blocks determined by using the formula described above

storgrp
    is the storage group where the file space will be enrolled.

You will need to update the MIGRATION_LEVEL_1 control file parameter (DGTVCNTL DATA in VMSYS:DFSMS.CONTROL) with the name of the file pool and file space you have chosen for ML1. GLOBAL_RESOURCE_ID and FULLY_QUALIFIED_LUNAME are used in creating the ML1 directory and they

ensure that ML1 and ML2 data can be uniquely identified across a TSAF collection. Ensure that these two keywords are also updated according to your installation's environment.

Refer to Chapter 5, "Customizing the DFSMS/VM Control File," on page 41 for more information on modifying the control file.

Note:

1. If only ML2 will be used for storage, you must still specify a file space for ML1, even though it can be of minimal size.

2. Ensure that the amount of catalog (storage group 1) space on your ML1 file pool is sufficient for your installation's data reference patterns and the management classes established. See *z/VM: CMS File Pool Planning, Administration, and Operation* for more details.

Update the ML1 File Pool for DFSMS/VM

You will need to make several changes to the DMSPARMS file of the migration level file pool. To modify the DMSPARMS file, you will have to stop the file pool server, make the modifications to the DMSPARMS file, and then restart the file pool server. Grant ML1 file pool administrator authority to all space management virtual machines and the DFSMS/VM master virtual machine by modifying the DMSPARMS file on the ML1 file pool server's 191 disk to include all of these virtual machines as administrators. If ML1 is defined as its own file pool, include the NODFSMS statement in the DMSPARMS file.

Increase the USERS parameter by the following formula:

```
        ( # tasks in all DFSMS servers * 8 ) + 50.
```

The default setup (3 servers and 8 tasks in each server) should be 242 (24*8 + 50).

ML1 can be defined in a remote file pool. See *z/VM: CMS File Pool Planning, Administration, and Operation* for more details. Remember that ML1 recalls take longer and data may be unavailable if the remote systems or data links go down.

Increase the MAXCONN value in the file pool server's directory entry by the following value:

```
        # tasks in all DFSMS servers * 2.
```

Step 5: Define a Server for Migration Level 2 (ML2)

Note: If the Migration Level 2 (ML2) function of DFSMS/VM is used, Tivoli® Storage Manager Extended Edition for z/OS and z/VM, V5.2 (5698-A11), Tivoli Storage Manager for z/OS and z/VM, V5.2 (5698-A13), or Tivoli Storage Manager for VM (5697-TS9) is required. All three of these program numbers have been withdrawn from marketing, and service has been discontinued.

ML2 data is stored in a TSM server. It is recommended that this TSM server be dedicated exclusively for use by DFSMS/VM, to keep ML2 data separate from other clients' data. Also, this server must be installed on the same z/VM system as the DFSMS/VM master and server machines.

Important

The TSM server EXPORT and IMPORT commands cannot be used to backup and restore a TSM server that contains DFSMS/VM ML2 data. Using EXPORT and IMPORT will cause ML2 data to be lost. See "Back Up and Restore a TSM Server That Contains ML2 Data" on page 27.

If you have decided to set up a TSM server for ML2 data, autolog that server at system IPL time.

To ensure that TSM works with DFSMS/VM, the TSM server must be set up correctly. This set up must be done from the TSM server or from a TSM administrator user ID with system privilege. Set up includes the following steps:

- Planning for TSM database and recovery log space
- Tailoring TSM server options file parameters

- Establishing TSM registration and authentication
- Defining a TSM policy domain
- Defining a TSM policy set
- Defining a TSM default management class
- Setting up TSM archive storage pools for ML2 data
- Defining TSM archive and backup copy groups
- Activating a policy set
- Registering DFSMS as a TSM node
- Backing up and restoring a TSM server that contains ML2 data.

TSM provides you with a standard set of policy objects that are activated during TSM installation. These policy objects include:

- Policy domain
  - Policy set
    - Management class
      - Backup copy group
      - Archive copy group.

Each of these policy objects, activated automatically during installation, is named **Standard**. If this TSM server is dedicated as the DFSMS/VM ML2 repository, the **Standard** policy domain, policy set, and management class work may be used. If this TSM server services other clients, you should define a policy domain, policy set, and management class for the exclusive use of DFSMS/VM.

Plan for TSM Database and Recovery Log Space

Set up your TSM database and recovery log according to the procedures outlined in the TSM documentation. That documentation also contains formulas for estimating the space required for your database and recovery log.

Note that the database capacity depends only on the number of files stored in TSM, and not on the amount of data associated with those files. To estimate the amount of space required for the TSM database for ML2:

- Estimate the number of files that will be migrated to ML2
- Estimate the number of ML2 backup versions that will be kept (number of backups that will be done before the DFSMS/VM DELETE ML2BACKUP command is used to erase old ML2 backup entries in TSM)
- Determine the size of the activity log.

Use the following to estimate the number of entries to be placed into TSM:

```
total # files in TSM = (# backup versions + 2) * number ML2 files
```

The size of the activity log depends on the length of time you want to keep the messages generated by the TSM server.

Tailor TSM Server Options File Parameters

There are recommended settings for two TSM server options file parameters:

- IDLETIMEOUT specifies the number of minutes the TSM server will wait before an idle client session is terminated. The DFSMS clients to the ML2 TSM server are special clients in that they are also server machines performing System-Managed Storage (SMS) functions. DFSMS clients may have TSM sessions open while performing these other SMS functions. It is recommended that IDLETIMEOUT be set at least as long as the longest running MANAGE command for the installation.

- MAXSESSIONS sets the maximum number of simultaneous sessions allowed between the TSM server and its clients (for example, DFSMS/VM virtual machines). A guideline for this number is

```
max_value = (3 * total_#_DFSMS_server_tasks) + 5
+ total_#_other_administrators
```

where `total_#_other_administrators` is the total number of administrators to this TSM server.

The DFSMS/VM virtual machines do not require an Admin Client Options file. However, the name of the TSM server (usually specified in Admin clients as the SERVERNAME parameter in the Admin Client Options file) must be specified on the MIGRATION_LEVEL_2 parameter in the DFSMS/VM control file, DGTVCNTL DATA.

Establish TSM Registration and Authentication

Issue the following commands to establish proper TSM registration and authentication practices for use by DFSMS/VM:

```
SET REGISTRATION CLOSED
SET AUTHENTICATION ON
SET PASSEXP 9999
```

Registration can be opened or closed. To control which clients have access to this TSM server, set registration to closed. You may want to dedicate this TSM server to DFSMS/VM to avoid workstation backups impacting z/VM space management.

Authentication can be set on or off. If authentication is set off, ML2 data can be accessed by unauthorized users. It is recommended that authentication be set to on to require password authorization before clients can access the TSM server.

PASSEXP indicates the number of days before you must change the DFSMS/VM to TSM password. When this period elapses, DFSMS/VM must be shutdown, the MIGRATION_LEVEL_2 parameter in the control file updated with a new password, and the TSM UPDATE NODE command used to set the new password for the DFSMS node. To avoid an impact on DFSMS/VM processing while updating the password, use a longer password expiration period. Since the DFSMS/VM control file contains this password, carefully control access to this control file.

Define a TSM Policy Domain

Define a TSM policy domain for exclusive use by DFSMS/VM. TSM installation activates a policy domain named **Standard**. If the TSM server is dedicated as the DFSMS/VM ML2 repository, the **Standard** policy domain can be used for DFSMS/VM.

If the TSM server services other clients, define another policy domain for exclusive use by DFSMS/VM. This can be done with the following command:

```
DEFINE DOMAIN dfsms_domain_name
```

where,

dfsms_domain_name
    can be any name you want to assign to the policy domain to be used by DFSMS/VM (for example, DFSMS).

You can verify this step by issuing the following command:

```
QUERY DOMAIN dfsms_domain_name FORMAT=DETAIL
```

Define a TSM Policy Set

Define a TSM policy set for exclusive use by DFSMS/VM. TSM installation activates a policy set named **Standard**. If the TSM server is dedicated as the DFSMS/VM ML2 repository, the **Standard** policy set can be used for DFSMS/VM.

If this TSM server services other clients, define another policy set for exclusive use by DFSMS/VM. This can be done with the following command:

```
DEFINE POLICYSET dfsms_domain_name dfsms_set_name
```

where,

dfsms_domain_name
  is the name of the policy domain dedicated to DFSMS/VM.

dfsms_set_name
  can be any name you want to assign to the policy set to be used by DFSMS/VM (for example, DFSMS).

You may want to verify this step by issuing the following command:

```
QUERY POLICYSET dfsms_domain_name dfsms_set_name FORMAT=DETAIL
```

Define a TSM Default Management Class

DFSMS/VM supports management classes containing space management criteria for migrating and expiring SFS files. TSM introduces its own management classes that specify criteria to manage data you put into TSM.

Set up one TSM management class for exclusive use by DFSMS/VM and make it the default management class for your DFSMS/VM policy set. TSM installation activates a management class named **Standard**. If the TSM server is dedicated as the DFSMS/VM ML2 repository, the **Standard** management class can be used for DFSMS/VM.

If the TSM server services other clients, define another management class for exclusive use by DFSMS/VM and make it the default management class. This can be done with the following commands:

```
DEFINE MGMTCLASS  dfsms_domain_name  dfsms_set_name  dfsms_mc_name
ASSIGN DEFMGMTCLASS dfsms_domain_name  dfsms_set_name dfsms_mc_name
```

where,

dfsms_domain_name
  is the name of the policy domain dedicated to DFSMS/VM.

dfsms_set_name
  is the name of the policy set dedicated to DFSMS/VM.

dfsms_mc_name
  can be any name you want to assign to the management class to be used by DFSMS/VM (for example, DFSMS).

You can verify this step by issuing the following command:

```
QUERY MGMTCLASS dfsms_domain_name  dfsms_set_name  dfsms_mc_name
                FORMAT=DETAIL
```

Set Up TSM Archive Storage Pools for ML2 Data

Define the archive storage pools you want to use and assign volumes to those storage pools, using the DEFINE STGPOOL and DEFINE VOLUME commands. During installation, TSM provides two predefined disk storage pools, BACKUPPOOL and ARCHIVEPOOL. DFSMS does not use the backup copy group (BACKUPPOOL), but does use the archive copy group (ARCHIVEPOOL).

If you want to place ML2 data initially on DASD and move it off to tape later, a tape storage pool can be defined (using the DEFINE STGPOOL and DEFINE VOLUME commands). Then the UPDATE STGPOOL command on poolname ARCHIVEPOOL with operand NEXTSTGPOOL may be used to specify the tape storage pool into which ML2 data can be moved.

If you want to place ML2 data directory onto tape in the ARCHIVEPOOL storage pool, you can use the DELETE STGPOOL command to delete ARCHIVEPOOL, and then use the DEFINE STGPOOL command to define ARCHIVEPOOL as a tape storage pool.

If you want to place ML2 data directory onto tape, but do not want to place it into the storage pool with name ARCHIVEPOOL, you can use the DEFINE STGPOOL command to define the tape storage pool with the desired name.

Note: Both DFSMS/VM and TSM use the word **migrate**, but each product has its own definition of **migrate**. In TSM documentation, it refers to the movement of files from one storage pool to another. In DFSMS/VM it means moving a file's data to secondary storage and marking the file **migrated** in the SFS catalog. There is no correlation between DFSMS/VM and TSM; to TSM, DFSMS/VM is just another client storing data into its archive repository. Keep this in mind as you use both products to support the ML2 function.

The storage pools assigned to the DFSMS/VM archive copy group must be able to contain all ML2 data that will be migrated there. Storage volumes can be added to a storage pool at any time.

The amount of space needed for ML2 is contingent on each installation's data reference patterns and the management classes established. However, the following formula can be used as a general guideline.

$$\left[\text{(Number of 4K blocks in all DFSMS/VM-managed local and remote file pools)} \right] * \left[\text{(Percentage of data that you expect to migrate to ML2)} \right] * \left[\frac{1}{3} \right]$$

Define TSM Archive and Backup Copy Groups

TSM allows one archive copy group and one backup copy group per management class. The name of each of these two copy groups must be **Standard**.

DFSMS/VM ML2 data is placed in the archive copy group of the default management class in the active policy set of the policy domain in which the DFSMS node is registered. In addition, DFSMS/VM requires that a small amount of information be placed in the backup copy group of the default management class.

The archive copy group must be set up with enough storage pool space to handle all ML2 data. Also, it is essential that the archive copy group used by DFSMS/VM has the RETVER parameter set to NOLIMIT. This parameter tells TSM not to expire data in this copy group; the client (in this case DFSMS) manages expiration of data. If the RETVER parameter is not set to NOLIMIT, TSM may erase your ML2 data.

TSM requires that a backup copy group be defined for a default management class. Because DFSMS/VM places a small amount of information in the database for the backup copy group, the backup copy group must be defined. However, DFSMS/VM does not require any data to be stored in the backup repository. Therefore, the backup copy group need not have a storage pool defined. Make sure that the backup copy group is defined with options RETEXTRA and RETONLY both set to the value NOLIMIT.

If the TSM server is dedicated as the DFSMS/VM ML2 repository, the **Standard** archive copy group can be used for DFSMS/VM, but it must be modified with the following command:

```
UPDATE COPYGROUP dfsms_domain_name dfsms_set_name
                 dfsms_mc_name STANDARD Type=Archive DEST=poolname
                 FREQUENCY=CMD RETVER=NOLIMIT MODE=ABSOLUTE
```

If you are creating a new archive copy group for use by DFSMS/VM, use the following command:

```
DEFINE COPYGROUP dfsms_domain_name dfsms_set_name
                 dfsms_mc_name STANDARD Type=Archive DEST=poolname
                 FREQUENCY=CMD RETVER=NOLIMIT MODE=ABSOLUTE
```

where,

dfsms_domain_name
　　is the name of the policy domain dedicated to DFSMS/VM.

dfsms_set_name
    is the name of the policy set dedicated to DFSMS/VM.

dfsms_mc_name
    is the name of the management class dedicated to DFSMS/VM.

poolname
    is the name of the archive storage pool to be used as the ML2 data repository for DFSMS/VM.

Since the purpose of ML2 is to put data on lower cost storage devices, *poolname* will most likely be set up as a tape storage pool. However, TSM tape and DASD storage pools can be chained together allowing multiple layers of storage within ML2. You can make *poolname* a DASD storage pool and then have it point to a tape storage pool, *poolname2*. This allows you to stage ML2 data on DASD and migrate it to ML2 tape at a later time (for example, when tape operators are available).

Migration of data from one TSM storage pool to another is done automatically through TSM storage pool thresholds. Be aware that you probably do not want to set up more than one layer of tape storage pools since this requires unnecessary tape mounts to move ML2 data from one storage pool to another within ML2.

You may want to verify this step by issuing the following command:

```
QUERY COPYGROUP dfsms_domain_name  dfsms_set_name dfsms_mc_name
               STANDARD FORMAT=DETAIL TYPE=ARCHIVE
```

Activate a Policy Set

Activate the policy set you defined for DFSMS/VM by issuing the following command:

```
ACTIVATE POLICYSET dfsms_domain_name dfsms_set_name
```

where,

dfsms_domain_name
    is the name of the policy domain dedicated to DFSMS/VM.

dfsms_set_name
    is the name of the policy set dedicated to DFSMS/VM.

Register DFSMS as a TSM Node

Issue the following command to register DFSMS as a node with the TSM server:

```
REGISTER NODE DFSMS password DOMAIN=dfsms_domain_name
                    COMPRESSION=NO ARCHDELETE=YES BACKDELETE=YES
```

where,

dfsms_domain_name
    is the name of the policy domain you have setup for exclusive use by DFSMS/VM, or STANDARD if you are using the standard policy domain.

COMPRESSION
    DFSMS/VM already performs compression on data being placed in ML2. Therefore, the COMPRESSION option of TSM should be set to NO.

ARCHDELETE/BACKDELETE
    ARCHDELETE and BACKDELETE must be set to YES in order for DFSMS ML2 functions to work.

DFSMS/VM puts ML2 data into the TSM server under the DFSMS node name. If authentication is set to on (the recommended setting), the password specified here must also be specified on the MIGRATION_LEVEL_2 parameter in the DFSMS/VM control file (DGTVCNTL DATA in VMSYS:DFSMS.CONTROL). You also specify the virtual machine identifier of the TSM server on this MIGRATION_LEVEL_2 parameter. After you change the control file, you need to stop and restart the DFSMS/VM master virtual machine in order for that change to take effect. See Chapter 5, "Customizing the DFSMS/VM Control File," on page 41 for more information on the MIGRATION_LEVEL_2 parameter.

You may want to verify this step by issuing the following command:

```
QUERY NODE DFSMS FORMAT=DETAIL
```

Note: Do not delete any data on the ML2 server under the DFSMS node.

Back Up and Restore a TSM Server That Contains ML2 Data

It is highly recommended that all files are backed up before they are migrated to either migration level 1 or migration level 2. In either case, you may still wish to backup the TSM server containing your ML2 data. This cannot be done with TSM EXPORT and IMPORT commands. Instead, use the following procedure to backup the TSM server:

1. Stop the TSM server.
2. Backup the TSM 191 disk.
3. Use DFSMS COPY or an equivalent product to backup all TSM recovery log and database volume minidisks. An equivalent product must be able to handle files created with the CMS RESERVE command. The target minidisks must be the same size and device type as the source minidisks.

   Use this same procedure to backup any disk volumes used for storage pool space, if any.
4. Copy all tape volumes.

To restore the TSM server, do the following:

1. Stop the TSM server.
2. Restore all TSM minidisks from the backups using DFSMS COPY or an equivalent product that can properly handle CMS RESERVE minidisk files.
3. After the TSM server has been restored, use the tape volume copies that were created during the backup operation to work with from now on.

See *z/VM: DFSMS/VM Storage Administration* for details on resynchronizing ML2 with primary storage.

Step 6: Stop and Restart SMSMASTR

After you change the control file and update the ML1 file pool and the ML2 server, you will need to stop and restart the DFSMS/VM master virtual machine (SMSMASTR) in order for the control file parameters to take effect. See *z/VM: DFSMS/VM Storage Administration* for details on starting and stopping the DFSMS/VM master virtual machine.

Step 7: Verify Secondary Storage Enrollment

If ML1/ML2 are defined and access is successful, you will receive the messages described below the next time the DFSMS virtual machine is initialized.

If you have enrolled ML1 properly, when DFSMS initializes, message 3018 will be displayed on the SMSMASTR virtual machine:

```
FSMZCG3018I Migration level 1 file space is MIGLEV01:DFSMS.
```

If you have enrolled ML2 properly, when DFSMS initializes, message 3250 will be displayed on the SMSMASTR virtual machine:

```
FSMZCI3042W ML2 password not specified in the DFSMS control file
FSMZCI3250I Migration level 2 server is DSSERV
```

Note: The ML2 password message is dependent on whether or not you have specified a password for the ML2 server.

If you have enrolled both ML1 and ML2 properly, both messages will be displayed on the SMSMASTR virtual machine:

If ML1 is not properly enrolled, when DFSMS is initialized, message 3220 will be displayed on the SMSMASTR virtual machine:

```
FSMZCG3220E File space portion of the MIGRATION_LEVEL_1 parameter must be an
enrolled user of the file pool specified on that parameter
```

If a file pool is not available or an invalid file pool is specified in the control file, message 3025 will be displayed on the SMSMASTR virtual machine and initialization will continue.

```
FSMZCG3025W File pool unavailable for migration level 1 file space MIGLEVL1:DFSMS.
```

Step 8: Specify File Pools to be Managed by DFSMS/VM

The DFSMS master and servers need administrator authority to any file pool that will be DFSMS/VM-managed. Additionally, SFS must be told that these file pools will be managed by DFSMS/VM.

