Directory Maintenance VM/ESA

Tailoring and Administration Guide

Release 5.0

SC23-0533-05

Directory Maintenance VM/ESA

Tailoring and Administration Guide

Release 5.0

SC23-0533-05

Note:

Before using this information and the product it supports, read the information in “Notices” on page 237.

| **Sixth Edition (February 2001)**

This edition applies to Version 1, Release 5, Modification 0 of IBM® Directory Maintenance (DirMaint VM/ESA®) (product number 5748-XE4) and to all subsequent releases and modifications until otherwise indicated in new editions.

| This edition replaces SC23-0533-04.

Contents

Preface

This book tells you how to tailor and administer your system using IBM Directory Maintenance VM/ESA.

The terminology used in this book is as follows:

DirMaint A abbreviation for the program name, (IBM Directory Maintenance VM/ESA).

DIRMAINT Default user ID of the disconnected service machine that owns and maintains the VM source directory. This default name is subject to customer tailoring.

DIRM Minimum abbreviation of the DIRMAINT command used to send a transaction to the DIRMAINT service machine when entered from the CMS command line. When issued from within a program, the full command name of DIRMAINT should be used, usually preceded by the keyword EXEC.

IBM Directory Maintenance VM/ESA (DirMaint) is a Conversational Monitor System (CMS) application that helps you manage your VM directory. Directory management is simplified by the DirMaint command interface and automated facilities. The DirMaint directory statement-like commands initiate directory transactions. The DirMaint error checking ensures that only valid changes are made to the directory, and that only authorized personnel are able to make the requested changes. Any transaction requiring the allocation or deallocation of minidisk extents can be handled automatically. All user initiated transactions can be password controlled and can be recorded for auditing purposes.

The DirMaint functions are performed by two disconnected virtual machines equipped with an automatic restart facility. The DIRMAINT virtual machine owns and manages the directory, and the DATAMOVE virtual machine performs the copying and formatting of CMS minidisks. Users invoke DirMaint functions by submitting commands to the DIRMAINT virtual machine. Large systems may have multiple DATAMOVE machines.

Except for the documented exit routines, this information should not be used for programming purposes. DirMaint provides a safe, efficient, and interactive way to maintain the VM/ESA user directory. You can manage the directory with DIRMAINT through the use of commands. Thus, with DIRMAINT, you avert errors that are often made during direct updating of the directory source file. You can also use DirMaint to audit the security of relevant tasks that it performs. DirMaint maintains the CP directory for the following VM control programs:

- Virtual Machine/Enterprise System Architecture 370 Feature Version 1 Release 1.5, Program Number 5664-112.
- Virtual Machine/Enterprise System Architecture Version 1 Release 2.1 or later, Program Number 5664-112.
- Virtual Machine/Enterprise System Architecture Version 2 Release 1, Program Number 5654-030.

The DirMaint product may be used in with:

- Resource Access Control Facility (RACF), Program Number 5740-XXH, Version 1.9.2.
- Other External Security Managers (ESMs) providing equivalent interfaces for password verification and audit recording; and optionally providing equivalent function for user enrollment and disenrollment, resource registration and removal, and resource authorization checking.

Who Should Read This Book

This book is intended for anyone responsible for tailoring, planning, updating, and maintaining the DirMaint product.

What You Should Know before Reading This Book

You should know about the purpose, structure, and contents of the system directories and how they can be used. This knowledge should also include the understanding of VM/ESA.

What This Book Contains

This book contains information on:

Chapter 1, “Introduction” on page 1 provides a brief description of DirMaint, and an overview of the tailoring tasks.

Chapter 2, “Directory Entries for the DirMaint Machines” on page 5, provides guidance for defining the DirMaint service machines to CP and migration from an earlier version of DirMaint.

Chapter 3, “Tailoring the DIRMAINT Service Machine” on page 39, provides guidance for tailoring the various data files used by the DIRMAINT service machine or migrating those files from a prior release of DirMaint Release 5.0.

Chapter 4, “Tailoring the DATAMOVE Service Machine” on page 75, provides information for starting the DATAMOVE service machine to format and copy minidisks.

Chapter 5, “Tailoring the DIRMSAT Service Machine” on page 81, provides guidance for bringing up the DirMaint satellite service machines.

Chapter 6, “DASD Management” on page 87, provides information for using DirMaint to perform DASD administration.

Chapter 7, “User Tailoring” on page 119, provides examples of commands through various exits and data files that are subject to tailoring in the user's virtual machine.

Chapter 8, “Delegating Administrative Authority” on page 135, provides information on delegating administrative authority.

Chapter 9, “Exit Routines” on page 141, provides information on how to use and tailor the DirMaint supplied exit routines.

Chapter 10, “Planning for Diagnosis” on page 197, provides an overview of the items to consider when diagnosing DirMaint problems.

The following are appendixes which include the format of the WAKEUP Times File, ESM considerations, a security self-assessment checklist and performance considerations.

Appendix A, “External Security Manager Considerations” on page 199, provides recommendations to follow that will help you improve your system security and integrity.

Appendix B, “Tuning DirMaint Performance” on page 209, describes some CP commands that may assist ypu in enchancing DirMaint performance.

Appendix C, “DirMaint Configuration Data Files” on page 215, provides a summary of each of the CONFIG* DATADVH entries provided by DirMaint Release 5.0.

Appendix D, “WAKEUP Command” on page 233, describes the format of the WAKEUP Times File.

Where to Find More Information

For a list of the books that can provide you with additional information on DirMaint and z/VM, see “Bibliography” on page 241.

How to Send Your Comments to IBM

Your feedback is important in helping us to provide the most accurate and high-quality information. If you have comments about this book or any other VM documentation, send your comments to us using one of the following methods. Be sure to include the name of the book, the form number (including the suffix), and the page, section title, or topic you are commenting on.