Update the DMSPARMS File

For each DFSMS/VM-managed file pool, the DMSPARMS file must be modified as follows:

1. Add an ADMIN statement for the DFSMS/VM master and all space management servers.
2. Increase the USERS parameters using the following equation:

```
        # of tasks on all servers * 8
```

3. Use the keyword DFSMS to indicate that this is a DFSMS/VM-managed file pool. For example, add the following statements (the virtual machine names are the defaults, your names may be different than those shown) to the DMSPARMS file:

```
    ADMIN SMSMASTR
    ADMIN SMSSRV01
    ADMIN SMSSRV02
    ADMIN SMSSRV03
```

If the keyword NODFSMS exists, change it to DFSMS. If the DFSMS keyword does not exist, add it.

Attention

Once the DFSMS keyword is placed in the DMSPARMS file, and the file pool is restarted, DFSMS/VM will be active for that file pool and will begin assigning management classes based on your installation's active configuration.

Update the File Pool PROFILE EXEC and SETUP EXEC

All DFSMS/VM-managed file pools will need access to the DFSMS/VM product code. You can do so either by adding statements to the PROFILE EXEC or by adding a link statement to the file pool directory entry (see). If you choose to gain access by updating the PROFILE EXEC, place the following statements in the PROFILE EXEC of the file pool server:

```
    CP LINK DFSMS 01B5 0192 RR
    ACCESS 0192 D
```

You must specify a value for the read password if the minidisk is protected by a password.

D is the mode letter used in the example, but the mode letter can be any available letter A through Z.

You should also ensure that the server will always be able to access the segment. (Refer to the *Program Directory for function level 221* for information about defining a saved segment.) Do this by including

SEGMENT RESERVE DFSMSSEG in the PROFILE EXEC when the segment is defined at a location within the virtual machine:

```
SEGMENT RESERVE DFSMSSEG
```

If you created a SETUP EXEC for your file pool (see *z/VM: CMS File Pool Planning, Administration, and Operation*), include the same link and access statements in the SETUP EXEC as you did for the PROFILE EXEC. The SETUP EXEC will not need the SEGMENT RESERVE statement.

Update the File Pool Directory Entry

The MAXCONN value in the file pool servers' directory entry should be increased by the following value:

```
# tasks on all space management servers * 2
```

The NAMESAVE DFSMSSEG statement must be included in the directory entry of any SFS file pool server managed by DFSMS/VM. (Refer to *Program Directory for function level 221* for information about defining a saved segment.)

If you did not choose to add statements to the PROFILE EXEC of the file pool server to gain access to the production disk, you must then add a LINK statement to the file pool server virtual machine directory entry:

```
LINK DFSMS 01B5 0192 RR
```

Since CMS automatically accesses a minidisk at virtual address 0192 as "D", no ACCESS statement is needed in the PROFILE EXEC of the file pool server when 0192 is used. However, if the "linked-as" virtual address (0192 in the above example) is not 0192, you must include an ACCESS statement in the PROFILE EXEC of the file pool server. For example, the following statement is placed in the directory entry for the file pool server:

```
LINK DFSMS 01B5 039F RR
```

Then the following ACCESS statement must be placed in the PROFILE EXEC of the file pool server:

```
ACCESS 039F D
```

"D" is the mode letter used in the example, but the mode letter can be any available letter A through Z.

Authorize the File Pool to DFSMS/VM

If you are using the authorization file (DGTVAUTH DATA in VMSYS:DFSMS.CONTROL), be sure to add the user IDs of all file pool server machines that are managed by DFSMS/VM to the authorization file. If you are using RACF/VM or an equivalent, be sure the file pool is correctly authorized. See Chapter 4, "Authorizing DFSMS/VM Users," on page 33.

Verify Segment Access and File Pool Setup

After you have made the changes listed above, bring up the file pool. Check the messages displayed on the console carefully. If you see any of the following messages, you are not correctly linked to the DFSMS/VM product code on the production disk (DFSMS 1B5). You will need to stop the file pool server and update the PROFILE EXEC or CP directory entry appropriately.

```
DMSCRL1136E Unable to gain access to library FSMINT

DMS5SH2030E Initialization error in exit routine SMSDFSMS
DMS5SH2030E Return code = -7, reason code(s) 0
```

```
DMSDCS283E The DFSMSSEG saved segment could not be reserved; return code
from SEGMENT RESERVE

DMS5SH2029I Initialization begins for DFSMS exit routine SMSDFSMS
DMSFSM1225W Load failed for shared segment DFSMSSEG, reason code = 405
DMS5SH2029I Initialization ends for DFSMS exit routine SMSDFSMS
```

Attention

If you have segment access problems, as shown in message DMSFSM1225W above, discontinue the verification process and fix the access problem before continuing. For information on access problem determination and resolution, see the *z/VM: DFSMS/VM Messages and Codes*.

Step 9: Verify Optional Customization

After customizing DFSMS/VM, you should verify that it works properly. In particular, you need to verify that secondary storage is available if you are using the space management migration capabilities of DFSMS/VM. If you are using both ML1 and ML2, verify that both are available.

Before proceeding with this verification, ensure you have activated a configuration that includes management classes that allow migration. See "Step 4: Verify Space Management Configuration" on page 12.

Note: Because no management class assignment takes place for any user ID beginning with the letters DFSMS, you will need to use a non-DFSMS/VM file space.

To begin verification of optional customization, you will need to:

1. Create a directory for a non-DFSMS/VM user ID in your DFSMS/VM-managed file pool. Assign a management class to this directory.

   Note: You should log onto the file pool server to ensure that no errors are detected during this process.

2. Enter ISMF by typing ISMF. Select the File application (Option 1). See Figure 17 on page 30.

   Note:

   a. Verification is expected to be run from the user ID DFSMS. User ID DFSMS will need to be able to access the directory created in Step 1 above in order to do the verification.

   b. Depending on the type of ACS processing you are running, information on the following panels may vary.

```
DGTSMMD2                    ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 8

SELECT ONE OF THE FOLLOWING:

  0  ISMF PROFILE     - Change ISMF user profile
                       - Control how ISMF logs and traces errors
  1  FILE             - Access and view attributes of SFS files and directories
                       - Create and modify lists of SFS files and directories
                       - Browse, edit, migrate, or recall SFS files
  2  MINIDISK         - Create and modify lists of minidisks
                       - Move and check minidisks
  3  MANAGEMENT CLASS - Define, alter, and display management class attributes
                       - Create and modify lists of management classes
  7  ACS              - Create, translate, validate, test, display, and
                         delete Automatic Class Selection routines
  8  CONFIGURATION    - Display or alter configurations
                       - Define, validate and activate a base configuration
  L  LIST             - Access, print, filter, sort, and erase previously
                         saved lists
  X  EXIT             - End ISMF using console, log and list defaults




USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 17. ISMF Primary Option Menu for storage administrators (DGTSMMD2)

3. The File Application Selection Entry panel appears (see Figure 18 on page 31). If the parent directory is accessed, press ENTER to see the File List panel and verify the management class of the directory

created in step 1. If the parent directory is not accessed, update the *fm/dirid* to specify the directory where your new directory resides before you press ENTER.

```
DGTDSJF1        FILE APPLICATION SELECTION ENTRY PANEL            Page 1 of 3
COMMAND ===>

SELECT SOURCE OF FILE LIST  ===> 2   (1 - saved list, 2 - new list)

1  REUSE A SAVED LIST:

   LIST NAME  ===>

2  GENERATE A NEW LIST FROM CRITERIA BELOW:

   FN FT ===> * *               (Fully or partially specified)

   FM/DIRID ===> *
            ===>
            ===>
                                (File mode letter or directory name)




USE ENTER TO PERFORM SELECTION; USE DOWN COMMAND TO VIEW NEXT SELECTION PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 18. File Application Selection Entry Panel

4. Check the management class by scrolling right to see your directory's management class. See Figure 19 on page 31.

```
DGTLGP81                     FILE LIST PANEL
COMMAND ===>                                          SCROLL ===> HALF
                                                 Entries 1-3 of 3
                                                 Data Columns 10-12 of 17
DIRID: *


ENTER LINE OPERATORS BELOW:


   LINE      FILE     FILE           REC   LAST REF   MANAGEMENT   MC
 OPERATOR    NAME     TYPE     FM   FMT    DATE       CLASS NAME   ASSIGN
 ---(1)----  --(2)--- --(3)--- (4)  (5)   ---(10)--- ---(11)---   -(12)-
            DGTTABL   MACLIB    A    F    1992/09/10  ACNTDEPT    DIR
            PROFILE   EXEC      A    F    1992/09/11  PURCHASE    FILE
            TABLES    MACLIB    A    F    1992/09/10  --NONE--    FILE
            CMND                A    -    ----------  STANDARD    CONFIG
 ---------- ---------- BOTTOM  OF   DATA ------------ ----------- -------




USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 19. File List Panel

Note: Because no migration takes place for any user ID beginning with the letters DFSMS, you will need to access an SFS file space whose directory name does not begin with DFSMS.

If you have enabled the migration capability of DFSMS/VM, you will need to verify that you have installed this correctly. Before proceeding with this verification, ensure you have activated a configuration that includes management classes that allow migration. See "Step 4: Verify Space Management Configuration" on page 12. To ensure proper migration, perform the following:

1. Ensure the user ID is authorized to perform DFSMS/VM functions and has access to the product code.

2. Access the SFS directory selected for this verification as C:

```
      ACCESS VMSYSU:userid  C
```

3. Select an SFS file to migrate. Ensure you select one that is a base file or alias, that is not an empty file, and to which you have write authority. The directory must be a file control directory. Issue the migrate command against the file:

```
      DFSMS MIGRATE filename  filetype  C
```

Note: The management class of the file must have a zero (0) in the secondary storage nonusage field in order for the file to be migrated directly to ML2. Any other value in this field causes the file to be migrated to ML1.

If you will be using ML1 and ML2, ensure that you migrate files to both of them in order to verify this function.

When the command completes, you will receive a reader file containing information regarding the completion of the command. Check the reader files to ensure there were no errors and that the file migrated successfully.

4. Issue the CMS command:

```
LISTFILE fn ft C (SHARE
```

You should see an asterisk (*) immediately after the *BASE* or *ALIAS* keyword on the output. This indicates that the file is migrated.

5. Issue a recall against the file that you migrated:

```
DFSMS RECALL fn ft C
```

When the command completes, you will receive a reader file containing information regarding the completion of the command. Type the CMS command:

```
LISTFILE fn ft C (SHARE
```

The asterisk is no longer displayed.

6. Migrate the file again as previously indicated. Then, XEDIT the file to automatically recall the file. No DFSMS/VM error messages should be issued and you should be put into an XEDIT session on that file. You may perceive a slight delay during the recall.

Step 10: Add Functions to DFSMS/VM

After you have installed DFSMS/VM, you may want to add a function to your existing system. Remember, you could have installed only one DFSMS/VM function or any combination of the three functions (space management, minidisk management, or removable media management). If you installed the desired component but did not implement it, complete any necessary work you may have skipped during your initial install. For example, you may need to create more user IDs or modify the file pool. If you did not install the desired component initially, you will need to reinstall the product. In either case, ensure you perform verification for the new function. Refer to Chapter 2, "Verifying the Installation Process," on page 5

Chapter 4. Authorizing DFSMS/VM Users

This chapter discusses how to authorize access to DFSMS/VM functions, and describes the default authorization process used. You control access to DFSMS/VM functions in several ways:

- Use the RACF/VM program product or an equivalent product which allows you to restrict the use of DFSMS/VM commands.
- Use the simple authorization method of listing users authorized to perform storage administrator functions in an authorization file.
- Use an installation-wide exit to replace or enhance the previous two methods. Refer to Chapter 6, "Using DFSMS/VM Installation-Wide Exits," on page 57
- Or, use another facility for security.

Using RACF/VM

DFSMS/VM uses the FACILITY class of RACF/VM for command level protection. All DFSMS/VM command entities begin with STGADMIN. A single generic profile (STGADMIN.*) can be created to protect all DFSMS/VM commands.

Note: If RACF is shared with z/OS™, or if RACF enhancements are installed, or if another product using the RACROUTE interface is being used, the SMSRACF EXEC may not work. Use the appropriate manuals for the product installed and the tables (Table 2 on page 35 and Table 3 on page 36) to determine how to protect DFSMS/VM functions.

If you need more flexibility, you can also create discrete profiles to specify authorization for each command, user-by-user. For more detailed information on using RACF/VM, see *z/VM: RACF Security Server Security Administrator's Guide*.

RACF/VM Requirements

When using RACF/VM, consider the following requirements and suggestions:

IUCV ALLOW
The RACF/VM virtual machine directory entry needs the IUCV ALLOW statement to permit the DFSMS and RMS master machines to communicate with the RACF/VM virtual machine.

Option MAXCONN
Increase MAXCONN value in the RACF/VM virtual machine CP directory entry by two: one for the DFSMS/VM master and one for the RMS master.

RACF ID
The RACF ID file (RACF SERVMACH) must be accessible by the masters and servers. Place it on a common disk. If the RACF SERVMACH file does not exist, create it. This file contains the name of the RACF/VM server machine which receives the IUCV CONNECT request from machines issuing RACROUTE.

Setting up RACF/VM

There are certain things that must be done for DFSMS/VM to use RACF/VM to provide command-level authorization. The following instructions discuss how to provide command-level authorization.

1. Use the RACF/VM command, ADDGROUP, to create the STGADMIN group.
2. Use the RACF/VM command, CONNECT, to connect all storage administrators to the STGADMIN group.

   Note: Also connect the user ID you are using for the product verification, for example, DFSMS.
3. Run the sample SMSRACF EXEC which you have customized for your installation.

4. Because the authorization file is accessed if RACF/VM fails for any reason, ensure that this file is empty or that it contains only one authorized user ID so that no user IDs are inadvertently given access to DFSMS/VM.

The SMSRACF EXEC that defines the necessary profiles is shipped with DFSMS/VM, and is located on the production disk. This exec must run in the MAINT virtual machine or a virtual machine with RACF/VM "special" authority. It may need to be customized for your installation. At minimum, if you have changed the default names of the masters and servers, this exec must be modified; or, if you have file pools other than VMSYSU being DFSMS/VM-managed, it will need to be customized.

The following provides information for defining the sample EXEC (SMSRACF EXEC):

1. Set up RACF/VM to allow the DFSMS/VM virtual machines to use the RACROUTE interface. This is accomplished by giving those machines (for example, SMSMASTR and RMSMASTR) UPDATE authority to the ICHCONN profile in the FACILITY class.

   The following series of RACF/VM commands can be used to setup RACF/VM for the default SMSMASTR and RMSMASTR user IDs.

   ```
   RAC RDEFINE FACILITY ICHCONN UACC(NONE)
   RAC PERMIT ICHCONN CLASS(FACILITY) ID(SMSMASTR) ACCESS(UPDATE)
   RAC PERMIT ICHCONN CLASS(FACILITY) ID(RMSMASTR) ACCESS(UPDATE)
   RAC SETROPTS CLASSACT(FACILITY)
   ```

   If you are not using the default user IDs, you must alter the user IDs accordingly.

   Note: During DFSMS/VM initialization, all DFSMS/VM virtual machines will attempt to connect to RACF/VM using the RPIUCMS INIT command. If this command fails or the connection to RACF/VM is lost for any reason, all machines use the authorization file for as long as DFSMS/VM remains active. None of the machines attempt to use RACF/VM again until DFSMS/VM is stopped and restarted.

2. After RACF/VM has been set up, you must define the profiles to protect the individual commands. DFSMS/VM fails authorization requests if the FACILITY class is active, since there is no profile (either generic or discrete) that protects a command. It is recommended that you establish a generic profile to protect all commands, and then define discrete profiles to override that authority.

For example, suppose that:

- You have a group of storage administrators and want to give them all equal authority to DFSMS/VM commands.

- You want to give general users authority only to use the DFSMS/VM MIGRATE and RECALL commands.

To do this, you would first put all storage administrators in the same RACF/VM group, perhaps called STGADMIN. Then you would enter the following series of commands to establish the RACF/VM profiles you need:

```
RAC SETROPTS GRPLIST GENERIC(FACILITY) GENCMD(FACILITY)    1
RAC RDEFINE FACILITY STGADMIN.* UACC(NONE)      2
RAC PERMIT STGADMIN.* CLASS(FACILITY) ID(STGADMIN) ACCESS(READ) 3
RAC RDEFINE FACILITY STGADMIN.MIGRATE UACC(READ)       4
RAC RDEFINE FACILITY STGADMIN.RECALL UACC(READ)
RAC RDEFINE FACILITY STGADMIN.APPC UACC(NONE)       5
RAC PERMIT STGADMIN.APPC CLASS(FACILITY) ID(VMSERVU) ACCESS(READ)
RAC PERMIT STGADMIN.APPC CLASS(FACILITY) ID(VMSERVS) ACCESS(READ)
```

In the first command **1**, you enable generic profiles for the FACILITY class. Next in **2**, you define a new generic FACILITY class profile that immediately prevents *anyone* from using *any* DFSMS/VM commands and functions. With **3**, you allow all users in the RACF/VM group STGADMIN to perform all DFSMS/VM commands and functions.

The next command **4** and the one following it, permits all users to use the MIGRATE and RECALL commands.

The last three commands **5**, allow SFS server virtual machines to connect to DFSMS/VM virtual machines. In this example, file pool servers VMSERVU and VMSERVS can connect to DFSMS/VM. You can either repeat the PERMIT command for each file pool server associated with a file pool to be managed

by DFSMS/VM, or define a RACF/VM group that contains the user IDs of all SFS file pool servers and issue a PERMIT command giving that group READ access to the STGADMIN.APPC profile.

Note: Only SFS file pool servers should be granted access to STGADMIN.APPC. If any other user is permitted access to that profile, data integrity may be compromised.

This example shows one way that you can use a combination of generic and discrete profiles to set up RACF/VM as you require. Additional individual PERMIT commands can be issued to define RACF/VM entities. , which shows the RACF/VM entities for DFSMS/VM, supplies the information you need.

The DISCARD (DFSMS, ISMF, and DFSMSRM), QUERY REQUEST (DFSMS, ISMF, and DFSMSRM), and REPORT (DFSMS) commands have both an owner (OWN) and an any user (ANY) discrete RACF entity associated with them. When using the DISCARD and QUERY commands, users solely authorized to read the OWN entity can query and discard requests that only they have issued. When using the REPORT command, users solely authorized to read the OWN entity can obtain a report of only their own file space.

The DISCARD, QUERY REQUEST, and the REPORT SPACEMANAGEMENT FILESPACE commands check for authorization by any user (ANY). If that authorization fails, the command checks for OWN authorization. Therefore, a user ID not authorized for the command has two access failures: one for the ANY entity and one for the OWN entity.

Note: QUERY REQUEST, DISCARD and REPORT FILESPACE commands check for ANY authorization before checking for OWN authorization. Any ID with only OWN authority records a failure for the ANY authorization. To prevent access failure notification each time a user solely authorized to OWN issues a command, define a specific ANY authorization with the AUDIT(NONE) parameter and without the NOTIFY parameter.

Table 2. RACF/VM Entities for DFSMS/VM

Command	RACF/VM Entity Checked
DFSMS/ISMF ACTIVATE	STGADMIN.ACTIVATE
DFSMS/ISMF ALTER	STGADMIN.ALTER
APPC	STGADMIN.APPC (only for SFS file pool servers)
ISMF BUILD LIST	STGADMIN.MINIDISK.BUILDLST
ISMF CHECK	STGADMIN.MINIDISK.CHECK
DFSMS CONVERT	STGADMIN.CONVERT
DFSMS DELETE	STGADMIN.DELETE
DFSMS/ISMF DISCARD (user requestid)	STGADMIN.DISCARD.OWN
DFSMS/ISMF DISCARD (any requestid)	STGADMIN.DISCARD.ANY
DFSMS MANAGE	STGADMIN.MANAGE
DFSMS MIGRATE	STGADMIN.MIGRATE
DFSMS/ISMF MOVE	STGADMIN.MINIDISK.MOVE
DFSMS/ISMF QUERY REQuests (* or own)	STGADMIN.QUERY.REQUEST.OWN
DFSMS/ISMF QUERY REQuests (ALL, FOR, or other user's)	STGADMIN.QUERY.REQUEST.ANY
DFSMS QUERY STATUS	STGADMIN.QUERY.STATUS
DFSMS RECALL	STGADMIN.RECALL
DFSMS REPORT SPACEMANAGEMENT FILESPACE (user requestid)	STGADMIN.REPORT.OWN
DFSMS REPORT SPACEMANAGEMENT FILESPACE (any requestid)	STGADMIN.REPORT.ANY
DFSMS REPORT SPACEMANAGEMENT STORGROUP	STGADMIN.REPORT.ANY

Command	RACF/VM Entity Checked
DFSMS STOP	STGADMIN.STOP

Using RACF/VM to Authorize RMS Users

Using procedures similar to those for other DFSMS/VM virtual machines, the RMS master can be set up to use the RACROUTE interface, and appropriate machines such as a tape management system (TMS) librarian can be connected to the STGADMIN group, or to an alternate user-defined group if desired. RACF/VM provides authorization by userid for command-level entities. The DFSMSRM command entities begin with STGADMIN.RM. A single generic profile (STGADMIN.RM.*) can be created to protect all DFSMSRM commands. Table 3 on page 36 describes the additional RACF/VM entities for DFSMS/VM. For more information on RMS authorization, refer to *z/VM: DFSMS/VM Removable Media Services*.

Table 3. RACF/VM Entities for IBM 3495 Tape Library Dataserver Support Functions

Command	RACF/VM Entity Checked
DFSMSRM DEMOUNT	STGADMIN.RM.DEMOUNT
DFSMSRM DISCARD (user requestid)	STGADMIN.RM.DISCARD.OWN
DFSMSRM DISCARD (any requestid)	STGADMIN.RM.DISCARD.ANY
DFSMSRM MOUNT	STGADMIN.RM.MOUNT
DFSMSRM QUERY LIBrary	STGADMIN.RM.QUERY.LIB
DFSMSRM QUERY REQuests (* or own)	STGADMIN.RM.QUERY.REQ.OWN
DFSMSRM QUERY REQuests (ALL, FOR, or other user's)	STGADMIN.RM.QUERY.REQ.ANY
DFSMSRM RESET DEVCAT	STGADMIN.RM.RESET.DEV
DFSMSRM SET DEVCAT	STGADMIN.RM.SET.DEV
DFSMSRM SET VOLCAT	STGADMIN.RM.SET.VOL
DFSMSRM SET VOLCAT BULK	STGADMIN.RM.SET.VOL.B
DFSMSRM STOP	STGADMIN.RM.STOP

Using the Authorization File

If RACF/VM release 1.9 or later, or its equivalent, is not installed or if there is an error communicating with RACF/VM, DFSMS/VM uses the authorization file. When using the authorization file for authority checking, DFSMS/VM differentiates between commands that can be issued by storage administrators (whose user IDs must be in the authorization file) and commands that can be issued by general users (in which case no checking of the authorization file occurs).

Because the RMS master is designed to work with a tape management system (TMS), the standard designation for all DFSMSRM commands is that they are issued by authorized users and are not available to end-users.

Table 4 on page 37 indicates which space management and minidisk management commands are general user and which are storage administrator commands.

If a command appears in both the general user and storage administrator columns, end users can only successfully issue DFSMS operational commands (such as QUERY and DISCARD) on requests that they have issued and can only reference data (using the MIGRATE, RECALL, and REPORT FILESPACE commands, for example) on files that they have sufficient SFS authority to do so.

Table 4. General User and Storage Administrator Commands

Command	General User	Storage Administrator
DFSMS/ISMF ACTIVATE		X
DFSMS/ISMF ALTER		X
ISMF BUILD LIST		X
ISMF CHECK		X
DFSMS CONVERT		X
DFSMS DELETE		X
DFSMS/ISMF DISCARD	X	X
DFSMS MANAGE		X
DFSMS MIGRATE	X	X
DFSMS/ISMF MOVE		X
DFSMS QUERY DFSMSLEVEL	X	X
DFSMS/ISMF QUERY REQUEST	X	X
DFSMS QUERY STATUS		X
DFSMS RECALL	X	X
DFSMS REPORT SPACEMANAGEMENT FILESPACE	X	X
DFSMS REPORT SPACEMANAGEMENT STORGROUP		X
DFSMS STOP SMS		X

Note: Care should be taken in granting authority to DFSMS/VM storage administrator commands. These commands should be restricted to storage administrators because their use can affect SFS data for which the issuer does not have authority. Those with authority to use the DFSMS/VM ALTER command, for example, can change the management class of a file or directory even if they do not have SFS authority for that file or directory.

Understanding the File Structure and Use

There are several requirements needed to define the DFSMS/VM authorization file. The following sections detail the requirements for the DFSMS/VM authorization file attributes and the structure of its contents.

Attributes of DFSMS/VM Authorization File

The DFSMS/VM authorization file is a CMS file with the following required attributes:

File name, File type, and File Placement

The required file name is DGTVAUTH, with the file type DATA. You must store this file, with read access by the master and server virtual machines, in the SFS directory VMSYS:DFSMS.CONTROL.

File Record Requirements

The record format must be fixed. The maximum record length allowed is 240.

File Updates

To implement changes to this file, the master and server virtual machines must be shut down and restarted. Users of ISMF storage administrator functions must exit and then reenter ISMF.

Structure of DFSMS/VM Authorization File

To use the file, enter the names of any user who needs storage administration authority in the authorization file as follows:

```
        USERID              * comment text
```

where, *USERID* is the user ID of a valid virtual machine that gets storage administration authority, and *
indicates that the following text is a comment.

For example, to authorize a user to use DFSMS/VM storage administration commands, the entry might be as follows:

```
        USERID04            * System Administrator for DFSMS/VM
```

Comments

Comments begin with the comment indicator, an asterisk (*). Comments cannot span lines. A comment stops at the end of a line (logical record length). To continue a comment on the next line, start the next line with the comment indicator.

Comments can start anywhere on the line. However, once the comment indicator is encountered, the remainder of the line is interpreted as a comment.

Blank Lines

Blank lines in the file are ignored.

Lowercase/Mixed-Case/Uppercase

Because all input is converted to upper case, any combination of case is acceptable for comments and user IDs.

Contents of the Authorization File

When you install DFSMS/VM, the authorization file exists and contains an entry for DFSMS.

File pools being DFSMS/VM-managed need to be included in the authorization file, as they need storage administration authority.

Using Another Facility for Security

If you choose a security facility that does not use the RACROUTE interface, or if you choose to devise your own method, you must customize the authorization process by replacing the CSL routine FSMVAUTH. See Chapter 6, "Using DFSMS/VM Installation-Wide Exits," on page 57 for more details.

You can use a security facility other than RACF/VM. If you choose a security facility that uses the RACROUTE interface, as RACF/VM does, you simply install that program instead of RACF/VM. DFSMS/VM uses whatever program answers to the RACROUTE interface. (For more information on this interface, see *z/VM: Security Server RACROUTE Macro Reference*).

Migrating from DFSMS/VM Function Level 210

The DGTQAUTH EXEC, an installation-wide exit available in DFSMS/VM Function Level 210, is not available with DFSMS/VM Function Level 221. If your system modified DGTQAUTH to authorize users of DFSMS/VM Function Level 210, you can continue to use DGTQAUTH by replacing the FSMVAUTH CSL routine to invoke it. See "Space Management Installation-Wide Exits" on page 58 for more information on the FSMVAUTH CSL routine.

Using the Authorization Process

The actual method of authorization varies from installation to installation depending upon whether you have changed the FSMVAUTH CSL routine and whether you use RACF/VM or a product using the RACROUTE interface.

Note: If you use the FSMVAUTH CSL routine shipped with DFSMS/VM, a return code of 4 will always be issued.

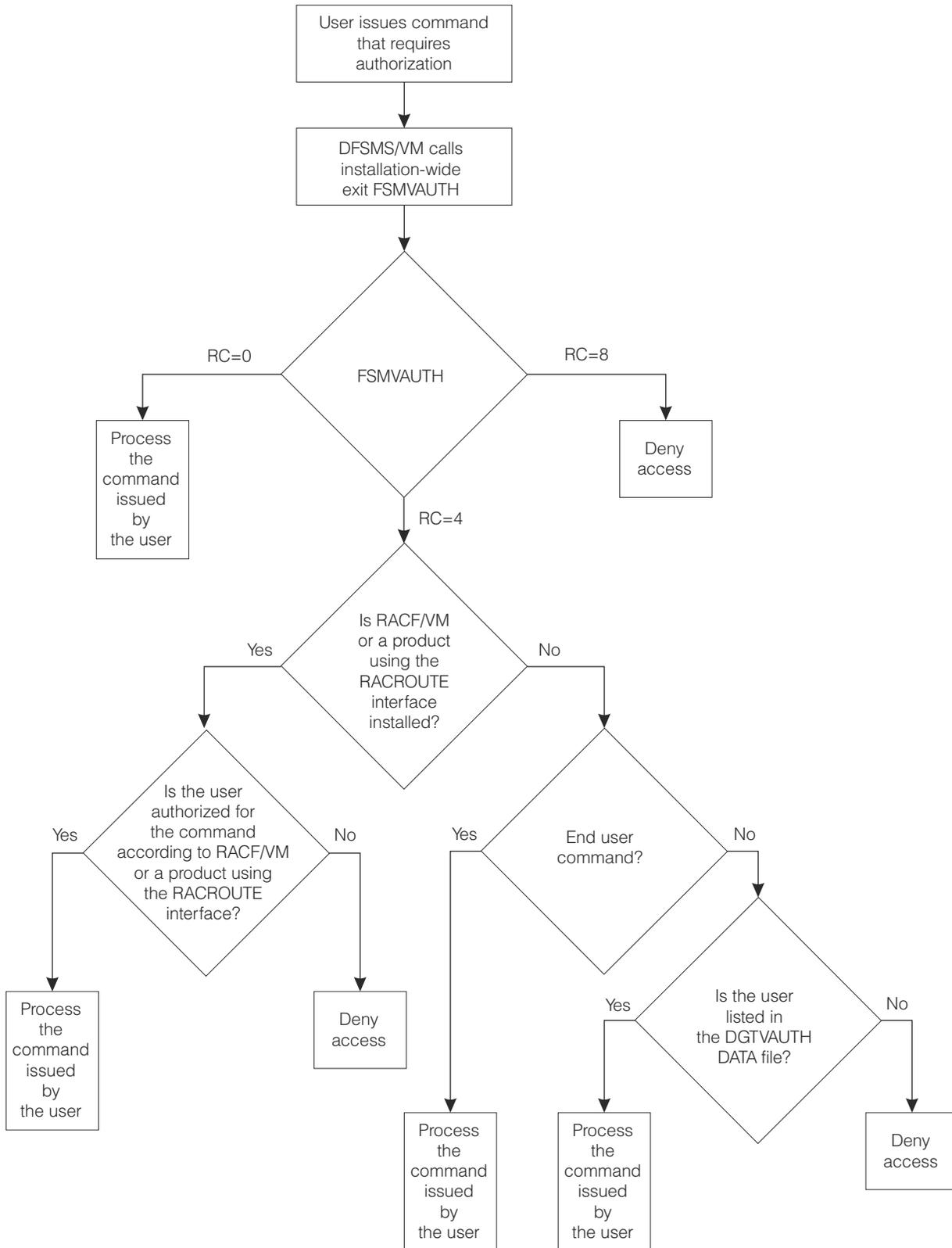shows the steps performed by DFSMS/VM to determine authority for functions.

Figure 20. Authorization Process for DFSMS/VM Commands

Chapter 5. Customizing the DFSMS/VM Control File

This chapter provides detailed information on customizing DFSMS/VM using the DFSMS/VM control file. The following topics are covered:

- File attributes and format
- Use of the DFSMS/VM defaults
- Detailed description of file keywords.

The DFSMS/VM control file specifies the names of the virtual machines in the DFSMS/VM configuration, as well as other parameters that control DFSMS/VM. You can modify this file to suit the needs of your installation.

Note: No virtual machine name can be the same as any other virtual machine name specified in the control file.

DFSMS/VM ships a sample control file specifying DFSMS/VM defaults. If your installation uses the defaults, DFSMS/VM's space and minidisk management machines can be brought up using this file.

Understanding File Structure and Use

There are several requirements needed to define the DFSMS/VM control file. The following subsections detail the requirements for the DFSMS/VM control file attributes and the structure of its contents.

Attributes of the DFSMS/VM Control File

The DFSMS/VM control file is a CMS file with the following required attributes:

File Name, File Type, and File Placement

The required file name is DGTVCNTL, with the file type DATA. You must store this file, with read access by the master and server virtual machines and ISMF storage administrators, in the SFS directory VMSYS:DFSMS.CONTROL.

File Record Requirements

The record format must be fixed. The maximum record length allowed is 240.

File Updates

To implement changes to this file, the master and server virtual machines must be shut down and restarted. Users of ISMF storage administrator functions must exit and then reenter ISMF.

Structure of DFSMS/VM Control File Content

Each entry in the DFSMS/VM control file is a keyword-value pair (a keyword followed by a corresponding value). However, some entries may have two or more values after the keyword. For example,

```
        DFSMS_MASTER_VM        SMSMASTR
```

Entries must conform to the following syntax rules:

- At least one blank must separate each keyword from its corresponding value.
- A keyword must precede a value.
- A keyword and its corresponding value must appear on the same line.

- A default value (if one exists) is used if the keyword-value pair is not specified in the control file. Some keywords do not have a default. See "Modifying the DFSMS/VM Control File" on page 43 for detailed explanations of all keyword-value pairs.
- Multiple keyword-value pairs on a single line are NOT permitted.

You should also be aware of the following points:

Ordering

Keyword-value pairs can appear in any order, subject to the entry rules previously defined.

Comments

Comments begin with an asterisk (*). Comments cannot span lines. That is, a comment stops at the end of the line (logical record length). To continue a comment on the next line, start the next line with an asterisk (*).

Comments can start anywhere on the line. They can start directly after a keyword value. However, once the comment indicator is encountered, the remainder of the line is interpreted as a comment.

Blank Lines

Blank lines in the file are ignored.

Lowercase/Mixed-Case/Uppercase

Any combination of case is acceptable for comments, keywords, and values. That is, these fields are not case-sensitive.

Multiple Use of a Keyword

If you specify a keyword more than one time in the DFSMS/VM control file, the keyword is flagged as an error and DFSMS/VM terminates. The only exceptions to this general rule on multiple specification of keywords are:

- MINIDISK_SERVER_VM
- DFSMS_SERVER_VM
- RM_AUTO_LIBRARY
- RM_FOREIGN_SERVER_VM
- RM_MANUAL_LIBRARY

You can use these keywords to specify the following:

- Multiple server virtual machines
- Multiple IBM 3494 tape libraries
- Multiple RMS manual libraries

Using the DFSMS/VM Control File Defaults

A default value (if one exists) is used if the keyword-value pair is not specified in the control file. However, if a default value does not exist for a keyword and the keyword requires a value, then the value must be specified in the control file. Not all control file entries have default values. Default values are explained as part of the keyword description.

Modifying the DFSMS/VM Control File

To modify the DFSMS/VM control file, you need detailed descriptions of the keywords, their functions, and the acceptable values associated with the keywords. This section includes the following information on keywords, grouped according to their general function type:

- Defining keywords required by all DFSMS functions (minidisk management, space management, and removable media services)
- Defining minidisk and space management keywords
- Defining minidisk management control file keywords
- Defining space management control file keywords
- Defining removable media services control file keywords.

Control File Keywords Required by All Functions

Both the WORK_DIRECTORY and GLOBAL_RESOURCE_ID control file keywords are required by minidisk management, space management and removable media services.

WORK_DIRECTORY

This keyword specifies a directory to be used as a work directory for the DFSMS virtual machines. The work directory must be in a file space whose first five characters are DFSMS. This file space must be enrolled in a file pool; if it is not, DFSMS/VM will not start. Make sure the file pool in which the work directory resides is running before DFSMS/VM is started. No more than seven subdirectories may be specified on the WORK_DIRECTORY keyword. This is a required keyword-value pair; if omitted, DFSMS does not start.

All DFSMS virtual machines must have administrator authority to the file pool containing the work directory. The work directory must not be a directory control directory.

Default value

There is no default value for this keyword-value pair; however, the sample control file uses VMSYSU:DFSMS.WORK as its work directory.

```
    WORK_DIRECTORY    filepool:DFSMSxxx.subdirectory    * Work directory
```

GLOBAL_RESOURCE_ID

This keyword specifies the name of the global advanced program-to-program communications (APPC) resource by which this DFSMS/VM system is known. This is a required keyword-value pair. If omitted, DFSMS/VM does not start. This resource ID must be unique within a transparent services access facility (TSAF) collection or DFSMS/VM will not start. (See *z/VM: Connectivity* for more information on resource IDs.)

The value specified is used in the directory naming structure of secondary storage and if you later change this parameter, you will have to rename the associated directory for ML1.

Default value

There is no default value for this keyword-value pair; however, the sample control file uses DFSMS001 as its global APPC resource ID.

```
    GLOBAL_RESOURCE_ID    appc_resource_id    * Global APPC resource
```

Control File Keywords Required by Both Space and Minidisk Management

The following section describes keywords used to control both space and minidisk management.

DFSMS_ACCOUNTING

This keyword specifies whether DFSMS/VM should perform accounting processing or not. DFSMS/VM accounting values are only valid on DFSMS/VM systems running on first level machines.

Valid values for this keyword value are yes (Y) and no (N). This keyword value is optional. If this keyword value is omitted, DFSMS/VM will not perform accounting processing.

See "Step 3: Enable the DFSMS/VM Accounting Function" on page 20 for further requirements.

Default value

The default value for this keyword is N.