- Visit the z/VM web site at:

  `http://www.vm.ibm.com/related/dirmaint`

  There you will find the feedback page where you can enter and submit your comments.
- Send your comments by electronic mail to one of the following addresses:

  Internet: pubrcf@vnet.ibm.com

  IBMLink™: GDLVME(PUBRCF)
- Fill out the Readers' Comments form at the back of this book and return it using one of the following methods:
  - Mail it to the address printed on the form (no postage required in the USA).
  - Fax it to 1-607-752-2327.
  - Give it to an IBM representative.

Chapter 1. Introduction

IBM Directory Maintenance VM/ESA (DirMaint) is a CMS application that helps top manage your VM directory. Directory statements can be added, deleted, or altered using the DirMaint directory statement-like commands. DirMaint provides automated validation and extent allocation routines to reduce the chance of operator error.

Getting Started

Before you can use DirMaint, you must install and customize it. Other than giving the DIRMAINT server your existing source directory file for initialization, no other tailoring, customization, or modification is required.

This book provides information to help you tailor DirMaint to suit your installations needs. Tailoring and administration involves setting up, configuring, and modifying DirMaint. The tailoring and administration tasks are:

Installation.
DirMaint works for and with other products that have certain requirements about the way you define DirMaint depends on the requirements associated with how it works for and with other products. Planning for those requirements is essential to installing DirMaint.

DirMaint is installed using the Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E) component of VM/ESA. For more information on installation planning considerations, see the *DirMaint Program Directory*.

Customization and Modification.
DirMaint must be adapted to work in your environment to handle your particular needs; to do so, you may need the IBM-supplied code. You must understand what DirMaint needs to perform the basic functions you desire, what options it offers, and what implications those options may have in your environment. These changes should be made by your system programmer.

Migration from an earlier version of DirMaint.
By knowing what is required to migrate from another release of DirMaint, you can plan when it will be done, how long it will take, who should do it, and which of the new features you should install. For DirMaint Release 5.0, you can use the VMSES/E component to migrate to the new release, with or without replacing an earlier release.

Tailoring.
By knowing what is required to tailor your data files, you can plan when the tailoring should be done, how long it should take, who should do it, and which of the new features you should implement.

Operation.
By understanding what DirMaint requires for operation, you can determine how it should be managed within your network. You may have to decide who will be handling the operations (system operators or automated procedures) and whether any training will be required.

Administration.
Knowing what DirMaint functions will be available to your users should help you determine what administrative tasks to perform. For example, will users be responsible for operational tasks on remote devices? Will users need to identify themselves on remote systems to which they submit jobs?

Diagnosis.
Problems are not always with the DirMaint product. They may be with communications lines or network connections with other systems. How you plan to handle problem situations and follow-up diagnosis at your installation can help speed recovery and save time.

Using DirMaint Commands

DirMaint runs on z/VM, which is an interactive, multiple-access operating system. Interactive means two-way communication between users and the system. Multiple-access means many people can use a z/VM system at the same time. Therefore, productivity is increased by sharing data more quickly and easily between you and other users.

The Control Program, usually referred to as CP, is a component of VM/ESA that manages the resources of a single computer so that multiple computing systems appear to exist. When you are working in the CP environment, you are provided with processor functions, input and output devices, and processor storage.

The Conversational Monitor System, usually referred to as CMS, performs two roles; as an end-user interface, it is the part of z/VM that is most often seen by your users. It is also the part of the operating system that supports the running of your programs, thus, it is your application programming interface.

Entering Commands

When you enter a command, VM/ESA must be able to recognize that it is a DirMaint command. As the following command shows, **dirm** must precede the command name.

Example—Entering a DirMaint Command: Enter:

```
dirm account ?
```

Where

dirm
Indicates a DirMaint command.

account
DirMaint command to be entered.

? Indicates the command parameters.

Note: When entering commands from the console of the service machines: DIRMAINT, DIRMSAT, and DATAMOVE, you must omit the keyword **dirm**.

For more information on entering commands, see *Directory Maintenance VM/ESA: Command Reference*.

Using the DirMaint HELP Facility

You can receive online information using the DirMaint HELP Facility. For example, to display a menu of DirMaint HELP information, enter:

```
DIRM HELP
```

Place the cursor under a command or topic you want information about and press Enter.

To display information about a specific DirMaint operand (ADD in this example), enter:

```
DIRM HELP ADD
```

or

```
DIRM HELP AD
```

or

```
DIRM HELP A
```

The DirMaint HELP facility recognizes the minimum abbreviation for a DIRMAINT operand.

To display information about the DIRMAINT command, enter:

```
DIRM HELP DIRM
```

You can also display information about a message (DVH1093 in this example), by entering:

```
DIRM HELP DVH1093
```

The DIRMAINT HELP command and its operands are described in *Directory Maintenance VM/ESA: Command Reference*.

Chapter 2. Directory Entries for the DirMaint Machines

This chapter describes defining the DirMaint service machines to CP.

Before you use a virtual machine, you must know how to communicate with the operating system you are going to run in your virtual machine. DirMaint involves using several virtual machines with each performing different tasks. The following sections describe the tasks you must perform to prepare to run DirMaint on a VM/ESA system.

DirMaint Install and Service User ID

The DirMaint install and service user ID, P748XE4M by default, owns:

- All DASD space containing IBM-supplied DirMaint product code
- Customer tailored files
- Customized exit routines.

Other local modifications to the product should also be located on disks owned by this user ID. All of these disks are maintained using the VM installation and service tool, VMSES/E.

MAINT User ID

The general user needs to access two product files: DIRMAINT EXEC and ACCESS DATADVH. These files usually reside on the MAINT 19E disk, the system Y-disk, and are copied from the P748XE4M 29E disk, a DirMaint production disk.

What is a Server?

A server in VM provides shared services to VM. The DirMaint product provides shared services to users with these servers:

Server Type	Description
DIRMAINT	The primary server; the DIRMAINT server handles all aspects of source directory manipulation and controls the actions of all other servers. There is only one DIRMAINT server.
DATAMOVE	A DATAMOVE server is responsible for manipulating minidisks on behalf of the DIRMAINT server. These tasks can include formatting, copying, and cleaning. There can be multiple copies of the DATAMOVE server.
DIRMSAT	The DIRMSAT server is responsible for manipulating the object directory on systems other than the system the primary DIRMAINT server is on, or on the same system if maintaining duplicate copies of the object. There may be multiple copies of the DIRMSAT server.

Notes:

1. For maximum integrity, duplicate satellite servers can be used on each system to maintain duplicate object directories.
2. It is not necessary to define all of these virtual machines, only the P748XE4M and the DIRMAINT user IDs are required.
3. If you choose not to define the DATAMOVE or DIRMSAT service machines now, you will lose the function they provide.

   You may define one or more DATAMOVE or DIRMSAT machines during installation and may define additional machines later, or you may remove them if they are no longer needed.

   For more information on the changes you will need to make to the product configuration definition files when you add or remove a DATAMOVE server, see Chapter 4, “Tailoring the DATAMOVE Service Machine” on page 75 and when you add or remove a DIRMSAT server, see Chapter 5, “Tailoring the DIRMSAT Service Machine” on page 81.
4. The DIRMAINT, DATAMOVE and DIRMSAT service machines must have access to the DirMaint service machine code. You must access:
   - P748XE4M 491 disk, for production use
   - P748XE4M 492 disk, for new service or modifications.

   The service machines will share this single disk by linking as their own 191 disk.

DirMaint Service Machine

The directory maintenance service machine user ID, DIRMAINT by default:

- Owns the CP source directory
- Receives transactions from authorized users
- Verifies that the transactions are valid
- Makes the appropriate updates to the source directory.

To place directory changes online and log the results, the directory maintenance service machines use the DIRECT command (for a VM/ESA 370 feature system) and the DIRECTXA command (for an VM/ESA feature system).

The service machine optionally:

- Monitors the directory for user IDs' passwords that have not been changed during installation
- Controls allocation of DASD space to user virtual machines.

If full DASD services are enabled, the server virtual machine:

- Allocates work among one or more DASD service machines
- Monitors the progress of each machine.

In a multiple system cluster, the DIRMAINT service machine will:

- Notify the satellite service machines in the cluster whenever an update is made to the source directory so the satellite servers can put the corresponding update online on the other systems in the cluster
- Run on any system in the cluster; however, it can only run on one system in the cluster at a time.
- Maintain a duplicate copy of the source directory on a second disk.

DASD Utility Service Machine

If a command changes any DASD space allocations, the productivity of the system administrator and the end user can be further improved by the DASD utility service machines.

The DASD servers, which have a user ID of DATAMOVE by default:

- Format newly allocated DASD space for the user with an optional user-specified minidisk label or block size.
- Format a new extension to receive files from an existing disk, copy files from an existing disk to the new extent, optionally with user-specified disk label or block size, and optionally format the old extent to prevent exposure of residual data to the next user who acquires the space.
- Format an old extension being deallocated again to prevent exposure of any residual data to the next user.

If the workload warrants, additional DASD server machines may be defined and the work divided among them.

In a multiple-system cluster, the DirMaint service machine will:

- Assign each DASD server to handle a specific system affinity
- Allocate work to the various DASD servers

For more information on how to determine the estimated amount of physical space required on DASD for storing files, see the *DirMaint Program Directory*.

Cluster Satellite Synchronization Server Machine

The DIRMSAT server is responsible for manipulating the object directory on systems other than the system on which the primary DIRMAINT server runs. DIRMSAT can also manipulate the object directory on the same system if it is maintaining duplicate copies of the object directory. There may be multiple copies of the DIRMSAT server.

In a multiple-system cluster, the satellite service machine, user ID DIRMSAT, by default will:

1. Receive notifications from the DirMaint server whenever the directory has been updated and put online by the DirMaint machine on the system where DirMaint is running.
2. Then place the update online on its own system to keep the object directories synchronized.

Administrative and Support Users

By entering commands, administrative users can initiate changes to the source directory. The system programmer and other members of the support staff are responsible for maintaining the DirMaint configuration. This is usually done by invoking the DIRMAINT EXEC, which sends the transaction to the DirMaint service machine. Generally, users do not have the authority to make changes to the DirMaint operational configuration; however, they may have the authority to make changes to the source directory.

To link and access the DirMaint user component, enter:

```
DIRMAINT EXECLOAD
```

General Users

Your installation can determine which DirMaint commands are available to users. This allows the user to make some changes to the directory without having to involve the system administrator, improving the productivity of both. A subset of the DIRMAINT commands may be used by the general user community. For more information about commands, see *Directory Maintenance VM/ESA: Command Reference*.

Common PROFILE

A PROFILE statement defines the start of a profile entry in the source directory. When PROFILE is specified, all profile entry definitions must follow the last DIRECTORY statement and the global definition section and precede the first USER statement. There are no restrictions on the number of profile entries that may be specified.

When the directory control statements are used in a profile definition, they perform the same function as they do when used in a user definition. However, they may operate differently when used in both the profile and user definitions.

Directory profiles are implemented by defining a profile entry in the source directory and specifying an INCLUDE statement in the user entries that refer to the profile. The DIRECT or DIRECTXA commands will run in less time if your system creates a PROFILE.

If you use a different name for your common profile, you will need to change the name on the INCLUDE statements for each sample directory shown.

If you choose not to use a profile, you will need to include the statements as shown in “Common PROFILE Example” on page 10 (excluding the PROFILE statement) in each sample directory shown.

Common PROFILE Example

```
1 PROFILE name
   2 IPL CMS PARM AUTOCR
   3 MACHINE ESA
   4 CONSOLE 009 3215
   5 SPOOL   00C 2540 READER *
   5 SPOOL   00D 2540 PUNCH  A
   5 SPOOL   00E 1403 A
   6 LINK MAINT    190 190 RR
   6 LINK MAINT    19D 19D RR
   6 LINK MAINT    19E 19E RR
```

Figure 1. Common PROFILE Example

These notes will help you with the example shown in Figure 1.