```
DFSMS_ACCOUNTING N
```

DFSMS_LOG_TO_CONSOLE

This keyword specifies the severity level of messages that are to be written to the console of the DFSMS master and server virtual machines. The value of this keyword-value pair specified is a one-character field, which may be one of the following:

0
No messages are to be logged

1
Severe messages are to be logged

2
Severe and error messages are to be logged

3
Severe, error, and warning messages are to be logged

4
Severe, error, warning, and informational messages are to be logged.

Message level 0 cannot be specified for both the console and log file destinations. Because DFSMS/VM must be able to log severe messages somewhere, one of these keywords must specify a message level of 1 or greater.

Note: To assist in problem diagnosis and problem resolution, it is recommended that either DFSMS_LOG_TO_CONSOLE or DFSMS_LOG_TO_FILE always be set to 4.

Default value

The default value of the keyword-value pair is 3.

```
DFSMS_LOG_TO_CONSOLE 3                   * Messages logged to console
```

DFSMS_LOG_TO_FILE

This keyword specifies the severity level of messages that are to be written to the log file of the DFSMS master and server virtual machines and the directory in which the log files are stored. The first value specified with this keyword is a one-character field, which may be one of the following:

0
No messages are to be logged

1
Severe messages are to be logged

2
Severe and error messages are to be logged

3
Severe, error, and warning messages are to be logged

4
> Severe, error, warning, and informational messages are to be logged.

The second value is a one- to 153-character directory name in which the log files will be created. The directory must not be a directory control directory. If the first value specified is 0, then the second value can be omitted.

Message level 0 cannot be specified for both the console and log file destinations. Because DFSMS/VM must be able to log severe messages somewhere, one of these keywords must specify a message level of 1 or greater.

Note: To browse current log files, you need to XEDIT the log files with the NOLOCK option because DFSMS/VM has locked the current files.

The top level directory that you specify should not have *DFSMS* as its first five characters. (A directory that begins with *DFSMS* cannot be managed by DFSMS/VM. It is recommended that you allow DFSMS/VM to manage the log files.) Because log files can grow rapidly, do not put log files in the same file space as the work directory. Doing so can use up required space in the work directory and DFSMS/VM will be unable to successfully complete requests.

The file name of a log file is the name of the master or server machine that created it. The file type is the date that the log file was created. A new log file is started with the first request after midnight of each day and is the active file for that day.

The following are examples of log file IDs:

```
SMSMASTR 20020331
SMSSRV01 20020330
DGTSRV01 20020401
```

Note: When testing your installation of DFSMS/VM, use at least level 3 for DFSMS_LOG_TO_FILE or DFSMS_LOG_TO_CONSOLE to fully capture all warning and error messages which might be encountered while verifying DFSMS.

Default value

The default value of this keyword is 0. If you want to log to a file, you must specify both message level and directory ID. If the log to file directory ID does not exist, DFSMS will create it for you. Even so, the filespace must already be enrolled.

```
DFSMS_LOG_TO_FILE     0 dirid              * Messages logged to file
```

Attention

DFSMS always keeps its log files open, so a full backup cannot be performed with the storage group utility, since the utility allows no files to be opened for WRITE.

If log files are placed in a storage group with other data, the DFSMS system must be down to issue FILEPOOL BACKUP against that storage group. You may want to consider putting the log files in their own storage group so that your FILEPOOL BACKUP is unaffected by DFSMS being up or down.

DFSMS_MASTER_VM

This keyword specifies the DFSMS virtual machine that accepts and dispatches all space management and minidisk management requests. A one- to eight-character user ID follows the keyword. The virtual machine name must not be the same as the name specified for any other virtual machine in the control file. This is a required keyword-value pair. If omitted, DFSMS/VM space and minidisk management do not start. RMS, however, will be able to start. If you specify this keyword, you then must specify at least one DFSMS or minidisk server or DFSMS master will not start.

Default Value

The keyword-value pair must be specified for minidisk management or space management. Be sure to define this virtual machine in the CP directory. There is no default value for this keyword; however, the sample control file has SMSMASTR as the virtual machine's name.

```
        DFSMS_MASTER_VM              SMSMASTR
```

Control File Keywords Required by Space Management

The following section describes keywords used to control space management.

Note: To run space management, you must specify the DFSMS master, server, work directory, active configuration file, and global resource ID in the control file.

To perform space management functions you must specify the DFSMS_SERVER_VM keyword in the control file. If you submit a space management request without DFSMS/VM servers defined in the control file, those requests are accepted but will never be processed. You can discard those requests by issuing either the DFSMS DISCARD or DFSMS STOP SMS (IMMEDIATE command.

ACTIVE_CONFIG_FILE_ID

This keyword specifies the name of the active configuration file that DFSMS/VM uses when it is initialized. This file is located in directory VMSYS:DFSMS.ACTIVECONFIG. This is a required keyword-value pair.

Be careful when changing this parameter. If this name is changed, you will not initialize with the configuration you last activated. You will need to reactivate your configuration. In addition, if you change this name to a name you have used in the past, there may be an old configuration that will now be used when DFSMS/VM initializes.

Default Value

The keyword-value pair must be specified for space management. There is no default value for this keyword-value pair; however, the sample control file uses ACTIVE CONFIG as its configuration file.

```
        ACTIVE_CONFIG_FILE_ID     ACTIVE CONFIG  * Active configuration file
```

DFSMS_SERVER_VM

This keyword specifies each DFSMS server virtual machine that you define for DFSMS/VM. Each keyword-value pair identifies a one- to eight-character user ID and a specified number of server tasks. Each server task can process various space management commands dispatched by the DFSMS master virtual machine.

You can specify the DFSMS_SERVER_VM keyword-value pair any number of times, depending on the number of server virtual machines you want. However, a virtual machine user ID value can appear only once in the control file; you cannot use duplicate server virtual machine user ID values.

At least one server machine must be specified for space management.

Following the virtual machine's ID is a number you specify indicating the number of server tasks to run concurrently on this virtual machine. This number must be greater than or equal to one. This number affects the MAXCONN you specify in your server directory entries. Refer to "Space Management Server Virtual Machine" on page 83 for details of specifying the MAXCONN of the server virtual machines.

The DFSMS master virtual machine dispatches work to available server virtual machines. The number of active server virtual machines should be directly proportional to the amount of work performed in parallel.

Default value

At least one server virtual machine must be specified. Be sure to define these virtual machines in the CP directory. There is no default value for the user ID. If the number of server tasks is not specified, a default

of 4 is used. The sample control file has three servers with 8 tasks each defined: SMSSRV01, SMSSRV02, SMSSRV03.

```
        DFSMS_SERVER_VM      SMSSRV01  8   * User ID and number of server tasks
        DFSMS_SERVER_VM      SMSSRV02  8   * User ID and number of server tasks
        DFSMS_SERVER_VM      SMSSRV03  8   * User ID and number of server tasks
```

FULLY_QUALIFIED_LUNAME

This keyword specifies a network addressable unit that is used for routing within the SNA network (systems network architecture) and is unique within the network. The *netid.luname* should be the same as that used for the LUNAME parameter in the start-up parameter file (VMSERVR DMSPARMS) for the coordinated resource recovery (CRR) server. The LU name is the APPC/VM VTAM Support (AVS) gateway name. If you have more than one AVS gateway, then pick only one AVS gateway name. (See *z/VM: CMS File Pool Planning, Administration, and Operation* for more information on LU names.)

The parameters must be separated by a period.

Be careful when selecting or changing this parameter for space management functions. The value specified is used in the directory naming structure of secondary storage and if you later change this parameter, you will have to rename the associated directory for ML1.

Default value

If this keyword-value pair is omitted, NONETWRK.NOLUNAME is used as the default. If you are not in a TSAF collection, or not using an SNA, use the default NONETWRK.NOLUNAME.

```
        FULLY_QUALIFIED_LUNAME NONETWRK.NOLUNAME        * Identifies the gateway
```

MIGRATION_LEVEL_1

This keyword specifies the file pool and file space that is the location of ML1, which is used to store migrated files. The first five characters of the *userid* specified must be DFSMS. Only one SFS file space is supported for MIGRATION_LEVEL_1 storage. The file pool may be local or remote. Having a remote MIGRATION_LEVEL_1 is an option for a system. However, recalls will take longer and data may be unavailable if the remote systems or data links go down.

If this keyword-value pair is omitted, DFSMS performs file expiration, but not file migration or recall.

This top level directory must be created and enrolled with the appropriate amount of SFS space to hold ML1 data. It is recommended that ML1 storage be defined in a file pool that is not being managed by DFSMS/VM. Refer to "Step 4: Define a File Space for Migration Level 1 (ML1)" on page 20 for more information on installing ML1 storage.

If ML1 is defined in a file pool that is being managed by DFSMS/VM, then the ML1 file space must be placed in a unique storage group.

Default value:

There is no default value for this keyword. If this keyword is specified, you must also specify at least one DFSMS_SERVER_VM keyword.

```
        MIGRATION_LEVEL_1 SFS     filepool:DFSMSxxx.   * Migrated files repository
```

MIGRATION_LEVEL_2

This keyword specifies the ML2 server which will contain the migrated data on ML2.

Important

To use ML2 space management capabilities, you must be utilizing the Language Environment® support that is included with z/VM.

The DFDSM™ and IUCV parameters are required on this keyword. The *identifier* is the z/VM user ID for the TSM server.

This keyword-value pair is not required in the control file. However, if it is specified, the MIGRATION_LEVEL_1 keyword must also be specified; otherwise DFSMS initialization will fail.

If this keyword-value pair is omitted, DFSMS will not perform recall or migration to ML2.

The password parameter for the MIGRATION_LEVEL_2 keyword is optional if you decide to set DFDSM authentication off. However, it is recommended that TSM authentication be set on and that a password be specified so that ML2 data can be protected. The password specified here must match the password specified when the DFSMS node is registered to TSM. If these passwords do not match, any attempt to access ML2 will fail. See for more details.

Default value:

There is no default value for this keyword-value pair.

```
MIGRATION_LEVEL_2 DFDSM IUCV   identifier   password
```

STORGRP_HIGH_THRESHOLD

This keyword specifies the high threshold value used during space management initiated by the DFSMS MANAGE command with the THRESHOLD option. Files in an SFS storage group are migrated and erased only if the space used by the storage group is greater than or equal to the percentage specified. The percentage is a value between 1 and 99. The high threshold value must be greater than the low threshold value. Refer to the *z/VM: DFSMS/VM Storage Administration* for more information on thresholds.

Default value

If this keyword-value pair is omitted, 74 is used as the default.

```
STORGRP_HIGH_THRESHOLD    74   * High threshold utilization percentage
```

STORGRP_LOW_THRESHOLD

This keyword specifies the low threshold value used during space management initiated by the DFSMS MANAGE command with THRESHOLD option. Files in an SFS storage group are migrated until the storage group reaches the space usage specified by the low threshold value. If 0 is specified for the percentage, all files in the SFS storage group eligible for migration are migrated. The percentage is a value between 0 and 98. The low threshold value must be less than the high threshold value. Refer to the *z/VM: DFSMS/VM Storage Administration* for more information on thresholds.

Default value

If this keyword-value pair is omitted, 40 is used as the default.

```
STORGRP_LOW_THRESHOLD    40   * Lower threshold utilization percentage
```

Control File Keywords Required by Minidisk Management

The following section describes keywords used to control minidisk management.

Note: To run minidisk management, you must specify the DFSMS master, minidisk server, work directory, and global resource ID in the control file.

DFSMS_OWNER_VM

This keyword specifies the DFSMS/VM owner virtual machine. A one- to eight-character user ID follows the keyword. The virtual machine name must not be the same as the name specified for any other virtual machine.

This keyword is used to protect any minidisk or minidisks that may have DFSMS/VM product code resident on them. If an unauthorized user issues a MOVE command, for example, for any minidisk that contains the DFSMS/VM product code, those users will receive a message indicating they are not authorized to issue such a request.

Default Value

If you do not use the keyword-value pair, DFSMS (the default value) is used.

```
    DFSMS_OWNER_VM          DFSMS          * User ID of owner
```

DIRECTORY_INTERFACE

This keyword specifies the directory maintenance virtual machine interface, defined by a one- to four-character value designating the method of communications. Valid message type values are SMSG, MSG, and WNG. For WNG, the minidisk server needs privilege class A, B, or C.

Requests to the directory maintenance virtual machine for user directory services come from the DGTQDIRM EXEC using the specified interface. DGTQDIRM EXEC can be tailored to meet local communication requirements; this keyword-value pair is provided as a convenience rather than a necessity.

Default Value

SMSG is the default and is assumed if this keyword-value pair is omitted.

```
    DIRECTORY_INTERFACE  SMSG            * Method of communication with
                                         * directory maintenance VM
```

DIRECTORY_TIMEOUT

This keyword specifies the directory maintenance interface timeout. A one- to four-digit value designates the number of seconds before a timeout occurs while waiting for a response to a directory maintenance request.

Default Value

The default value is 120 seconds and is used if you omit this keyword-value pair.

```
    DIRECTORY_TIMEOUT    120             * Number of seconds to wait for a
                                         * directory maintenance response
```

DIRECTORY_VM

This keyword specifies the user ID of the directory maintenance virtual machine. A one- to eight-character user ID identifies the directory maintenance virtual machine.

All requests for user directory services go to this virtual machine through the directory exit, DGTQDIRM EXEC, described in . DGTQDIRM EXEC must be tailored to meet local needs if the default is not used.

Default Value

When you do not include a keyword-value pair, the default value, DIRMAINT, is used.

```
    DIRECTORY_VM          DIRMAINT       * User ID of directory maintenance
```

DIRECTORY_VOLSER_RESET

This keyword specifies the volume serial number (volser). The directory reset volser is a one- to six-character value designating a volser (preferably fictitious).

The server virtual machine resets all of its required but internally used minidisk definitions to reside on this volume at the conclusion of each minidisk job. This is done so that overlapping extents do not occur unnecessarily in the directory.

Default Value

The default value of $$$$$$ is assumed if this keyword-value pair is omitted.

```
DIRECTORY_VOLSER_RESET  $$$$$$      * Not-in-use MDISK volume serial
```

MINIDISK_SERVER_VM

This keyword specifies each minidisk server virtual machine that you define for DFSMS/VM. Each keyword-value pair identifies a one- to eight-character user ID.

You can specify the MINIDISK_SERVER_VM keyword-value pair any number of times, depending on the number of server virtual machines you want. However, a virtual machine user ID value can appear only once in the control file; you cannot use duplicate server virtual machine user ID values.

The DFSMS master virtual machine dispatches work to available server virtual machines. The number of active server virtual machines should be directly proportional to the amount of work performed in parallel.

Default Values

If minidisk management work needs to be performed, at least one minidisk server must be specified. Be sure to define these virtual machines in the CP directory. There is no default value for the user ID. The sample control file has three servers defined: DGTSRV01, DGTSRV02, and DGTSRV03.

```
MINIDISK_SERVER_VM       DGTSRV01    * User ID of server virtual machine 1
MINIDISK_SERVER_VM       DGTSRV02    * User ID of server virtual machine 2
MINIDISK_SERVER_VM       DGTSRV03    * User ID of server virtual machine 3
```

MINIDISK_IPLTEXT_LIMIT

This keyword specifies whether or not to prevent a minidisk server virtual machine from moving a minidisk where possible IPL text is detected.

There are two acceptable values. N prevents the movement of such minidisks, and Y permits a minidisk management move of such minidisks without moving the IPL text; that is, it moves all text *except* the IPL text. You must also specify Y as the value for MINIDISK_MISMATCH_LIMIT if you want to move IPL minidisks.

If you specify N as the value for the keyword, you *cannot* override this restriction by means of the Move Entry Panel data entry field, MOVE IF IPL TEXT PRESENT. Also, no attempt is made to move minidisks with IPL text present.

However, if you specify Y as the value for this keyword, you can selectively prevent or permit moves of such minidisks by means of the MOVE IF IPL TEXT PRESENT data entry field on the Move Entry panel.

Default Value

The default value N is assumed if this keyword-value pair is omitted.

```
MINIDISK_IPLTEXT_LIMIT   N    * Prevent moving IPL-text Mdisks
```

MINIDISK_LINK_MODE_LIMIT

This keyword specifies the conditions under which a minidisk server virtual machine might establish a link to the source minidisk. The 1-digit value indicates the type of link.

The acceptable values are:

1

Establishes a link if there are no existing links at the time. The server virtual machine issues a CP link mode EW for the minidisk.

2

Establishes a link if there are no existing write links at the time. The server virtual machine issues a CP link mode SM for the minidisk.

3

Establishes a link despite existing read/write links. The server virtual machine issues a CP link mode MW for the minidisk.

If you specify value 2 or value 3, the minidisk is still moved if there are current read links. Also, references to the minidisk still address the same locations on the DASD even after the move.

Specifying value 3 is not recommended unless you are sure no concurrent writes will take place. More than one user writing to the same virtual device can result in a permanent loss of data.

Note: The above issues are effective only at the time of the initial link of the server virtual machine to the source minidisk.

Default Value

The value 1 is the default and is assumed if this keyword-value pair is omitted.

The server, depending on the job, needs to link to minidisks to gain access to the data residing there. To ensure data integrity, value 1 is highly recommended as the link condition.

When a storage administrator submits a MOVE request, the value in the EXISTENCE LINKS option of the request can not exceed the MINIDISK_LINK_MODE_LIMIT value. If it does, the MOVE request will fail.

```
MINIDISK_LINK_MODE_LIMIT  1    * Establish link if No links (W)
```

MINIDISK_MISMATCH_LIMIT

This keyword specifies whether or not to prevent the minidisk server virtual machine from moving a CMS minidisk when there is an inconsistency between the sizes of CP and CMS.

There are two acceptable values. N prevents the movement of such minidisks, and Y permits the movement of such minidisks.

If you specify N as the value for this keyword, you *cannot* override this restriction by means of the Move Entry Panel data entry field, MOVE IF CP/CMS SIZE MISMATCH; the minidisk server virtual machine never attempts to move minidisks with CP/CMS size inconsistency. However, if you specify Y for this keyword value, you can selectively prevent or permit moves of such minidisks by means of the MOVE IF CP/CMS SIZE MISMATCH data entry field on the Move Entry panel.

Default Value

The default value N is assumed if this keyword-value pair is omitted.

```
MINIDISK_MISMATCH_LIMIT  N    * Prevent move if size mismatch
```

MINIDISK_RESERVED_LIMIT

This keyword specifies whether to allow or prohibit ISMF MOVE operations for reserved minidisks. Specify a one-character value to designate whether moves of CMS reserved minidisks are permitted.

There are two acceptable values. N prevents a server virtual machine from moving reserved minidisks, and Y permits a minidisk server virtual machine to move reserved minidisks.

If you specify N as the value for this keyword, you *cannot* override this restriction by means of the Move Entry Panel data entry field, MOVE IF RESERVED MINIDISK; a minidisk server virtual machine never attempts to move minidisks that are reserved. However, if you specify Y for this keyword value, you can selectively prevent or permit moves of such minidisks by means of the MOVE IF RESERVED MINIDISK data entry field on the Move Entry panel.

Default Value

The default value N is assumed if this keyword-value pair is omitted.