1 Specifies *name* of the PROFILE.

Where:

name
Is a 1 to 8 character alphanumeric string. A valid character is any character that can be used in a user ID name. Only one profile name may be specified on a PROFILE statement. The name assigned to the profile entry references the profile. A suggested profile name is COMMON.

2 Specifies CMS to be IPLed when the user ID is logged on. AUTOCR avoids a VM READ when the VM/ESA message is issued by CP.

3 Specifies the machine type to be used.

`MACHINE ESA, MACHINE XA, and MACHINE XC`

are all acceptable for use by any of the DIRMAINT service machines: DIRMAINT, DATAMOVEs, and DIRMSATs. MACHINE 370 is also acceptable for use by any of the DIRMAINT service machines; unless prohibited by the level of CMS in use.

The MACHINE statement:

- Is optional on a VM/ESA feature system, if:
  - The virtual machines that include the profile run in VM/ESA 370 mode.
  - The virtual machine directory entries include a MACHINE statement.
  - A VM/ESA Version 2 Release 1.0 system uses a GLOBALOPTS MACHINE directory statement to set the default for all virtual machines.
- Is not valid on a VM/ESA 370 feature system

4 Specifies the console device of the virtual machine

5 Specifies the reader, punch, and printer devices of the virtual machine.

6 Specifies links to CMS system disks.

Directory Statements for the DIRMAINT Virtual Machine

The directory statement examples are divided into two parts, with two alternatives for each part. The first part contains all of the non-DASD directory statements, with alternatives for 370 feature systems and ESA feature systems. The second part contains all of the DASD directory statements, with alternatives for new customers and migrating customers.

For a new customer, IBM recommends using the default of DIRMAINT as the user ID of the directory management service machine.

For a migrating user, you can:

- Rename your existing DIRMAINT server, for example, use **DIRMR4** and reuse the DIRMAINT user ID for DirMaint Release 5.
- Use a different user ID, for example, use **DIRMR5** for the new release and keep your existing DIRMAINT user ID for your old release.
- Use the DIRMAINT user ID for both releases, and allow them to coexist throughout the migration period.

Notes:

1. If you are a migrating customer and want to keep the two releases on separate virtual machines, use the example for new customers.
2. If you are defining DIRMAINT in a multiple-system CSE cluster, IBM recommends that you define DIRMAINT on all systems in the cluster so that it can be brought up on an alternate system if the primary system is unavailable. The use of shared spool files is recommended. Neither input nor output spool files of DIRMAINT should be placed on the CSE exclusion list.

DIRMAINT Non-DASD Directory Statements for 370 Feature Systems

```
USER DIRMAINT 1  NOLOG 2  16M 3  16M 3  BG 4  64 5
     6 INCLUDE COMMON
     7 ACCOUNT SYSTEM SYSPROG
     8 IUCV ANY PRIORITY MSGLIMIT 100
     9 OPTION BMX CONCEAL ECMODE REALTIME
```

Figure 2. DIRMAINT Non-DASD Directory Statements for 370 Feature Systems

These notes will help you with the example shown in Figure 2.

1 Identifies the user ID of the DIRMAINT virtual machine.

2 Identifies the password for the DIRMAINT virtual machine.

3 Specifies the required virtual storage size. DIRMAINT will function on as little as 6MB of storage for small source directories. IBM recommends using the largest virtual machine size available. If you chose to use a default size less than 16MB, you should leave the maximum size at 16MB.

4 Specifies the privilege classes based on VM-provided defaults.

This virtual machine is authorized to use:

- Privilege class B; this is required to do the following:

  - Allows DIRMAINT to suppress CP message headers, by using the CP MSGNOH command.
  - Allows DIRMAINT to issue the DIAGNOSE code X'84'.
  - Allows DIRMAINT to issue the DIAGNOSE code X'3C'.
  - Allows DIRMAINT to issue the DIAGNOSE code X'A0' for these conditions:
    - RACF® installed and controlling passwords; DIRMAINT must be authorized to issue subcode X'04' of DIAGNOSE code X'A0'.
    - An ESM other than RACF; DIRMAINT must be authorized to use the ESM interface for password verification. This may issue subcode X'04' of DIAGNOSE code X'A0'.
  - Allows DIRMAINT to issue the DIAGNOSE code X'D4' for SECLABEL use. With SECLABEL support and Mandatory Access Control enabled for spool files, DIRMAINT must be authorized to use DIAGNOSE code X'D4'.

    Note: SECLABEL support and MAC for spool files must be enabled if your system needs to comply with the TCSEC for a class B1 Trusting Computing base (TCB).

  Note: By default, these are all CP privilege class B functions. If your site has changed these to other privilege classes, then DIRMAINT must be authorized for the privilege classes containing these functions.

- Privilege class G; this is for general user commands.

5 Specifies the dispatching priority for the DIRMAINT virtual machine.

6 Specifies the directory PROFILE

7 Specifies the appropriate account to charge for the DIRMAINT virtual machine installation, and to route the printed output to the system programmer responsible for DirMaint operation.

8 Specifies the IUCV statement. This is optional unless your system is running with RACF or another ESM, and chooses to record commands received by DIRMAINT and messages sent by DIRMAINT in the ESM audit log. If ESM audit logging is enabled, an initial value of 100 is suggested for the MSGLIMIT.

Note: The ESM audit logging capability must be enabled if your system needs to comply with the TCSEC for either a class C2 or class B1 TCB.

9 Specifies directory options required by the DIRMAINT user ID.

DIRMAINT Non-DASD Directory Statements for ESA Feature Systems

```
USER DIRMAINT 1 NOLOG 2 16M 3 16M 3 BDG 4
     5 INCLUDE COMMON
     6 ACCOUNT SYSTEM SYSPROG
     7 D8ONECMD FAIL LOCK
     8 IUCV ANY PRIORITY MSGLIMIT 100
     9 OPTION CONCEAL D84NOPAS IGNMAXU
```

Figure 3. DIRMAINT Non-DASD Directory Statements for ESA Feature Systems

These notes will help you with the example shown in Figure 3:

1 Identifies the user ID of the DIRMAINT virtual machine.

2 Identifies the password for the DIRMAINT virtual machine.

3 Specifies the required virtual storage size; DIRMAINT will function on as little as 6MB of storage for small source directories. IBM recommends using the largest virtual machine size available. If you chose to use a default size less than 16MB, you should leave the maximum size at 16MB.