```
MINIDISK_RESERVED_LIMIT   N    * Prevent reserved Mdisk moves
```

Control File Keywords Required by Removable Media Services (RMS)

The following section describes keywords used to control removable media services (RMS).

Note: To run RMS, you must specify the DFSMSRM master, automatic tape library, work directory, and global resource ID in the control file.

DFSMSRM_MASTER_VM

This keyword specifies the DFSMS removable media services (RMS) master virtual machine, identified by a one- to eight-character user ID. The virtual machine name must not be the same as the name specified for any other virtual machine. This virtual machine handles all DFSMS requests for RMS functions, such as:

- Mounting or demounting cartridges
- Assigning scratch pools to library devices
- Assigning cartridges to scratch pools
- Providing library information
- Stopping the RMS master.

If you installed an IBM 3495 Tape Library Dataserver, this keyword must be included in the control file.

Default value

The keyword-value pair must be specified for operation of the RMS master. There is no default for this keyword; however, the sample control file has RMSMASTR as the virtual machine name.

```
DFSMSRM_MASTER_VM   RMSMASTR
```

RM_ACCOUNTING

This keyword specifies whether RMS should perform accounting processing or not. DFSMS/VM accounting values are only valid on DFSMS/VM systems running on first level machines.

Valid values for this keyword are yes (Y) and no (N). This keyword value is optional. If this keyword is omitted, RMS will not perform accounting processing.

See "Step 3: Enable the DFSMS/VM Accounting Function" on page 20 for further requirements.

Default value

The default value for this keyword is N.

```
RM_ACCOUNTING   N
```

RM_AUTO_LIBRARY

This keyword specifies an automatic tape library used by the z/VM system.

Each automated library is identified by a one- to 32-character name, followed by the actual five-character automatic tape library sequence number (must match the number provided by the IBM hardware), and a one- to eight-character user ID identifying the user to receive information related to RMS master processing events.

This keyword is used once for each automatic tape library used.

Default value

There are no default values for this keyword; however the sample control file has VMSYSATL as the library name. No default value is provided for the sequence number or user ID.

Note: This information must be provided prior to initializing the RMS master virtual machine.

```
RM_AUTO_LIBRARY  VMSYSATL seqnum userid
```

RM_DEFAULT_SCRATCH_POOL

This keyword and its associated value specify the name of the scratch category to be treated as the default scratch pool.

DFSMS/VM supports 16 scratch categories named SCRATCH*n*, where *n* is 0-9 or A-F. The customer may select which scratch pool is to be considered the default for mount requests that specify SCRATCH, with no suffix character defining the pool. Valid values are SCRATCH0 through SCRATCH9 and SCRATCHA through SCRATCHF.

Default value

The default value provided in the control file is SCRATCH0 and is enforced if this parameter is not found in the control file.

```
RM_DEFAULT_SCRATCH_POOL SCRATCH0
```

RM_FOREIGN_SERVER_VM

This keyword specifies a user ID for a service virtual machine that will handle requests from foreign hosts for library operations. This keyword may be specified one or more times. In an environment in which all processors attached to a library are capable of controlling the library, this keyword is not applicable.

However, in an environment in which there is a processor that is connected to a library tape drive but does not support library control operations, a foreign host server may be identified by means of this parameter. The foreign host server handles requests from a foreign host. See *z/VM: DFSMS/VM Removable Media Services* for more details.

This keyword is not required in the control file, even if a foreign-host server is being utilized. If this keyword is included, the specified one- to eight-character user ID is autologged by the RMS master (using the DGTQAUTO installation-wide exit) during RMS master initialization.

Default value

The foreign server is not typically required in the z/VM environment. Thus, the product default is that the foreign server is not autologged, nor is its definition required.

```
RM_FOREIGN_SERVER_VM  rmforsrv
```

RM_LOG_TO_CONSOLE

This keyword specifies the severity level of messages that are to be written to the console of the RMS master machine. The value of this keyword-value pair specified is a one-character field, which may be one of the following:

0

No messages are to be logged

1

Severe messages are to be logged

2

Severe and error messages are to be logged

3

Severe, error, and warning messages are to be logged

4

Severe, error, warning, and information messages are to be logged.

Message level 0 cannot be specified for both the console and log file destinations. Because RMS must be able to log severe messages somewhere, one of these keywords must specify a message level of 1 or greater.

Note: To assist in problem diagnosis and problem resolution, it is recommended that either RM_LOG_TO_CONSOLE or RM_LOG_TO_FILE always be set to 4.

Default value

The default value of the keyword-value pair is 3.

```
        RM_LOG_TO_CONSOLE  3            * Messages logged to console
```

RM_LOG_TO_FILE

This keyword specifies the severity level of messages that are to be written to the log file of the RMS master machine and the directory in which the log files are stored. The first value specified with this keyword is a one-character field, which may be one of the following:

0

No messages are to be logged

1

Severe messages are to be logged

2

Severe and error messages are to be logged

3

Severe, error, and warning messages are to be logged

4

Severe, error, warning, and information messages are to be logged.

The second value is a one- to 153-character directory name in which the log files will be created. The directory must not be a directory control directory. If the first value specified is 0, then the second value can be omitted.

Message level 0 cannot be specified for both the console and log file destinations. Because RMS must be able to log severe messages somewhere, one of these keywords must specify a message level of 1 or greater.

Note: To browse current log files, you need to XEDIT the log files with the NOLOCK option because DFSMS/VM has that current file locked.

If you are running space management, the top level directory that you specify should not have *DFSMS* as its first five characters. (A directory that begins with *DFSMS* cannot be managed by DFSMS/VM. It is recommended that you allow DFSMS/VM to manage the log files.) Because log files can grow rapidly, you should not put log files in the same file space as the work directory. Doing so can use up required space in the work directory and DFSMS/VM will be unable to successfully complete requests.

The file name of a log file is the name of the master or server machine that created it. The file type is the date that the log file was created. A new log file is started with the first request after midnight of each day and is the active file for that day.

The following are examples of log file IDs:

```
RMSMASTR 20020331
RMSSRV01 20020330
```

Default value

The default value for this keyword is 0. If you want to log to a file, you must specify both message level and directory ID. If the log to file directory ID does not exist, DFSMS/VM will create it for you. Even so, the filespace must already be enrolled.

```
RM_LOG_TO_FILE    0 dirid        * Messages logged to file
```

Attention

DFSMS/VM always keeps its logs files open, so a full backup cannot be performed with the storage group utility, since the utility allows no files to be opened for WRITE.

If log files are placed in a storage group with other data, the DFSMS/VM system must be down to issue FILEPOOL BACKUP against that storage group. You may want to consider putting the log files in their own storage group so that your FILEPOOL BACKUP is unaffected by DFSMS being up or down.

If your log file space is not DFSMS-managed, you may want to implement an alternate technique for deleting old log-file data and avoiding full file pool conditions.

RM_MANUAL_LIBRARY

This keyword allows you to specify the manual libraries in the control file, and is specified once for each manual library. The one- to 32-character library name may be specified in mount requests handled by the RMS master, in which case, the mount request is forwarded to the one- to eight-character user ID associated with the keyword-value specification. The capacity of RMS master to forward manual mount requests to an appropriate operator allows the DFSMSRM mount command to be used installation-wide.

Default value

There is no default value for this keyword; however the sample control file has MANUALIB as the library name and the user ID is OPERATOR.

```
RM_MANUAL_LIBRARY  MANUALIB OPERATOR
```

RM_REQUEST_QUEUING

This keyword specifies whether or not requests for mount and fill operations are to be queued in the library manager component if the library is in a pause mode, or if such requests are to be rejected. Queued requests are handled as soon as the library is no longer in pause mode.

Default value

The default value for this keyword is Y.

```
      RM_REQUEST_QUEUING  Y
```

RM_USE_GIVE

This keyword specifies whether a CP GIVE command or a CP DETACH/ATTACH command is used to provide the tape device to the target user for any request that specifies device attachment upon completion. RM_USE_GIVE can be set to either of the following:

- N – use a CP DETACH or CP ATTACH command instead of a CP GIVE command.
- Y – use a CP GIVE command instead of a CP DETACH or CP ATTACH command.

Default value

The default value for this keyword is Y.

```
      RM_USE_GIVE  Y
```

RM_WRITE_PROTECT

This keyword specifies the default value for the setting of write protection on the tape drive, after a mount request has been fulfilled. The default value specified in the control file applies only when the READONLY/READWRITE option is omitted in the mount request syntax.

Acceptable values are READONLY (set logical write protection on) and READWRITE (ensure logical write protection is not on).

Default value

The default value of this keyword-value pair is READONLY.

```
      RM_WRITE_PROTECT  READONLY
```

Chapter 6. Using DFSMS/VM Installation-Wide Exits

This chapter contains Programming Interface information.

This chapter is intended to help you understand what installation exits are available, what default installation exits are shipped with the product, and how you can modify them.

To accommodate installation requirements, DFSMS/VM provides several installation-wide exits. Except for the DGTQAUTO, FSMACCNT, and FSMVAUTH exits, the exits can be divided into three groups: minidisk management, space management, or removable media services exits.

How to Modify Installation-Wide Exits

If you change any of the installation-wide exits, apply those changes using VMSES/E. Never modify the base code. Instead, copy the part you wish to change to the VMSES/E local modification disk, make and test your change, and then apply the change. This can be done as follows:

Updating EXECs

1. Place changed copy of the EXEC on VMSES/E disk LOCAL.
2. Rebuild FSMBLEXC by using the following command:

```
VMFBLD PPF override compname FSMBLEXC (ALL
```

where *override* is the name of the $PPF file for your installation, and *compname* indicates the DFSMS/VM component you have installed.

Updating CSL Routines

1. Create changed text deck and put on VMSES/E disk LOCAL.
2. Rebuild FSMTXTLB by using the following command:

```
VMFBLD PPF override compname FSMTXTLB (ALL
```

3. Rebuild the CSL routine by using the following command:

```
VMFBLD PPF override compname type (ALL
```

where *type* is FSMNCCSL if you are implementing space management or RMS – otherwise *type* is FSMBLCSL.

For more information, refer to the *z/VM: CMS Application Development Guide for Assembler*.

Common Installation-Wide Exits

Some installation-wide exits are common to all three functions of DFSMS/VM. They are listed as follows:

DGTQAUTO
EXEC called to autolog a server

FSMACCNT
CSL routine called before an accounting record is generated.

FSMVAUTH
CSL routine called to perform authorization checking.

Note: The DGTQAUTH EXEC is not available with DFSMS/VM Function Level 221. If your system replaced the DGTQAUTH EXEC to authorize users of DFSMS/VM Function Level 210, you can continue to use the DGTQAUTH EXEC by creating your own FSMVAUTH CSL routine to invoke it.

Minidisk Installation-Wide Exits

The minidisk management exits are all EXEC exits and may be written in EXEC, EXEC2 or REXX. They are invoked using EXEC protocol. The exits that are shipped with the product are all written in REXX. The following are minidisk management exits:

DGTQDIRM
> Directory Maintenance Interface

DGTQLINK
> Server Minidisk Linking

DGTQSTEP
> Server Logical Step Progression.

Note: Due to the merging of the minidisk master and DFSMS/VM master, five minidisk exits (DGTQCMSF, DGTQLOCL, DGTQMMSG, and DGTQMESG, DGTQXMIT) have been removed and are no longer available.

Space Management Installation-Wide Exits

Due to performance considerations, the space management exits, except for the optional ACS REXX exit, are either CSL routines or executable modules. The space management exits are:

FSMMECHK
> A CSL routine called before a file is migrated or erased.

IGDACSMC DFSMS
> An ACS REXX exit called to determine a management class during ACS processing.

IGDACSMC MODULE
> An ACS module exit used to determine a management class during ACS processing.

For detailed ACS information, refer to the *z/VM: DFSMS/VM Storage Administration*.

Removable Media Services Installation-Wide Exits

RMS processing includes several types of installation-wide exit that may be modified to customize support of the IBM 3495 Tape Library Dataserver functions. All of the removable media services exits are CSL routines.

Note: Use of the RMS type of installation-wide exit is optional. This type of installation-wide exit allows your installation to add to or change default processing in certain areas of RMS processing.

The following types of installation-wide exit are used in RMS processing:

FSMRMATE
> Called to issue commands for device attachment

FSMRMDEV
> Called to select a real device if none is specified in the request syntax

FSMRMDTE
> Called to issue commands for device detachment

FSMRMPRE
> Called before the processing of an authorized request

FSMRMPRO
> Called after the processing of an authorized request

FSMRMSHR
> Called to verify that volumes and categories specified in requested commands do not violate partitioning criteria for IBM 3495 Tape Library Dataservers accessed concurrently by other operating systems.

For detailed information on the removable media services exits, refer to *z/VM: DFSMS/VM Removable Media Services*.

DGTQAUTO—Master XAUTOLOG of Server Virtual Machines

The DGTQAUTO EXEC provides the means for the DFSMS/VM master machine to log on server machines. This requires either that the master machines be class A or B machines, or that the server machines contain the XAUTOLOG statement in their CP directory entry.

The SMSMASTR invokes the DGTQAUTO EXEC during initialization for each space management server virtual machine. The RMSMASTR invokes the DGTQAUTO EXEC during each initialization for each foreign server.

If during normal operation, the DFSMS master machine has outstanding space management work and detects that a space management server machine is not logged on, it autologs it via the DGTQAUTO EXEC.

The minidisk servers are handled differently than the space management servers. They are not autologged at DFSMS master machine initialization. Rather, they are only autologged when there is minidisk management work to do.

When the DGTQAUTO EXEC is invoked by a master machine it is passed the user ID of the server machine to be logged on. As shipped, the DGTQAUTO EXEC issues the XAUTOLOG command against the user ID that was passed to it from the master machine, and returns the return code from the command. The following is an example of the XAUTOLOG command:

```
CP XAUTOLOG userid
```

The DFSMS master machine displays the return code and continues. A nonzero return code from the EXEC is ignored by the DFSMS master.

You may need to make modifications to the EXEC for your installation requirements.

Input

The user ID of the server virtual machine to be automatically logged on.

Output

Return code from the XAUTOLOG command.

DGTQDIRM—Server Directory Interfacing

Use this exit to interface with DIRMAINT or its equivalent.

The minidisk server virtual machine invokes the DGTQDIRM exit in the following circumstances:

- A directory request operation is required.
- A message (possibly from a directory maintenance virtual machine) is received.

The server virtual machine expects this EXEC to return immediately every time; that is, all waiting is done by the server virtual machine code, not in this exit.

Input

A directory interface function and parameters:

Input Function	Input Arguments
AMDISK — Add minidisk	AMDISK directory_request parameters
DMDISK — Delete minidisk	DMDISK director_request parameters
MMDISK — Mirror minidisk	MMDISK directory_request parameters

Input Function	Input Arguments
RMDISK — Redefine minidisk	RMDISK directory_request parameters
IPL_MSG — Analyze a message	IPL_MSG message_text
IPL — Initialize the EXEC	IPL

Output

A return code of the directory interface function: when the return code is greater than zero and if an associated message exists, the message is appended to the return code. This EXEC must not use EXIT, but must return to the caller. The value returned by this EXEC is a character string composed of the return code and, optionally, the associated message.

- **IPL (normal exit):**

  0

    Initialized

- **IPL (error exit):** Server initialization will fail

  4

    DIRMAINT userid missing

  8

    DIRMAINT userid invalid

  12

    Communication method missing

  16

    Communication method invalid

- **IPL_MSG (normal exit):**

  140

    Response not evaluated

- **IPL_MSG (error exit):** Server initialization will fail

  130

    Server not authorized

- **Other (normal exit):**

  0

    Command request accepted

- **Other (error exit):** Server operation will fail

  4

    DIRM_VM not logged on

  8

    Reserved

  12

    Command request rejected

DGTQLINK—Server Minidisk Linking

The minidisk server virtual machine invokes the DGTQLINK exit prior to each CP LINK command attempt for source and target minidisks, and after the detach of source and target minidisks. The default DGTQLINK exit supplied with DFSMS/VM provides a means to warn users and optionally force them to relinquish access to a disk so that the server virtual machine can access a disk for a move or check operation. In addition, previously linked users are notified that a disk is once again available.

Before the link is attempted, the minidisk server virtual machine sends a message to users currently linked to the minidisk requesting that they detach. After the move processing is complete, the server virtual machine notifies the users previously linked that the minidisk is again available.

Input

The following parameters:

- (1) Current operation phase

    IPL_LINK (initialize)
    LINKB (link before use)
    LINKA (link after use)

- (2) Link to user ID
- (3) Link to virtual address
- (4) Link as virtual address
- (5) Link access mode
- (6) Link attempt number
- (7) Link attempt maximum

Output

Return code describing the success of the function.

- Normal exit:

    0
    execution completed successfully

- Error exit:

    parm(1)
    missing or invalid

DGTQSTEP—Server Logical Step Progression

The move and check processes performed by the minidisk server virtual machine are divided into logical steps (see Table 5 on page 61). The server virtual machine invokes the DGTQSTEP exit at the start of each logical step. Based on the return code from the DGTQSTEP exit, the server virtual machine proceeds in one of the following ways:

- Proceed normally (RC=0)
- Skip this step (RC=4)
- Abort this job (RC=8)

This exit allows local control of the requested DFSMS/VM move and check operations.

Input

The functions MOVE or CHECK and the step number.

Table 5. Step Numbers for Move and Check

Step Numbers for MOVE	Step Numbers for CHECK
1. Parameter list syntax analysis	1. Parameter list syntax analysis
2. Source minidisk acquisition	2. Source minidisk acquisition
3. Target minidisk acquisition	3. DFSMS check and deliver output

Step Numbers for MOVE	Step Numbers for CHECK
4. Physical data movement	4. Housekeeping
5. Logical data movement	
6. Housekeeping	

Output

Return codes as described. It is recommended that DGTQSTEP return code 0, or return code 8 if appropriate. Return code 4 should be used with caution. For example, if the step-in-progress is move syntax analysis and DGTQSTEP EXEC returns code 4, this step is skipped. The remaining steps are dependent upon data whose integrity has not been verified, and thus results are not guaranteed.

Note: The housekeeping steps will not be skipped, no matter what return code is returned from the exit.

FSMACCNT— Accounting Exit

The FSMACCNT accounting exit is a CSL routine that provides the means for an installation to modify an accounting record just prior to generating the record or to stop a record from being generated. You may customize the output of an accounting record or suppress the generation of accounting records by replacing the CSL routine FSMACCNT. See "Updating CSL Routines" on page 57 for details. As shipped, the CSL routine returns a 0 return code (write the record).

Usage Note

FSMACCNT is invoked at the end of the specified DFSMS/VM processing. A 70 character buffer, representing columns 9–78 in the accounting record, is passed to the FSMACCNT accounting exit. This exit is invoked as follows:

Invocation	**FSMACCNT,**_retcode_	_,buffer_ _,command_

Parameter
    Meaning

retcode
    (output, int, 4) return code passed back to DFSMS/VM.

buffer
    (input/output, char, 70) character variable of length 70, which is used for both input and output. On input, this buffer is formatted as described above for columns 9–78 in the DFSMS/VM USER accounting record format (if record type is USMS) or DFSMS/VM SYSTEM accounting record format (if record type is SSMS). On output, this buffer is used to replace columns 9–78 in the accounting record about to be generated.

The following return codes are expected from the installation exit:

Return code
    Meaning

0
    Proceed with writing this accounting record with the returned buffer placed in columns 9–78.

8
    Do not write this accounting record.

Any other return code will be treated as return code 8. (Do not write this accounting record).

DFSMS/VM Accounting Records

DFSMS/VM produces two types of accounting records: user and system.

USER

DFSMS/VM virtual machines write USER accounting records at the end of request processing to reflect CPU and storage resources spent to process a user's request. A USER accounting record type is identified by a 'U' in column 75.

SYSTEM

DFSMS/VM virtual machines write SYSTEM accounting records to reflect CPU and logging storage resources spent as overhead for DFSMS/VM processing. A SYSTEM accounting record type is identified by an 'S' in column 75.

The contents of the buffer contain the portion of the user or system accounting records, columns 9–78, which can be modified by an installation just prior to generation of the accounting record. For more detail on the record layouts and field descriptions, refer to the *z/VM: DFSMS/VM Storage Administration*.

FSMMECHK—Exit before Migration or Expiration of a File

DFSMS MANAGE and MIGRATE processing allows files to be migrated or erased. The FSMMECHK CSL routine allows the installation to monitor this process and optionally prevent this migration or erasure from occurring. For example, the installation may desire that certain files not be migrated until a current backup copy of the file has been made.

To customize this exit, you will need to replace it. Refer to "Updating CSL Routines" on page 57 for details.

Just before DFSMS/VM migrates or erases a file, it invokes the FSMMECHK exit. This exit tells DFSMS/VM whether or not to continue with the migration or erasure of the file. This exit is invoked as follows:

Call to DMSCSL	**FSMMECHK,**_retcode_	_,fn ft dirid_ _,length1_ _,function_ _,command_ _,userid_

Parameter
    Meaning

retcode
    (output, int, 4) return code passed back to DFSMS/VM.

fn ft dirid
    (input, char, *) contains the file name, file type, and directory ID for the file being processed. The filename is left justified in positions 1–8. It is followed by 1 space. The file type is left-justified in positions 10–17. It is followed by 1 space. The directory ID is left-justified in positions 19–171. For example:

```
fn      ft      filepoolid:userid.n1.n2...
```

length1
    (input, int, 4) contains the length of the preceding character parameter. It can be a maximum of 171 characters long.

function
    (input, char, 8) contains the function performed on the file. This will be one of the following:

- MIGRATE (migrating to ML1)
- MIGRATE2 (migrating to ML2)

- ERASE (erasing the file).

command
(input, char, 8) command that is being executed on the file. This will be one of the following:

- MIGRATE (migrating because of the MIGRATE command)
- ERASE (erasing the file).

userid
(input, char, 8) contains the ID of the user who issued the command.

The following return codes are expected from the installation exit:

Return code
   Meaning

0

Proceed with this file.

4

Proceed with this file, and do not call this exit for this function for any more files associated with this command if a MANAGE command has been issued.

8

Proceed with this file, and do not call this exit for any function for any more files associated with this command if a MANAGE command has been issued.

12

Do not proceed with the file.

16

Do not proceed with the file, and do not call this exit for this function for any more files associated with this command if a MANAGE command has been issued.

20

Do not proceed with the file, and do not call this exit for any function for any more files associated with this command if a MANAGE command has been issued.

Other

Any other return code is treated as if a 12 has been returned. This is an invalid return code. An error message will be logged.

IBM's version of the CSL routine, which is shipped with DFSMS/VM, returns a return code of 8 on the first call, thereby eliminating further calls to the CSL routine for this command.

FSMMECHK is called at least once for each MANAGE command.

- If the return code from FSMMECHK is 4 or 16, the exit is not called again for that function (MIGRATE, MIGRATE2, or ERASE) for the duration of that MANAGE command.
- If the return code from FSMMECHK is 8 or 20, the exit is not called again for any function for the duration of the MANAGE command.
- If FSMMECHK is not called for a particular file, it is migrated or erased as indicated by the management class attributes.
- And, if another MANAGE command is issued, FSMMECHK is again called at least once.

When a DFSMS MIGRATE command is processed, the start of the FSMMECHK exit is handled differently than when a DFSMS MANAGE command is processed and files are migrated due to MANAGE processing. For each file that the DFSMS MIGRATE command selects, the FSMMECHK exit is called regardless of previous return codes that may have been returned for that DFSMS MIGRATE command.

FSMVAUTH—Authorization Exit

DFSMS/VM has a security scheme that primarily uses RACF/VM and an authorization file. To allow your installation to use an alternate security method, DFSMS/VM offers an authorization exit in the form of a

CSL routine. You may customize the authorization process by replacing the CSL routine, FSMVAUTH. As shipped, the CSL routine returns a return code of 4 (indicating no decision has been made).

DFSMS/VM once supported DGTQAUTH EXEC as an authorization exit. This interface has been replaced. If your installation replaced this exit, the solution for the installation is as follows:

1. Write an authorization exit (a CSL routine) called FSMVAUTH, which starts the DGTQAUTH EXEC with the currently supported parameters.

2. Translate the results of the DGTQAUTH EXEC into a return code supported by the FSMVAUTH CSL routine.

3. Place this CSL routine on your local disk to have VMSES/E build it to the build disk, in order to make it known to DFSMS/VM. Refer to "Updating CSL Routines" on page 57.

The original DGTQAUTH invoked DGTFIS27. If your modified version invokes this routine, that call must be deleted, because DGTFIS27 is no longer a shipped part. This routine has been replaced by the authorization scheme described in Figure 20 on page 40.

Usage Note

All file pools that are managed by DFSMS/VM must be authorized to the APPC command. However, only file pools should be authorized to the APPC command to prevent unauthorized access to secondary storage.

When a command (such as DELETE, CONVERT, MANAGE) utilizing a file pool is issued, the authorization exit (FSMVAUTH) is called once to verify that the file pool is authorized to connect to DFSMS/VM and once for authorization checking of the command. When verifying that the file pool is authorized to connect to DFSMS/VM, the user ID of the file pools server is passed in the user ID parameter, and the command is set to APPC. When a DISCARD (DFSMS or DFSMSRM), QUERY REQUEST (DFSMS or DFSMSRM) or REPORT SPACEMANAGEMENT FILESPACE (DFSMS) command is issued, the authorization exit (FSMVAUTH) is called to check for ANY authority (for example, DISCARDA), and if that authorization fails, it is then called to check for OWN authority (for example, DISCARDO).

The exit is invoked as follows:

Call to DMSCSL	**FSMVAUTH,**_retcode_	_,userid_ _,command_

Parameter
   Meaning

retcode
   (output, int, 4) return code passed back to DFSMS/VM.

userid
   (input, char, 8) userid of the command issuer or file pool server.

command
   (input, char, 8) contains the command that called the exit. This can be one of the following:

DFSMS/VM commands

ACTIVATE
   User copies and loads contents of source configuration file into active configuration file via the DFSMS or ISMF ACTIVATE command

ALTFILE
   User changes the management class of files or directories via the DFSMS or ISMF ALTER command

APPC
   User connects to a DFSMS/VM-managed resource reserved for file pool servers via the APPC command

BUILDLST
User builds a minidisk list via the ISMF BUIDL LIST command

CHKDSK
User checks a minidisk via the ISMF CHECK command

CONVERT
User assigns management classes to files and directories via the DFSMS CONVERT command

DELFILE
User deletes files from DFSMS/VM ML1 and ML2 storage via the DFSMS DELETE command

DISCARDA
User discards another's request via the DFSMS or ISMF DISCARD (any) command

DISCARDO
User discards his own request via the DFSMS or ISMF DISCARD (user) command

MANAGE
User expires and migrates files via the DFSMS MANAGE command

MIGRATE
User moves files from SFS to DFSMS/VM storage via the DFSMS MANAGE command

MOVEDISK
User moves a minidisk via the DFSMS or ISMF MOVE command

QUERYA
User queries another's request via the DFSMS or ISMF QUERY (any) command

QUERYO
User queries his own request via the DFSMS or ISMF QUERY (user) command

QUERYS
User issues query status via the DFSMS QUERY STATUS command

RECALL
User moves a migrated file from DFSMS/VM to SFS via the DFSMS RECALL command

REPORTA
User reports on another's file space/storage group via the DFSMS REPORT SPACEMANAGEMENT FILESPACE (any) or DFSMS REPORT SPACEMANAGEMENT STORGROUP command

REPORTO
User reports on his own file space via the DFSMS REPORT SPACEMANAGEMENT FILESPACE (user) command

STOPSMS
User shuts down the DFSMS master and server machines via the DFSMS STOP command.

DFSMSRM commands

RMDISCDA
User cancels a command issued by another virtual machine via the DFSMSRM DISCARD (any) command

RMDISCDO
User cancels a command issued by its own virtual machine via the DFSMSRM DISCARD (own) command

RMDMOUNT
User requests a volume on a specified drive be demounted via the DFSMSRM DEMOUNT command

RMMOUNT
User specifies a volume to be mounted via the DFSMSRM MOUNT command

RMQUERYA
User queries a request issued by another virtual machine via the DFSMSRM QUERY REQuests (any) command

RMQUERYL
User obtains information about the status of library, volume, or device via the DFSMSRM QUERY
LIBrary command

RMQUERYO
User queries a request issued by its own virtual machine via the DFSMSRM QUERY REQuests (own)
command

RMRSETDC
User disassociates a category from a device via the DFSMSRM RESET DEVCAT command

RMSETDC
User assigns a category to a library device via the DFSMSRM SET DEVCAT command

RMSETVC
User assigns a volume to a different category via the DFSMSRM SET VOLCAT command

RMSETVCB
User requests processing of a list of volume assignments to a different category via the DFSMSRM SET
VOLCAT BULK command

RMSTOP
User quiesces or halts the RMS master via the DFSMSRM STOP command.

The following return codes are expected from the installation exit:

Return code
  Meaning

0
  The exit has determined that the user is authorized to issue the command or perform the function.

4
  The exit cannot determine authority. Revert to DFSMS/VM authority checking. (This is the way the exit
  is shipped with the product.)

8
  The exit has determined that the user is not authorized to issue the command or perform the function.

When DFSMS/VM performs authority checking, it invokes the FSMVAUTH CSL routine to allow the
installation to determine the authority. If the exit returns a code of 4, it means that the exit has not
determined the authority, and that DFSMS/VM must do its normal authority checking. Refer to Figure 20
on page 40 to see the steps performed by DFSMS/VM to determine authority for functions.

Chapter 7. Customizing ISMF

ISMF, an ISPF application, has a structure modeled after ISPF. This chapter gives instructions on how to customize ISMF panels, messages, and EXECs. See the *ISPF Dialog Management Guide and Reference* for more details on the topics covered in this chapter.

Preparing to Customize ISMF

Because customization must be reapplied with each release, we recommend against customizing ISMF. If you decide you must customize ISMF, keep the following in mind:

1. Put copies of any modified ISMF panels, messages, or execs on the VMSES/E local modification disk, and issue the VMSES/E **VMFBLD** command as shown in the following example:

   ```
   VMFBLD PPF override compname partname (ALL
   ```

   where: *partname* can be:

   > FSMBLEXC — execs
   > DGTPLIB — if a panel
   > DGTPBLIB — if an uppercase panel
   > DGTMLIB — if a message
   > DGTMBLIB — if an uppercase message.

2. ISMF is copyrighted. Under the IBM licensing agreement, you can modify ISMF for your own use; you cannot modify it for commercial resale.

3. Never alter anything on the VMSES/E base disk. Make a copy of any file you need to alter and put the copy on the local modification disk; then apply the change using VMSES/E.

Other restrictions apply to the individual parts of ISMF.

Identifying Parts of ISMF That Can Be Customized

You can customize the following parts for all ISMF applications:

- Panels
- Messages
- EXECs.

Except for EXECs that are shipped as separate files, the parts of ISMF you can customize are shipped in individual libraries:

- Panel Library

  ISMF allows you to make the following changes to the panel library:

  > Change the initial priming values shipped with ISMF
  > Change the default values for data entry panels
  > Provide additional restrictions to values entered for certain fields on panels
  > Remove fields from functional panels
  > Change highlighting and color
  > Change the panel format
  > Modify existing functional panel text and help text
  > Add new fields to panels
  > Add new panels.

- Message Library

  ISMF allows you to modify existing messages and add new messages.

Panels and Messages

To customize ISMF panels and messages, you need to find the corresponding ISMF library:

1. Invoke ISMF. The ISMF Primary Option menu appears.
2. Type CMS FILEDEF on the command entry line to see which file definitions (filedefs) are in effect for panel and message libraries.

   The CMS FILEDEF command displays the file ID associated with the DDNAME for the library. Table 6 on page 70 below lists the DDNAMEs used by ISMF for messages and panels:

Table 6. ISMF DDNAMEs for Panels and Messages				
Library	**Language**	**DDNAME**	**FN**	**FT**
Panel	AMENG	$DGTPLIB	DGTPLIB	MACLIB
Panel	KANJI	$DGTPLIB	DGTPALIB	MACLIB
Panel	UCENG	$DGTPLIB	DGTPBLIB	MACLIB
Message	AMENG	$DGTMLIB	DGTMLIB	MACLIB
Message	KANJI	$DGTMLIB	DGTMALIB	MACLIB
Message	UCENG	$DGTMLIB	DGTMBLIB	MACLIB

Customizing ISMF EXECs

As part of customizing ISMF to suit your needs, you can modify the EXEC provided to run ISMF (ISMF EXEC) or other EXECs that are part of DFSMS/VM (for example, PRNTMINI EXEC).

Finding the EXEC You Want to Change

ISMF EXECs have a file type of EXEC and are placed on the production disk as part of the installation process.

Modifying an EXEC

You can add code to the REXX EXECs to provide additional functions, or you can modify existing functions to suit your needs.

If you are making a modification to ISMF EXEC, the following is an appropriate procedure to follow:

1. Make a copy of the ISMF EXEC and put it on the local modification disk.
2. Modify the copy of ISMF EXEC, either on the disk or on another minidisk from which you will be able to test the modified EXEC. (You should flag the changed or added lines to maintain a change history of customized EXECs.)
3. Test the EXEC and replace it on the disk as appropriate. See "Preparing to Customize ISMF" on page 69 for more details.

The example that follows describes a modification to ISMF EXEC that changes the location for PRNTMINI EXEC output. The PRNTMINI EXEC uses ISPF file-tailoring services to create output in the file associated with the $DGTFILE filedef, which is in the ISMF EXEC. In this example, the $DGTFILE filedef is modified so that output associated with this filedef is placed in DGTFILE MACLIB on the A-disk rather than in a CMS file.

Figure 21 on page 71 shows the original section of ISMF EXEC that provides the $DGTFILE filedef.

```
/***********************************************************/
/* ALLOCATE DGTFILE                                        */
/***********************************************************/
    'FILEDEF $DGTFILE DISK ISPNULL ISMFLIST 'user_disk,
          '(LRECL 255 RECFM V';
```

Figure 21. Original Section of ISMF EXEC

After modification, the $DGTFILE filedef portion of ISMF EXEC looks like <u>Figure 22 on page 71</u>; note the extra error checking and handling that is added to determine if the target MACLIB exists:

```
/***********************************************************/
/* ALLOCATE DGTFILE                                        */
/***********************************************************/
/*  'FILEDEF $DGTFILE DISK ISPNULL  ISMFLIST A ';  */  /*@UPD*/
      'STATE  DGTFILE MACLIB A'                       /*@UPD*/
     IF RC = 0 THEN                                   /*@UPD*/
       DO;                                            /*@UPD*/
        'FILEDEF $DGTFILE DISK DGTFILE  MACLIB A';  /*@UPD*/
       END;                                           /*@UPD*/
     ELSE                                             /*@UPD*/
       DO;                                            /*@UPD*/
          CONTINUE = 'NO ';                           /*@UPD*/
          ISMFRC = 20;                                /*@UPD*/
          ERROR_MSG = 'DGTUV998 ISMF FAILED, RETURN CODE: 'ISMFRC;
          CALL ADD_MSG;                               /*@UPD*/
          ERROR_MSG =  'DGTUV995  DGTFILE MACLIB A, NOT FOUND' ;
          CALL ADD_MSG;                               /*@UPD*/
       END;                                           /*@UPD*/
```

Figure 22. Modified Section of ISMF EXEC

Chapter 8. Performance and Tuning

Whether you are just an occasional user of DFSMS/VM, or if you are involved in configuring and defining the DFSMS/VM machines or any other complex storage management effort, you may want to become familiar with the material in this chapter.

Improving Performance

One way to improve performance of DFSMS/VM is to include the QUICKDSP and SHARE RELATIVE options in the CP directory entry of the server virtual machines.

Using the QUICKDSP Option

The QUICKDSP option designates a virtual machine as having immediate access to system resources. The result is that the virtual machine is added to the dispatch list immediately, whenever the virtual machine has work to do. This enables the DFSMS/VM virtual machines to process results quickly.

The QUICKDSP option can be specified either in the directory OPTION statement when defining a virtual machine, or it can be set dynamically with the CP SET QUICKDSP command.

Using the SHARE RELATIVE Option

The SHARE RELATIVE option changes the system-resource access priority for a virtual machine, giving it favorable position in the z/VM dispatch queue. The value specified with this option must follow your installation's operating procedures. Consider making the SHARE RELATIVE value less than that of the file pool servers. Consult your performance person for assistance in determining the correct value for your installation.

The SHARE RELATIVE option can be specified either in the directory SHARE statement when defining a virtual machine, or with the CP SET SHARE command with the RELATIVE operand. For information on these options, refer to *z/VM: CP Commands and Utilities Reference* or *z/VM: CP Planning and Administration*.

Setting up the TSM Server

TSM has a lot of flexibility in its hierarchy of storage for data, allowing you a great amount of flexibility in configuring the hierarchy. Ultimately, you should keep in mind that the purpose of DFSMS/VM using TSM as a repository is to provide ML2 to tape capability; therefore, you may want to set up TSM archive storage for ML2 in one of the following ways:

- One storage pool containing tape storage, or
- Two storage pools, where the first storage pool is a DASD staging area and the second storage pool is tape.

You can configure the TSM server with the QUICKDSP and SHARE RELATIVE directory options, to increase performance during processing.

If ML2 is being used, there are recommended settings for the following TSM system parameters:

1. Set registration to closed. By having registration closed, as opposed to open, applications cannot connect to the server and register themselves. DFSMS must be a registered node with the TSM server.
2. Set authentication to on in order to prevent unauthorized users from accessing the migrated data. If authentication is set to off, any user on the system can alter or erase ML2 data.

Chapter 9. Removing DFSMS/VM

This chapter describes the backout procedure to remove DFSMS/VM from your system after you have installed it and have begun migrating files. Removal of DFSMS/VM from production requires the following steps:

1. Recall **ALL** files that are migrated.
2. Update the primary file pools being managed by DFSMS/VM.
3. Perform miscellaneous cleanup.
4. Use the VMSES/E DELETE command to remove DFSMS/VM from your system.

   Refer to *z/VM: VMSES/E Introduction and Reference* for detailed information on removing a component.

Attention

Files that are backed up while in ML2 can only be restored with the help of the ML2 function. Files that are backed up while in ML1 can only be restored with the help of the ML1 function. As a result, the SFS FILEPOOL RESTORE command may not work if the DFSMS/VM migration function is removed. Therefore, it is recommended that you backup primary storage after you have recalled all files.

Recalling All Files to Primary Storage

Before you begin recalling all files, remove the capability for users to migrate files.

Important

If some files are still migrated when you remove DFSMS from operation, the data in those files will be inaccessible. Therefore, it is extremely important that no files remain migrated when you remove DFSMS from production.

To disable migration capability, you can do one of the following:

• Remove the DFSMS MODULE from *public* access. This prevents any general user from issuing a DFSMS command (including the MANAGE and MIGRATE commands, commands that cause migration to occur). However, retain access to the DFSMS MODULE for yourself, since you will need to issue DFSMS commands in later steps of this procedure.

• Use RACF/VM or the authorization exit to disable general user authority to the MIGRATE and MANAGE commands.

One reason to use space management is so that you can overcommit SFS storage groups. If you have been migrating files for an extended period of time, you may not have enough physical space in your primary storage groups to hold all the data to be recalled. Since SFS divides physical space at a storage group level, you should verify each storage group's ability to handle all the migrated data that is to be recalled. To do this, use the following procedure for each storage group in each file pool being managed by DFSMS/VM:

1. Determine how many blocks are available in each storage group for each DFSMS/VM-managed file pool. By issuing the `QUERY FILEPOOL STORG` command for the file pool containing the storage group, you can see how many blocks are available and how many are used.

   Below is a partial sample output of a QUERY FILEPOOL STORG command.