4 Specifies the privilege classes based on VM-provided defaults.

This virtual machine is authorized to use:

- Privilege class B; this is required to do the following:
  - Allows DIRMAINT to suppress CP message headers by using the CP MSGNOH command.
  - Allows DIRMAINT to issue the DIAGNOSE code X'84'.
  - Allows DIRMAINT to issue the DIAGNOSE code X'3C'.
  - Allows DIRMAINT to issue the DIAGNOSE code X'A0' for these conditions:
    - RACF installed and controlling passwords; DIRMAINT must be authorized to issue subcode X'04' of DIAGNOSE code X'A0'.
    - An ESM other than RACF; DIRMAINT must be authorized to use the ESM interface for password verification. This may issue subcode X'04' of DIAGNOSE code X'A0'.
  - Allows DIRMAINT to issue the DIAGNOSE code X'D4' for SECLABEL use. With SECLABEL support and Mandatory Access Control (MAC) enabled for spool files, DIRMAINT must be authorized to use DIAGNOSE code X'D4'.

    Note: SECLABEL support and MAC for spool files must be enabled if your system needs to comply with the TCSEC for a class B1 TCB.

  Note: By default, these are all CP privilege class B functions. If your site has changed these to other privilege classes, then DIRMAINT must be authorized for the privilege classes containing these functions.
- Privilege class D; this is required to issue the CP QUERY ALLOC command for displaying the number of cylinders or pages that are allocated, in use, and available for DASD volumes attached to the system. The DIRMAINT server will then map these as used extents.

  Note: There are risks associated with granting a user ID class D authority. You may want to create a separate class for the CP QUERY command granting only a few user IDs authorization.
- Privilege class G; this is for general user commands.

5 Specifies the directory PROFILE

6 Specifies the appropriate account to charge for the DIRMAINT virtual machine installation, and to route the printed output to the system programmer responsible for DirMaint operation.

7 The D8ONECMD statement is an optional statement. This ensures all DIRM CP and DIRM CMS commands are properly audited.

8 The IUCV statement is optional unless your system is running with RACF or another ESM, and chooses to record commands received by DIRMAINT and messages sent by DIRMAINT in the ESM audit log. If ESM audit logging is enabled, an initial value of 100 is suggested for the MSGLIMIT.

Note: The ESM audit logging capability must be enabled if your system needs to comply with the TCSEC for either a class C2 or class B1 TCB.

9 Specifies the directory options required by the DIRMAINT user ID.

The CONCEAL option tells CP to automatically restart the DIRMAINT machine if certain error conditions occur.

The D84NOPAS option is optional for a standalone system without an ESM, or for a standalone system with RACF as the ESM. It may also be optimal for systems with other ESMs. This is recommended for systems in a multiple system cluster. It is required in a multiple system cluster for any directory entries containing the SYSAFFIN EXIST_AT, SYSAFFIN LOGON_AT, or SYSAFFIN NOLOG_AT statements.

The IGNMAXU option allows the DIRMAINT machine to LOGON or to be AUTOLOGged on, even if your installation has reached a limit on the maximum number of users allowed on your system when this limit is already exceeded.

DIRMAINT DASD Directory Statements for New Customers

```
1  LINK P748XE4M 491 591 M  - Product code, primary.
2  LINK P748XE4M 492 592 M  - Product code, alternate.
3  LINK P748XE4M 11F 11F M  - Interface code, primary.
4  LINK P748XE4M 41F 21F M  - Interface code, alternate.
5  LINK MAINT    123 123 MW - Object directory disk.
6  MDISK 155 MR - Read/write scratch space, A-disk.
7  MDISK 1FA MR - Spool file staging space, Z-disk.
8  MDISK 1DF MR - Primary directory files.
9  MDISK 2DF MR - Optional secondary directory files.
10 MDISK 1AA MR - Primary transaction history files (optional).
11 MDISK 2AA MR - Optional secondary history files.
12 MDISK 1DB MR - Primary directory backup (optional).
13 MDISK 1DE MR - Directory edit scratch disk.
14 MDISK 15D RR - Intersystem locking disk.
```

Figure 4. DIRMAINT DASD Directory Statements for New Customers

The Dirmaint product code and data files should be on conventional minidisk space rather than SFS space. However, any of these DIRMAINT MDISKS, except the 15D disk, may in fact reside in a directory in a shared file system.

Note: Although this is allowed, it is not recommended. It would be possible to get into a situation where the shared file servers are out of available DASD space and are not operational; DirMaint would be unable to allocate additional DASD space because the SFS severs are not operational.

If you choose to use shared file pool space, IBM recommends that a separate DIRCONTROL directory be used in the place of each *disk*. You will need to enter GRANT AUTHORITY command to all of the DIRMSAT for READ access (or READ and NEWREAD access if you chose to use FILECONTROL directories) to the directories that replace the 1DF and 2DF disks. You will need to update the DVHPROFA DIRMAINT and the DVHPROFA * files for each of the DIRMSAT machines accordingly. For more information, see “DVHPROFA DIRMAINT” on page 43

Ensure that access to minidisks is controlled by either passwords or explicit link authorization, as determined by the minidisk owner. Minidisk passwords are now optional for controlling minidisk directory links. None of the DirMaint MDISKs should have any directory passwords (except for the 11F disk, which must have a read password of ALL, or if an ESM is installed, must be defined as UACC (READ) or PUBLIC (READ)). Any virtual machine needing access to these disks should do so through a LINK directory statement. Using passwords on any of these disks increases the risk of unauthorized access to your system.

Access to backup tapes of any of these disks must be carefully controlled to prevent unauthorized access to your system.