```
                VMSYSU    File Pool Storage Groups

Start-up Date 11/12/92                          Query Date 11/13/92
Start-up Time 07:50:03                          Query Time 07:58:51
=======================================================================
STORAGE GROUP INFORMATION

 Storage        4K Blocks       4K Blocks
 Group No.       In-Use           Free
    1             59 -   3%       2180
    2            507 -   4%      12963
 32767          1384 -  93%        106
=======================================================================
```

2. Determine how many blocks are required in primary storage to recover migrated files. Enter the following DFSMS command for each storage group in each file pool managed by DFSMS/VM:

```
    DFSMS REPORT SPACEMANAGEMENT STORGROUP groupnumber filepool
```

Figure 23 on page 76 shows some sample lines from the report.

```
    DFSMS REPORT SPACEMANAGEMENT STORGROUP processing started for file pool VMSYSU
    and storage group 2


    Files migrated for this storage group:                        15
      Migration Level 1:                                           8
      Migration Level 2:                                           7
    Logical 4K blocks currently in use by this storage group:     608
    Physical 4K blocks in use by this storage group:              353
    Physical 4K blocks in use by primary storage for this storage group:  340
    Physical 4K blocks in use by secondary storage for storage group:     13  1
      Migration Level 1:                                           6
      Migration Level 2:                                           7
    Physical 4K blocks saved for this storage group due to compaction:    255  2

    DFSMS REPORT SPACEMANAGEMENT STORGROUP completed with no errors
```

Figure 23. Sample Output from DFSMS REPORT SPACEMANAGEMENT STORGROUP Command

The combined total of **1** and **2** is the amount of additional space that will be needed to recall all files that are currently migrated for this storage group. If this number is higher than the amount of space available in the storage group (determined in step "1" on page 75), then add additional storage to the storage group before continuing the backout procedure. If the number of blocks used by all users is less than or equal to the number of blocks available in the storage group, then proceed to the next step of the backout procedure.

3. The report run in step "2" on page 76 lists users and whether or not they have migrated files. For every user who has migrated files, you can issue the following DFSMS/VM command:

```
    DFSMS REPORT SPACEMANAGEMENT FILESPACE filepoolid:userid
```

Figure 24 on page 77 gives an example of the output produced by the DFSMS REPORT SPACEMANAGEMENT FILESPACE command.

```
DFSMS REPORT SPACEMANAGEMENT FILESPACE processing started for file pool VMSYSU
and userid VMTEST2


Files migrated for this file space:                                5
  Migration Level 1:                                               2
  Migration Level 2:                                               3
Logical 4K blocks currently in use by this file space:           104
Physical 4K blocks in use by this file space:                     29
Physical 4K blocks in use by primary storage for this file space:  26
Physical 4K blocks in use by secondary storage for this file space: 5
  Migration Level 1:                                               3
  Migration Level 2:                                               2
Physical 4K blocks saved for this file space due to compaction:   75


   1  The following files are migrated:

                    Physical   Management  Date Last  Date      Migration
   Filename Filetype 4K Blocks    Class     Referenced Migrated    Level

   2  Directory VMSYSU:VMTEST2.

   TEST1    DATA     5          DEFAULT    10/30/92   10/29/92    1
   TEST1    ML1      50         ML1        10/30/92   10/30/92    1
   TEST10   DATA     48         DEFAULT    10/30/92   10/30/92    1
   TEST11   DATA     7          DEFAULT    10/30/92   10/30/92    1
   TEST12   DATA     3          DEFAULT    10/30/92   10/30/92    1


   DFSMS REPORT SPACEMANAGEMENT FILESPACE completed with no errors
```

Figure 24. Sample Output from DFSMS REPORT SPACEMANAGEMENT FILESPACE Command

4. Recall all files in the storage group. Now that you have made sure there is enough space in the primary storage group to accommodate all of the files in storage, you can recall all the files. Use the reports generated in step "3" on page 76 to obtain a list of all files you want to recall.

   There is a list of all migrated files, separated by subdirectory, after the file space utilization numbers in the report. The beginning of the list is identified with the message marked by **1**, and each subdirectory containing migrated files is identified with the message marked by **2**.

   For each directory listed, issue the command

   ```
   DFSMS RECALL * * directory_name
   ```

   to recall all files in that directory. Check the returned reader file to be certain that all eligible files have indeed been recalled. For those eligible files that have not been recalled, examine the error messages produced and follow the suggested action for the error message.

   You may want to issue the DFSMS REPORT SPACEMANAGEMENT STORGROUP command again to ensure that all migrated files have been recalled.

After all files have been recalled, you can stop DFSMS by using the DFSMS STOP SMS command. After doing so, any attempts to create files will generate error messages on the file pools being managed, as the file pool attempts to communicate with DFSMS. You can avoid creating these messages by leaving DFSMS running until you have completed the next step of the backout procedure (described in "Updating File Pools Managed by DFSMS/VM" on page 77).

Updating File Pools Managed by DFSMS/VM

Once all files have been recalled, reset the primary (managed) file pools to the state they were in before you installed DFSMS/VM. Do this by using the following procedure:

Note: For some of these changes to be made, the file pool server must be stopped. Other changes will require the file pool server to be logged off and then logged on.

1. Shut down or stop the file pool server.
2. Remove the DFSMS parameter from the DMSPARMS file for each file pool server. This parameter was added during DFSMS initialization. As long as the DFSMS parameter is in the file, DFSMS attempts to assign management classes to files and directories created in that file pool.
3. Remove ADMIN statements for the DFSMS/VM master and server virtual machines, which by default are SMSMASTR, SMSSRVnn, and DGTSRVnn.
4. Adjust the USERS parameter to what it was before installing DFSMS/VM. During DFSMS installation, it was recommended that you increase the USERS parameter to account for additional work done by DFSMS.
5. Adjust the MAXCONN value in the CP directory for every SFS file pool server virtual machine to what it was before installing DFSMS.
6. Remove the NAMESAVE DFSMSSEG statement from the CP directory entry for the file pool servers.
7. Remove the SEGMENT RESERVE DFSMSSEG statement from the PROFILE EXEC of the file pool servers.
8. Remove the LINK statement to the disk containing the DFSMS code from the PROFILE EXEC and SETUP EXEC.
9. Restart the file pool server.
10. Do file pool backups for all affected storage groups using the FILEPOOL BACKUP command or other backup tool.
11. Back up the "control data" using the FILESERV BACKUP command.

Miscellaneous Cleanup

After you have recalled all files and reset the primary file pools to their "pre-DFSMS" state, you can do the following:

Note: The following steps assume that other systems running DFSMS/VM are not using the same ML1 file pool and/or work and log directories. If this is not the case, do not erase any directories or delete the ML1 file pool being used.

1. Remove the user IDs of the DFSMS and RMS masters from the AUTOLOG process in use at your installation.
2. Erase the DFSMS work directory as it was specified in the control file (DGTVCNTL DATA in VMSYS:DFSMS.CONTROL).
3. Erase the DFSMS log files, if any exist, and the DFSMS log file directory if it was created for DFSMS processing. The log file directory is specified in the control file (DGTVCNTL DATA in VMSYS:DFSMS.CONTROL).
4. Erase the VMSYS:DFSMS file space by using the DELETE USER DFSMS VMSYS: (NOCONFIRM command.
5. Erase any files you placed on a public-access disk (such as ISMF-related files or the DFSMS MODULE).
6. Delete the CP directory entries for the DFSMS master and servers, the minidisk servers, and the RMS master.
7. Shut down the VMSYS file pool server in order to update its DMSPARMS file as follows:
   a. Remove ADMIN statements for DFSMS/VM master and server and RMS master virtual machines.
   b. Reduce the value used for the USERS parameter.
   c. Restart the file pool.
8. Purge the saved segment by using the PURGE NSS NAME DFSMSSEG command.
9. Delete the ML1 file pool (if this file pool was exclusively for ML1 and used by DFSMS only, with no other systems migrating to it.) If another DFSMS system is migrating files to this file pool, then you

cannot delete the file pool. If, however, the *file space* being used by *this* DFSMS system is not being used by any other DFSMS system, then you can remove the file space for this DFSMS.

10. Remove DFSMS/VM from DIRMAINT or the directory maintenance manager in use at your installation. For DIRMAINT, this means removing references to DFSMS/VM virtual machines from the DIRMAINT DATA and EXTENT CONTROL files.

11. Remove DFSMS/VM RACF profiles from RACF/VM or the security manager in use at your installation.

12. Delete the ML2 server, and any other resources associated with this user ID, if it was exclusively dedicated for DFSMS/VM ML2 data. If the ML2 server was shared with other applications, delete the data associated with node DFSMS.

Remove DFSMS/VM from Your System

If you wish to remove the DFSMS/VM code from your system, deleting the CP directory entry of the DFSMS installer virtual machine will return the installer's minidisk space. If you installed the DFSMS/VM HELP files to a system disk, you will need to manually remove those files.

Appendix A. DFSMS/VM Sample CP Directory Entries

Important

The specific CP directory entries shown in this appendix include those for the space and minidisk management master virtual machine, the space management server virtual machine(s), the minidisk management server virtual machine(s), and the RMS master virtual machine.

On a more general note, please be aware that any user ID issuing a DFSMS or DFSMSRM command, or using the CSL or ISMF interface, must have an IUCV ANY statement in its directory. See *z/VM: CP Planning and Administration* for more information on including this statement in a user directory.

Common CP Directory Entries

All master and server virtual machines require that the following common statements are included in the z/VM directory entries:

MACHINE ESA
> The master and server virtual machines run in ESA architecture mode.
>
> **Note:** Use of the MACHINE ESA statement requires APAR VM66110 to be applied; otherwise, the DFSMS servers will not initialize.

IPL CMS
> The master and server virtual machines require an initial program load (IPL) of CMS at machine logon time.

009 CON
> A virtual console device at virtual address X'009'

00C RDR
> A virtual reader device at virtual address X'00C'

00D PUN
> A virtual punch device at virtual address X'00D'

00E PRT
> A virtual printer device at virtual address X'00E'.

Note: The directory entries also include MDISK statements for product code, and/or READ WRITE work areas. Our samples use 3390 device types. MDISK statements must be modified with the appropriate device type, starting cylinder number, and volume label. If there is no indication in the provided sample directory entries that the size needs to be modified, then the size displayed in the samples is adequate for all device types.

Space and Minidisk Management Master Virtual Machine

is a sample CP directory entry for the space and minidisk management master virtual machine.

Note: Use of the MACHINE ESA statement requires APAR VM66110 to be applied; otherwise, the DFSMS servers will not initialize.

```
USER SMSMASTR password 32M 32M BG
 ACCOUNT 12345678
 MACHINE ESA
 IPL CMS
 IUCV ALLOW
 IUCV *IDENT RESANY GLOBAL REVOKE
 OPTION MAXCONN 400 QUICKDSP ACCT
 SHARE RELATIVE 1300
 NAMESAVE DFSMSSEG
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 MDISK 0191 3390 scyl 001 volid RR readpw writepw multipw
 LINK DFSMS 01B5 0192 RR
 LINK MAINT 0190 0190 RR
 LINK MAINT 019E 019E RR
```

Figure 25. DFSMS Master Virtual Machine CP Directory Entry

The user ID (SMSMASTR) in this CP directory entry must be the same as the user ID on the DFSMS_MASTER_VM parameter in the DFSMS/VM control file (DGTVCNTL DATA in VMSYS:DFSMS.CONTROL):

```
DFSMS_MASTER_VM        SMSMASTR              * USERID OF DFSMS MASTER
```

Directory Requirements

Consider the following directory requirements for the master virtual machine:

Storage Size
Use a virtual storage size of 32MB.

Privilege Class
The master virtual machine requires privilege class G, but the sample definition includes privilege class BG for the purpose of supporting the default autolog process. You must specify the appropriate privilege class according to the autolog process to be used at your installation.

Machine Mode
DFSMS/VM supports ESA mode.

IUCV ALLOW
This option is needed to allow other virtual machines to communicate with this virtual machine.

IUCV *IDENT
Allows this virtual machine to connect to the CP system service IDENTIFY to identify local and global resource IDs.

Option MAXCONN
Specify a MAXCONN value equal to the number of all tasks running in DFSMS/VM servers defined in the DFSMS/VM control file plus the maximum number of concurrent DFSMS/VM operations expected (determining management classes migrates, recalls, manage, and so forth), along with the number of minidisk servers, plus 50.

If you will be running with INVOKE ACS FOR FILE CREATION set to YES, ensure that your MAXCONN is set high enough to handle a connection for every simultaneous file and directory create. Refer to the *z/VM: DFSMS/VM Storage Administration* for more information on enabling ACS. For a system of 500 users and three DFSMS/VM servers with 8 tasks each and ACS enabled for file create, specify a MAXCONN value of at least 574. This assumes that each user can simultaneously create a file or directory or issue a DFSMS command.

If you will be running with INVOKE ACS FOR FILE CREATION set to DEFER or INHERIT, ensure that your MAXCONN is set high enough to handle a connection for every simultaneous directory create. For a system of 500 users and three DFSMS servers with eight tasks each and ACS disabled for file create, specify a MAXCONN value of at least 324. This assumes that not more than 250 users will be creating a directory or issuing a recall or migrate or other DFSMS command at the same moment. Use the following formula:

```
          (# of all tasks in DFSMS servers) +
          (# of concurrent DFSMS commands) +
              (# of minidisk servers) + 50
```

Note:

1. Each DFSMS/VM server can run a user-specified number of tasks (for example, 1, 5, 20). This allows a DFSMS/VM server to execute more than one unit of work at a time. The number of tasks is specified in the DFSMS/VM control file on the DFSMS/VM server virtual machine keyword.

2. The MAXCONN value controls the amount of concurrent work that the master can execute. If this value is set too low, your users will be unable to connect to this virtual machine, and their requests will fail. If this value is too high and you have that number of tasks needing to execute concurrently, you could see a performance degradation on this virtual machine.

QUICKDSP
Consider specifying that the virtual machine is to be added to the dispatch list immediately whenever there is work to be done, without waiting on the eligible list.

ACCT
Specify ACCT to allow all DFSMS/VM virtual machines to create accounting records.

SHARE RELATIVE
Consider specifying that this virtual machine receive a relative share of 1300 (or a value that meets your installation's requirements) of the system resources. (This number should be less than SFS recommendations for file pools and larger than a user relative share).

NAMESAVE
Allows access to the DFSMS/VM saved segment, for space management master virtual machines only.

Minidisk 0191
The 0191 minidisk holds the PROFILE EXEC, which is placed on this minidisk during installation. The DFSMS/VM server virtual machines link and access this as their 0191 disk. The 0191 minidisk is a read-only disk. Modify the device type, starting cylinder, and volume ID for your device.

LINK DFSMS 01B5
This virtual machine links to DFSMS 01B5 to gain access to the product code.

Space Management Server Virtual Machine

Figure 26 on page 83 is a sample CP directory entry for a space management server virtual machine.

Note: Use of the MACHINE ESA statement requires APAR VM66110 to be applied; otherwise, the DFSMS servers will not initialize.

```
USER SMSSRV01 password 32M 32M BG
 ACCOUNT 12345678
 MACHINE ESA
 IPL CMS
 OPTION MAXCONN 50 QUICKDSP ACCT
 OPTION LNKSTABL LNKEXCLU LNKNOPAS D84NOPAS
 OPTION DEVINFO
 SHARE RELATIVE 1300
 NAMESAVE DFSMSSEG
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 LINK SMSMASTR 0191 0191 RR
 LINK DFSMS 01B5 0192 RR
 LINK MAINT 0190 0190 RR
 LINK MAINT 019E 019E RR
```

Figure 26. Space Management Server Virtual Machine CP Directory Entry

The set of space management server virtual machine user IDs must all be identified in the DFSMS/VM control file (DGTVCNTL DATA in VMSYS:DFSMS.CONTROL) on DFSMS_SERVER_VM parameters:

```
DFSMS_SERVER_VM      SMSSRV01  # tasks    * USERID OF DFSMS SERVER 1
```

Note: Repeat definitions for each space management server virtual machine.

Directory Requirements

Consider the following directory requirements for the space management server virtual machines:

Storage Size
Use a virtual storage size of 32MB.

Privilege Class
The space management server virtual machine privilege class is BG.

Machine Mode
DFSMS/VM supports ESA mode.

Option MAXCONN
Specify a MAXCONN value that considers the number of tasks for space management server virtual machines. Use the following formula:

```
        (#tasks on this server * 3) + 10
```

QUICKDSP
Consider specifying that the virtual machine be added to the dispatch list immediately whenever there is work to be done, without waiting on the eligible list.

ACCT
Specify ACCT to allow all DFSMS/VM virtual machines to create accounting records.

Option LNKSTABL LNKEXCLU LNKNOPAS D84NOPAS
This statement is necessary to successfully link to the minidisk during a MOVE, CHECK, or when generating a minidisk list. The link may fail without this statement depending on the control file keyword given.

Attention: If you are in Cross System Extension (CSE) and one of the systems in the CSE complex is running on an operating system level less than VM/ESA Version 1.1.1, then remove LNKSTABL and LNKEXCLU from the OPTION statement in the minidisk server's directory entry.

Option DEVINFO
This statement is necessary to successfully link to minidisks during build list operations.

SHARE RELATIVE
This is a space management function only. Consider specifying that this space management server virtual machine receives a relative share of 1300 (or a value that meets your installation's requirements) of the system resources. (This number should be less than SFS recommendations for file pools and larger than a user relative share).

NAMESAVE
Allows access to the DFSMS/VM saved segment, for space management server virtual machines only.

LINK SMSMASTR 0191
Gain access to the common PROFILE EXEC. The 0191 minidisk is a read-only disk.

LINK DFSMS 01B5
This virtual machine links to DFSMS 01B5 to gain access to the product code.

Minidisk Management Server Virtual Machine

is a sample CP directory entry for a minidisk management server virtual machine.

Note: Use of the MACHINE ESA statement requires APAR VM66110 to be applied; otherwise, the DFSMS servers will not initialize.