The logon password of any virtual machine with either read or write access to any of these disks or with access to the backup tapes for any of these disks must be carefully chosen and controlled, to minimize the risk of unauthorized access to your system. DirMaint also supports control of minidisk links by an ESM. The MDISK addresses shown in Figure 4 are arbitrary. However, if any of these addresses are changed, you must make corresponding changes to several data files. The

addresses shown in “DIRMAINT DASD Directory Statements for New Customers” have been chosen to provide a mnemonic association between the address and the purpose for which the disk is used.

These notes will help you with the example shown in Figure 4 on page 15.

1 **- 4** **and 6 - 12** If you chose to locate any of the MAINT or P748XE4M “disks” being linked by the DIRMAINT machine into shared file system space, omit the LINK or MDISK statements for them from the directory entry of the DIRMAINT machine.

LINK statements for the MAINT disks have been omitted from the directory entry of the DIRMAINT machine. If they are not contained in the included PROFILE, they will have to be added here. If you have installed the optional national language HELP files, you should also include a LINK statement for those disks, either in the directory entry of the DIRMAINT machine or in the included profile.

5 The choice of 123 as the address of the object directory disk is arbitrary. It must match the address used on the DIRECTORY statement in the USER INPUT file. The PLANINFO file shows this as a link to the MAINT 123 disk. The DirMaint Release 4 documentation uses BC4. In practice, it appears that most customers use either the real system residence volume disk address (the system IPL address) or the virtual address used by the MAINT user ID to refer to the system residence volume (usually 123).

The use of link mode MW on the object directory disk is correct. This is necessary for the MAINT user ID to update the CP nucleus on the system residence volume without shutting DIRMAINT down. This is one of the rare situations where MW is appropriate for use with a CMS application program.

6 Specifies the read/write scratch space, A-disk.

7 Specifies the spool file staging space, Z-disk.

8 The 1DF **8** and 1AA **10** disks may, but need not reside on a single DASD volume, possibly the same volume or volumes as the 191, 11F, and 19E disks. Also, the 2DF **9** and 2AA **11** disks, if used, may reside on a single DASD volume, possibly the same volume as the 192, 41F, and 29E disks. It is recommended that the two groups reside on different physical DASD volumes attached to different physical control units connected to different physical channels or adapters. This allows one set to remain available in the event of a hardware error that makes the other set unavailable, which may make the difference whether or not the system remains operational pending repairs. Maximum redundancy will be obtained if the 1DB **12** disk is on a third volume, control unit, and channel or adapter. The other disks may be on any volumes.

9 The 2DF disk is optional. It ensures that the DirMaint machine can continue operations without loss of data in the event of a hardware or human error that prevents use of the 1DF disk. There is a slight degradation in response time because of this redundancy.

If the risk of regression is acceptable, you may omit the 2DF disk and rely on the 1DB disk for backup.

Note: If the 1DF disk becomes unavailable either temporarily or permanently, your directory could be regressed to the time of the previous backup.

By default, a backup is taken once per day, after Midnight. If you are running without a 2DF disk, you may schedule additional backups throughout the day. (For example, 0800 (8:00AM.), 1200 (Noon), and 1600 (4:00PM.).) For more information on scheduling additional backups, see "DIRMAINT DATADVH" on page 56.

Note: IBM recommends use of the 2DF disk.

10 The 1AA disk is optional. The most complete log of DIRMAINT activity is the DIRMAINT machine console spool file. By default, these console spool files are kept for nine days, for nine invocations of the DVHNDAY EXEC. The activity archive files on the 1AA disk contain less detail than the console spool file, but are retained until the disk becomes nearly full. Alternatively, if your system has an ESM with the ability for authorized virtual machines to write log records into the ESM audit trail, the DIRMAINT machine can use the ESM audit trail instead of the 1AA disk. Depending on the ESM in use, it may be either easy or difficult to isolate the DIRMAINT records from the other data in the ESM audit trail. Either form of recording involves a certain amount of overhead.

Note: IBM recommends use of an audit trail other than the console spool file. If your system is not using an ESM or is not including DIRMAINT records in the ESM audit trail, then you should use the 1AA disk. If you are recording the DIRMAINT activity in the ESM audit trail, you may omit the duplicate recording to the 1AA disk and obtain a slight improvement in system performance.

11 The 2AA disk is optional. If used, the contents are a duplicate of the 1AA disk.

Note: IBM recommends that the 2AA disk be defined and CMS formatted, but recommends that you do not enable the duplicate logging. This avoids the additional processing required when other type of logging occurs. However, this provides for continuous logging in the event of a hardware error that makes the 1AA disk unavailable. If a hardware failure should occur, DirMaint will shutdown, therefore IBM recommends to update the DVHPROFA file to the 2AA disk. For more information, see "DVHPROFA DIRMAINT" on page 43.

12 Specifies the primary directory backup. This is optional.

13 The 1DE disk is required on a 370 feature system and must be CP formatted as DRCT space. The 1DE disk is not required on an ESA feature system. If present, it is ignored.

Example—CKD Device: For a CKD device, it should be one cylinder larger than the DRCT space on your system residence volume:

```
SYSRES 123        DVH1DE 1DE
PERM 0000 0000    PERM 0000 0000
...
DRCT 0101 0130    DRCT 0001 0030
...
```

Thus, the 1DE disk would have a total size of 31 cylinders.

Example—FBA Device: For an FBA device, it should be 32 blocks larger than the DRCT space on your system residence volume.

```
SYSRES 123                              DVH1DE 1DE
PERM 0   3 (4 pages, 32 blocks)         PERM 0    3
PERM 4 150 (147 pages, 1176 blocks)     DRCT 4  150
...
```

Thus, the 1DE disk would have a total size of 1208 FB-512 blocks.

14 The 15D disk does not need to be formatted and contains no data. The ability or inability to obtain a link to the disk at any given time synchronizes directory updates between the DIRMAINT machine and the various DIRMSAT service machines in use. The DirMaint machine needs to link to the 15D disk before the DIRMSAT machine otherwise it is possible that no Dirmaint commands will get processed.

The 15D disk **is not** required on a system maintaining only a single copy of the object directory, for example, a 370 feature or a standalone ESA feature. If present, it is ignored.

The 15D disk **is** required on:

- Either a 370 or an ESA feature system maintaining duplicate copies of the object directory
- An ESA feature system in a multiple system CSE cluster.

DIRMAINT DASD Directory Statements for Migrating Customers

```
1 MDISK 491 RR - The Release 4 191 disk.
1 LINK *        491 191 M  - To run release 4.
  MDISK 112 MR - The Release 4 112 disk.
  MDISK 193 MR - The Release 4 193 disk.
  MDISK 194 MR - The Release 4 194 disk.
  MDISK 195 MR - The Release 4 195 disk.
  MDISK 196 MR - The Release 4 196 disk.
  MDISK 1A5 MR - The Release 4 1A5 disk.
  MDISK 1B0 MR - The Release 4 1B0 disk.
  MDISK 1CA MR - The Release 4 directory edit disk.
1 LINK P748XE4M 491 591 M  - Product code, primary.
  LINK P748XE4M 492 592 M  - Product code, alternate.
  LINK P748XE4M 11F 11F M  - Interface code, primary.
  LINK P748XE4M 41F 21F M  - Interface code, alternate.
2 LINK MAINT    123 BC4 MW - Object directory disk.
  MDISK 155 MR - Read/write scratch space, A-disk.
3 MDISK 1FA MR - Spool file staging space, Z-disk.
4 MDISK 1DF MR - Primary directory files.
  MDISK 2DF MR - Optional secondary directory files.
5 MDISK 1AA MR - Primary transaction history files (optional).
  MDISK 2AA MR - Optional secondary history files.
5 MDISK 1DB MR - Primary directory backup (optional).
6 MDISK 1DE MR - Directory edit scratch disk.
7 MDISK 15D RR - Intersystem locking disk.
```

Figure 5. DIRMAINT DASD Directory Statements for Migrating Customers

The Dirmaint product code and data files should be on conventional minidisk space rather than SFS space. However, any of these DIRMAINT MDISKS, except the 15D disk, may in fact reside in a directory in a shared file system.

Note: Although this is allowed, it is not recommended. It would be possible to get into a situation where the shared file servers are out of available DASD space and are not operational; DirMaint would be unable to allocate additional DASD space because the SFS severs are not operational.

If you choose to use shared file pool space, IBM recommends that a separate DIRCONTROL directory be used in the place of each *disk*. You will need to enter GRANT AUTHORITY command to all of the DIRMSAT for READ access (or READ and NEWREAD access if you chose to use FILECONTROL directories) to the directories that replace the 1DF and 2DF disks. You will need to update the DVHPROFA DIRMAINT and the DVHPROFA * files for each of the DIRMSAT machines accordingly. For more information, see “DVHPROFA DIRMAINT” on page 43

Ensure that access to minidisks is controlled by either passwords or explicit link authorization, as determined by the minidisk owner. Minidisk passwords are now optional for controlling minidisk directory links. None of the DirMaint MDISKs should have any directory passwords (except for the 11F disk, which must have a read password of ALL, or if an ESM is installed, must be defined as UACC (READ) or PUBLIC (READ)). Any virtual machine needing access to these disks should do so through a LINK directory statement. Using passwords on any of these disks increases the risk of unauthorized access to your system.

Access to backup tapes of any of these disks must be carefully controlled to prevent unauthorized access to your system.

The logon password of any virtual machine with either read or write access to any of these disks or with access to the backup tapes for any of these disks must be carefully chosen and controlled, to minimize the risk of unauthorized access to your system. DirMaint also supports control of minidisk links by an ESM. The following notes will help you with the example shown in Figure 5 on page 19.

Notice that with few exceptions, the directory entry is identical to what would be obtained by adding the DASD directory statements for new customers to your existing DIRMAINT machine directory entry.

1 The old MDISK 191 has been changed to MDISK 491 RR with a link statement added as * 491 191 MR. What will become the new 191 disk has been added as a

```
LINK P748XE4M 491 591 RR.
```

You can temporarily switch from one release level to the other or switch between test and production or between primary and backup levels by detaching the present 191 disk, linking to the other disk as 191, and re-IPLing CMS. For a permanent change, update the LINK directory statement to link to the desired disk as 191. In either case, you may need to move some data files around in the process. For more information, see the *DirMaint Program Directory*.

2 Specifies the object directory disk is linked as address BC4 rather than 123.

The choice of BC4 as the address is arbitrary. However, it must match the address used on the DIRECTORY statement in the USER INPUT file. The DirMaint Release 4 documentation uses BC4. In practice, it appears that most customers use either the real system residence volume disk address (the system IPL address) or the virtual address used by the MAINT user ID to refer to the system residence volume (usually 123). The choice is yours. In the Release 4 documentation this was shown as an MDISK BC4. This overlapped the MAINT 123 disk, sometimes causing confusion. IBM suggests that the MDISK BC4 entry be changed to a LINK to the MAINT disk that it formerly overlapped, usually 123. The PLANINFO file shows this as a link to the MAINT 123 disk.

3 The 1FA disk is optional. You may customize the DVHPROFA DIRMAINT file to reuse the 112 disk instead.

4 The 1DF disk is optional. You may customize the DVHPROFA DIRMAINT file to reuse the 196 disk instead.

5 The 1AA and 1DB disks are optional. You may customize the DVHPROFA DIRMAINT file to reuse the 193 disk instead.

6 The 1DE disk is optional. You may customize the DVHPROFA DIRMAINT file to reuse the 1CA disk instead.

7 The 15D disk is optional. You may customize the DVHPROFA DIRMAINT file to reuse the 194 disk instead. The DirMaint machine needs to link to the 15D disk before the DIRMSAT machine otherwise it is possible that no Dirmaint commands will get processed.

Note: If you choose to reuse the Release 4 disks for Release 5 purposes as shown in Figure 5 on page 19, remember to keep these disks when you permanently retire Release 4 and delete the other Release 4 disks.