```
USER DGTSRV01 password 32M 32M BG
 ACCOUNT 12345678
 MACHINE ESA
 IPL CMS
 OPTION QUICKDSP ACCT
 OPTION LNKSTABL LNKEXCLU LNKNOPAS D84NOPAS
 OPTION DEVINFO
 CONSOLE 009 3215
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 LINK SMSMASTR 0191 0191 RR
 MDISK 0500 3380 001 001 $$$$$$ MR
 MDISK 05FF 3380 002 001 $$$$$$ MR
 LINK DFSMS 01B5 0192 RR
 LINK MAINT 0190 0190 RR
 LINK MAINT 019E 019E RR
```

Figure 27. Minidisk Management Server Virtual Machine CP Directory Entry

The set of minidisk management server virtual machine user IDs must all be identified in the DFSMS/VM control file (DGTVCNTL DATA in VMSYS:DFSMS.CONTROL) on MINIDISK_SERVER_VM parameters:

```
MINIDISK_SERVER_VM   DGTSRV01
```

Note: Repeat definitions for each minidisk management server virtual machine.

Directory Requirements

Consider the following directory requirements for the minidisk management server virtual machines:

Storage Size
Use a virtual storage size of 32MB.

Privilege Class
The server virtual machine privilege class is BG.

Machine Mode
DFSMS/VM supports ESA mode.

QUICKDSP
Consider specifying that the virtual machine be added to the dispatch list immediately whenever there is work to be done, without waiting on the eligible list.

ACCT
Specify ACCT to allow all DFSMS/VM virtual machines to create accounting records.

Option LNKSTABL LNKEXCLU LNKNOPAS D84NOPAS
This statement is necessary to successfully link to the minidisk during a MOVE, CHECK, or when generating a minidisk list. The link may fail without this statement depending on the control file keyword given.

Attention: If you are in Cross System Extension (CSE) and one of the systems in the CSE complex is running on an operating system level less than VM/ESA Version 1.1.1, then remove LNKSTABL and LNKEXCLU from the OPTION statement in the minidisk server's directory entry.

Option DEVINFO
This statement is necessary to successfully link to minidisks during build list operations.

LINK SMSMASTR 0191
Gain access to the common PROFILE EXEC. The 0191 minidisk is a read-only disk.

Minidisk 0500
DFSMS uses the minidisk at address 0500 when processing a source minidisk for a MOVE or CHECK operation. Although the server virtual machine's 0500 minidisk must be defined in the directory, the actual extents (starting cylinder, number of cylinders, and volume serial number) are not important.

Minidisk 05FF
DFSMS uses the minidisk at address 05FF when processing a target minidisk for a MOVE operation. Although the 05FF minidisk must be defined in the directory, the actual extents (starting cylinder, number of cylinders, and volume serial number) are not important.

LINK DFSMS 01B5
This virtual machine links to DFSMS 01B5 to gain access to the product code.

RMS Master Virtual Machine

is a sample CP directory entry for the RMS master virtual machine.

Note: Use of the MACHINE ESA statement requires APAR VM66110 to be applied; otherwise, the DFSMS servers will not initialize.

```
USER RMSMASTR password 32M 32M BG
 ACCOUNT 12345678
 MACHINE ESA
 STDEVOPT LIBRARY CTL
 IPL CMS
 IUCV ALLOW
 IUCV *IDENT RESANY GLOBAL REVOKE
 OPTION MAXCONN 400 QUICKDSP ACCT
 SHARE RELATIVE 1300
 CONSOLE 009 3215 T DFSMS
 SPOOL 00C 2540 READER *
 SPOOL 00D 2540 PUNCH  A
 SPOOL 00E 1403 A
 MDISK 0191 3390 scyl 001 volid MR readpw writepw multipw
 LINK DFSMS 01B5 0192 RR
 LINK MAINT 0190 0190 RR
 LINK MAINT 019E 019E RR
```

Figure 28. RMS Master Virtual Machine CP Directory Entry

The user ID (RMSMASTR) in this CP directory entry must be the same as the user ID on the DFSMSRM_MASTER_VM parameter in the DFSMS/VM control file (DGTVCNTL DATA in VMSYS:DFSMS.CONTROL):

```
DFSMSRM_MASTER_VM        RMSMASTR               * USERID OF DFSMSRM MASTER
```

Directory Requirements

Consider the following directory requirements for the RMS master virtual machine:

Storage Size
Use a virtual storage size of 32MB.

Privilege Class
The RMS master virtual machine requires privilege class BG in order to support device attachment.

Machine Mode
RMS supports ESA mode.

STDEVOPT
Specifies that the optional storage device management functions are available to the RMS master virtual machine.

The sample STDEVOPT control statement uses the LIBRARY parameter to specify the RMS master is authorized to control a IBM 3495 Tape Library Dataserver, and the CTL parameter to specify that the RMS master virtual machine is authorized to issue tape library control commands. The STDEVOPT statement must precede any device statements you code in the user entry or profile. The library control parameter is required for the RMS master virtual machine.

When using the RMS master on a second level VM guest, both the RMS master user ID in the second level guest as well as the first level user ID IPLing the guest must be defined with the STDEVOPT option.

LIBRARY
Specifies whether the virtual machine is authorized to control an IBM 3495 Tape Library Dataserver.

NOCTL
Specifies that the virtual machine is *not* authorized to control a tape library. This is the default.

CTL
Specifies that the virtual machine is authorized to issue tape library control commands.

IUCV ALLOW
This option is needed to allow other virtual machines to communicate with this virtual machine.

IUCV *IDENT
Allows this virtual machine to connect to the CP system service IDENTIFY to identify local and global resource IDs.

Option MAXCONN
Specify a MAXCONN value equal to the maximum anticipated number of concurrent incoming RMS requests. Some factors to consider in arriving at this value are the number of user virtual machines that will be invoking RMS functions and the multitasking capabilities of these machines. For example, if all IBM 3495 Tape Library Dataserver mount requests for your installation are handled by a TMS virtual machine, you will need a significantly smaller MAXCONN value than you would if your end users issue RMS requests directly. Also, be sure to include storage administrators and other members of the operations staff who may be issuing RMS requests, as well as any foreign-host server machines that are defined.

QUICKDSP
Consider specifying that the virtual machine is to be added to the dispatch list immediately whenever there is work to be done, without waiting on the eligible list.

ACCT
Specify ACCT to allow the RMS master virtual machine to create accounting records.

SHARE RELATIVE
Consider specifying that this virtual machine is to receive a relative share of 1300 (or a value that meets your installation's requirements) of the system resources. (This number should be less than SFS recommendations for file pools and larger than a user relative share).

Minidisk 0191
The 0191 minidisk holds the PROFILE EXEC, which is placed on this minidisk during installation. Modify the device type, starting cylinder, and volume ID for your device.

LINK DFSMS 01B5
The RMS master virtual machine links to DFSMS 01B5 to gain access to the product code.

Appendix B. Sample ACS Routines, REXX Exit and Source Configuration Files

To assist an installation in setting up ACS processing, we are providing sample configurations, ACS routines, and a REXX Exit. The samples may be implemented to meet the installation storage management goals, or they may be modified. Or, this section may be by-passed and new routines can be written. See *z/VM: DFSMS/VM Storage Administration* for details about writing the REXX Exit and the ACS routine, and for information about the sample ACS routines and configurations.

The following sample files are located on the LOCALSMP disk (DFSMS 1C2). Refer to for more information.

Configuration		ACS Routine	
File Name	File Type	File Name	File Type
SMPCNFGL	CONFIG	SMPACSL	ACS
SMPCNFG2	CONFIG	SMPACS2	ACS
SMPCNFGS	CONFIG	SMPACSS	ACS
SMPCNFGR	CONFIG	SMPREXX	EXEC

Note: Before modifying these samples or writing new routines, it is necessary to have a thorough understanding of the chapters on management classes, configurations, ACS, ACS language reference, and the appendix on sample systems. These topics are located in the *z/VM: DFSMS/VM Storage Administration*.

Sample ACS Routine

If you choose to use the sample ACS routines, you will need to determine whether your installation uses the sample files for a small or large system. You may want to use the unmodified samples, or you may want to modify either or both the sample ACS routine and the sample ACS Source Configuration file. It should be noted that the ACS Source Configuration file contains an ACS routine.

Using the Sample ACS Routine (As Is):

- Ensure that the sample source configuration resides on an SFS file pool. In this scenario, only the sample source configuration file is needed, since it contains the base configuration information as well as the translated sample ACS routine.
- Ensure that the user ID you're logged onto has WRITE access and that the DFSMS Master has READ access to the sample source configuration file in order to activate it.
- Go into ISMF and activate the source configuration using the ISMF or DFSMS command. (See .)

Modifying the Sample ACS Routine

- Ensure that the sample source configuration file and the sample ACS routine reside on an SFS file pool.

- Ensure that the user ID you're logged onto has WRITE access and that the DFSMS Master has READ access to the sample source configuration file in order to activate it.
- Use the ISMF Automatic Class Selection (ACS) Application to:
  - Edit the sample ACS routine (ISMF or any editor)
  - Translate the modified ACS routine (ISMF)
  - Validate the ACS routine (optional)
  - Test the ACS routine (optional).

  For details on using the ACS Application, see the *z/VM: DFSMS/VM Storage Administration*.
- Go into ISMF and activate the source configuration using the ISMF or DFSMS command. (See "Step 4: Verify Space Management Configuration" on page 12.)

Modifying the Sample ACS Source Configuration File:

You may want to change only the base configuration information. This information consists of:

- A description of the configuration
- The name of the management class to be used as the default in your system
- The specification of whether to disable ACS processing when a file is created.

It should be noted that if you change the default management class, you may need to define a new management class and modify the ACS routine. See *z/VM: DFSMS/VM Storage Administration* for details on the ISMF Management Class Application. Also refer to "Modifying the Sample ACS Routine" on page 89 for instructions on modifying the ACS routine.

- Use the ISMF Configuration Application to Alter the sample base configuration information. See *z/VM: DFSMS/VM Storage Administration* for details on using the ISMF Configuration Application.
- Go into ISMF and activate the source configuration using the ISMF or DFSMS command. (See "Step 4: Verify Space Management Configuration" on page 12.)

Sample REXX Exit

If you choose to use the sample REXX exit, you may want to use the unmodified samples, or you may want to modify either or both the sample REXX Exit and the sample REXX source configuration file. You should consider compiling the REXX exit for better performance. See *z/VM: DFSMS/VM Storage Administration* for details concerning using the REXX compiler and writing the REXX exit.

Using the Sample REXX Exit

- Ensure that the sample REXX Source Configuration file resides on an SFS directory.
- Ensure that the user ID you're logged onto has WRITE access and that the DFSMS Master has READ access to the sample REXX Source Configuration file in order to activate it.
- Make modifications to the sample REXX exit, if you wish.
- Compile the REXX exit, if you wish.
- Copy the REXX exit, or the compiled version if you chose to compile the REXX Exit, into file name IGDACSMC, file type DFSMS, directory name VMSYS:DFSMS.ACSEXITS.
- Go into ISMF and activate the REXX Source Configuration file using the ISMF or DFSMS command. (See "Step 4: Verify Space Management Configuration" on page 12.)

Note that the ISMF ACS class application does not provide services to edit, translate, validate, or test the REXX exit. Also, note that the REXX Source Configuration file does not contain the REXX exit.

Modifying the Sample REXX Source Configuration File

You may want to change only the base configuration information. This information consists of:

- A description of the configuration
- The name of the management class to be used as the default in your system
- The specification of whether to disable ACS processing when a file is created.

Note that if you change the default management class, you may need to define a new management class and modify the REXX exit. See *z/VM: DFSMS/VM Storage Administration* for details on the ISMF Management Class Application.

- Ensure that the REXX Source Configuration file resides in an SFS file pool.
- Compile the sample REXX exit if you wish.
- To use the REXX exit, copy it (or the compiled version if you chose to compile the REXX exit) into file name IGDACSMC, file type DFSMS, directory name VMSYS:DFSMS.ACSEXITS.
- Use ISMF Configuration Application to alter the sample base configuration information.

  For details on using the Configuration Application, refer to the *z/VM: DFSMS/VM Storage Administration*.
- Go into ISMF and activate the REXX Configuration using the ISMF or DFSMS command. (See "Step 4: Verify Space Management Configuration" on page 12.)

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

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

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

This information may contain 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 actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrate 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Programming Interface Information

This book primarily documents information that is NOT intended to be used as Programming Interfaces of z/VM.

This book also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of z/VM. This information is identified where it occurs by an introductory statement to the chapter or section.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on IBM Copyright and trademark information (https://www.ibm.com/legal/copytrade).

Adobe is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

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

Terms and Conditions for Product Documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal Use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see:

• The section entitled **IBM Websites** at IBM Privacy Statement (https://www.ibm.com/privacy)

• Cookies and Similar Technologies (https://www.ibm.com/privacy#Cookies_and_Similar_Technologies)

Bibliography

This topic lists the publications in the z/VM library. For abstracts of the z/VM publications, see *z/VM: General Information*.

Where to Get z/VM Information

The current z/VM product documentation is available in IBM Documentation - z/VM (https://www.ibm.com/docs/en/zvm).

z/VM Base Library

Overview

- *z/VM: License Information*, GI13-4377
- *z/VM: General Information*, GC24-6286

Installation, Migration, and Service

- *z/VM: Installation Guide*, GC24-6292
- *z/VM: Migration Guide*, GC24-6294
- *z/VM: Service Guide*, GC24-6325
- *z/VM: VMSES/E Introduction and Reference*, GC24-6336

Planning and Administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6261
- *z/VM: CMS Planning and Administration*, SC24-6264
- *z/VM: Connectivity*, SC24-6267
- *z/VM: CP Planning and Administration*, SC24-6271
- *z/VM: Getting Started with Linux on IBM Z*, SC24-6287
- *z/VM: Group Control System*, SC24-6289
- *z/VM: I/O Configuration*, SC24-6291
- *z/VM: Running Guest Operating Systems*, SC24-6321
- *z/VM: Saved Segments Planning and Administration*, SC24-6322
- *z/VM: Secure Configuration Guide*, SC24-6323

Customization and Tuning

- *z/VM: CP Exit Customization*, SC24-6269
- *z/VM: Performance*, SC24-6301

Operation and Use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6260
- *z/VM: CMS Primer*, SC24-6265
- *z/VM: CMS User's Guide*, SC24-6266
- *z/VM: CP Commands and Utilities Reference*, SC24-6268

- *z/VM: System Operation*, SC24-6326
- *z/VM: Virtual Machine Operation*, SC24-6334
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6337
- *z/VM: XEDIT User's Guide*, SC24-6338

Application Programming

- *z/VM: CMS Application Development Guide*, SC24-6256
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6257
- *z/VM: CMS Application Multitasking*, SC24-6258
- *z/VM: CMS Callable Services Reference*, SC24-6259
- *z/VM: CMS Macros and Functions Reference*, SC24-6262
- *z/VM: CMS Pipelines User's Guide and Reference*, SC24-6252
- *z/VM: CP Programming Services*, SC24-6272
- *z/VM: CPI Communications User's Guide*, SC24-6273
- *z/VM: ESA/XC Principles of Operation*, SC24-6285
- *z/VM: Language Environment User's Guide*, SC24-6293
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-6295
- *z/VM: OpenExtensions Callable Services Reference*, SC24-6296
- *z/VM: OpenExtensions Commands Reference*, SC24-6297
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-6298
- *z/VM: OpenExtensions User's Guide*, SC24-6299
- *z/VM: Program Management Binder for CMS*, SC24-6304
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-6313
- *z/VM: REXX/VM Reference*, SC24-6314
- *z/VM: REXX/VM User's Guide*, SC24-6315
- *z/VM: Systems Management Application Programming*, SC24-6327
- *z/VM: z/Architecture Extended Configuration (z/XC) Principles of Operation*, SC27-4940

Diagnosis

- *z/VM: CMS and REXX/VM Messages and Codes*, GC24-6255
- *z/VM: CP Messages and Codes*, GC24-6270
- *z/VM: Diagnosis Guide*, GC24-6280
- *z/VM: Dump Viewing Facility*, GC24-6284
- *z/VM: Other Components Messages and Codes*, GC24-6300
- *z/VM: VM Dump Tool*, GC24-6335

z/VM Facilities and Features

Data Facility Storage Management Subsystem for z/VM

- *z/VM: DFSMS/VM Customization*, SC24-6274
- *z/VM: DFSMS/VM Diagnosis Guide*, GC24-6275
- *z/VM: DFSMS/VM Messages and Codes*, GC24-6276
- *z/VM: DFSMS/VM Planning Guide*, SC24-6277

- *z/VM: DFSMS/VM Removable Media Services*, SC24-6278
- *z/VM: DFSMS/VM Storage Administration*, SC24-6279

Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6281
- *z/VM: Directory Maintenance Facility Messages*, GC24-6282
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6283

Open Systems Adapter

- Open Systems Adapter/Support Facility on the Hardware Management Console (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC14-7580-02.pdf), SC14-7580
- Open Systems Adapter-Express ICC 3215 Support (https://www.ibm.com/docs/en/zos/2.3.0?topic=osa-icc-3215-support), SA23-2247
- Open Systems Adapter Integrated Console Controller User's Guide (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/SC27-9003-02.pdf), SC27-9003
- Open Systems Adapter-Express Customer's Guide and Reference (https://www.ibm.com/docs/en/SSLTBW_2.3.0/pdf/ioa2z1f0.pdf), SA22-7935

Performance Toolkit for z/VM

- *z/VM: Performance Toolkit Guide*, SC24-6302
- *z/VM: Performance Toolkit Reference*, SC24-6303

The following publications contain sections that provide information about z/VM Performance Data Pump, which is licensed with Performance Toolkit for z/VM.

- *z/VM: Performance*, SC24-6301. See z/VM Performance Data Pump.
- *z/VM: Other Components Messages and Codes*, GC24-6300. See Data Pump Messages.

RACF® Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6305
- *z/VM: RACF Security Server Command Language Reference*, SC24-6306
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6307
- *z/VM: RACF Security Server General User's Guide*, SC24-6308
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6309
- *z/VM: RACF Security Server Messages and Codes*, GC24-6310
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6311
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6312
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6324

Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6316
- *z/VM: RSCS Networking Exit Customization*, SC24-6317
- *z/VM: RSCS Networking Messages and Codes*, GC24-6318
- *z/VM: RSCS Networking Operation and Use*, SC24-6319
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6320

TCP/IP for z/VM

- *z/VM: TCP/IP Diagnosis Guide*, GC24-6328
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6329
- *z/VM: TCP/IP Messages and Codes*, GC24-6330
- *z/VM: TCP/IP Planning and Customization*, SC24-6331
- *z/VM: TCP/IP Programmer's Reference*, SC24-6332
- *z/VM: TCP/IP User's Guide*, SC24-6333

Prerequisite Products

Device Support Facilities

- Device Support Facilities (ICKDSF): User's Guide and Reference (https://www.ibm.com/docs/en/SSLTBW_2.5.0/pdf/ickug00_v2r5.pdf), GC35-0033

Environmental Record Editing and Printing Program

- Environmental Record Editing and Printing Program (EREP): Reference (https://www.ibm.com/docs/en/SSLTBW_2.5.0/pdf/ifc2000_v2r5.pdf), GC35-0152
- Environmental Record Editing and Printing Program (EREP): User's Guide (https://www.ibm.com/docs/en/SSLTBW_2.5.0/pdf/ifc1000_v2r5.pdf), GC35-0151

Related Products

XL C++ for z/VM

- *XL C/C++ for z/VM: Runtime Library Reference*, SC09-7624
- *XL C/C++ for z/VM: User's Guide*, SC09-7625

z/OS

IBM Documentation - z/OS (https://www.ibm.com/docs/en/zos)

Index

IBM®

Product Number: 5741-A09

Printed in USA