Directory Statements for the DATAMOVE Virtual Machine

The directory statement examples are divided into two parts, with two alternatives for each part. The first part contains all of the non-DASD directory statements, with alternatives for 370 feature systems and ESA feature systems. The second part contains all of the DASD directory statements, with alternatives for new customers and migrating customers.

For a new customer, IBM recommends taking the default of DATAMOVE as the user ID of your DASD utility service machine.

For a migrating user, you can:

- Rename your existing DASD utility user ID, for example use **DATAR4** and reuse the DATAMOVE user ID for DirMaint Release 5.
- Use a different user ID, for example use **DATAR5** for the new release and keep your existing DATAMOVE user ID for your old release.
- Use the DATAMOVE user ID for both releases, and allow them to coexist throughout the migration period.

Notes:

1. If you are a migrating customer and want to keep the two releases on separate virtual machines, use the example for new customers.
2. If you are defining DATAMOVE in a multiple-system CSE cluster, IBM recommends that you define DATAMOVE on all systems in the cluster so that it can be brought up on an alternate system if the primary system is unavailable. The use of shared spool files is recommended. Neither the DATAMOVE input nor output spool files should be placed on the CSE exclusion list.
3. In a multiple system cluster you will probably have multiple virtual machines active and performing the DATAMOVE function at the same time on different systems in the cluster. It is possible to accomplish this if each of those virtual machines have the same user ID. IBM recommends that you avoid this and use a different user ID for each of the virtual machines performing the DATAMOVE function.

   If you find that your system startup procedures cause a DATAMOVE machine to be autologged on more than one system at a time, you will find that the first DATAMOVE server with any given user ID to be autologged should start as usual, and that any DATAMOVE servers with that same user ID that are subsequently started will issue an error message and immediately LOGOFF. This is because of the inability to obtain write access to the necessary minidisks.

   Any other DATAMOVE server with a different user ID should not encounter this difficulty and should start as usual.

DATAMOVE Non-DASD Directory Statements for 370 Feature Systems

```
USER DATAMOVE 1  NOLOG 2  16M 3  16M 3  BG 4  64 5
     6 INCLUDE COMMON
     7 ACCOUNT SYSTEM SYSPROG
     8 IUCV ANY PRIORITY MSGLIMIT 100
     9 OPTION BMX CONCEAL ECMODE REALTIME
```

Figure 6. DATAMOVE Non-DASD Directory Statements for 370 Feature Systems

These notes will help you with the example shown in Figure 6.

1 Identifies the user ID of the DATAMOVE virtual machine.

2 Identifies the password for the DATAMOVE virtual machine.

3 Specifies the required virtual storage size. DATAMOVE will function on as little as 6MB of storage for small source directories. IBM recommends using the largest virtual machine size available. If you chose to use a default size less than 16MB, you should leave the maximum size at 16MB.

4 Specifies the privilege classes based on VM-provided defaults. Privilege class B allows DATAMOVE to suppress CP message headers by using the CP MSGNOH command. Privilege class G is for general user commands.

5 Specifies the dispatching priority for the DATAMOVE virtual machine.

6 Specifies the directory PROFILE

7 Specifies the appropriate account to charge for the DATAMOVE virtual machine installation, and to route the printed output to the system programmer responsible for DATAMOVE operation.

8 Specifies the IUCV statement. This is optional unless your system is running with RACF or another ESM, and chooses to record commands received by DATAMOVE and messages sent by DATAMOVE in the ESM audit log. If ESM audit logging is enabled, an initial value of 100 is suggested for the MSGLIMIT.

Note: The ESM audit logging capability must be enabled if your system needs to comply with the TCSEC for either a class C2 or class B1 TCB.

9 Specifies the required options for the DATAMOVE virtual machine user ID.

DATAMOVE Non-DASD Directory Statements for ESA Feature Systems

```
USER DATAMOVE 1  NOLOG 2  16M 3  16M 3  BG 4
     5 INCLUDE COMMON
     6 ACCOUNT SYSTEM SYSPROG
     7 D8ONECMD FAIL LOCK
     8 IUCV ANY PRIORITY MSGLIMIT 100
     9 OPTION CONCEAL IGNMAXU
```

Figure 7. DATAMOVE Non-DASD Directory Statements for ESA Feature Systems

These notes will help you with the example shown in Figure 7.

1 Identifies the user ID of the DATAMOVE virtual machine.

2 Identifies the password for the DATAMOVE virtual machine.

3 Specifies the required virtual storage size. DATAMOVE will function on as little as 6MB of storage for small source directories. IBM recommends using the largest virtual machine size available. If you chose to use a default size less than 16MB, you should leave the maximum size at 16MB.

4 Specifies the privilege classes based on VM-provided defaults. Privilege class B allows DATAMOVE to suppress CP message headers by using the CP MSGNOH command. Privilege class G is for general user commands.

5 Specifies the directory PROFILE

6 Specifies the appropriate account to charge for the DATAMOVE virtual machine installation, and to route the printed output to the system programmer responsible for DATAMOVE operation.

7 The D8ONECMD statement is an optional statement. This ensures all DIRM DATAMOVE CP and DIRM DATAMOVE CMS commands are properly audited.

8 The IUCV statement is optional unless your system is running with RACF or another ESM, and chooses to record commands received by DATAMOVE and messages sent by DATAMOVE in the ESM audit log. If ESM audit logging is enabled, an initial value of 100 is suggested for the MSGLIMIT.

Note: The ESM audit logging capability must be enabled if your system needs to comply with the TCSEC for either a class C2 or class B1 TCB.

9 Specifies the directory options required by the DATAMOVE user ID.

DATAMOVE DASD Directory Statements for New Customers

```
1 LINK P748XE4M 491 591 RR - Product code, primary.
2 LINK P748XE4M 492 592 RR - Product code, alternate.
3 LINK P748XE4M 11F 11F RR - Interface code, primary.
4 LINK P748XE4M 41F 21F RR - Interface code, alternate.
5 MDISK 155 MR - Read/write scratch space, A-disk.
6 MDISK 1FA MR - Spool file staging space, Z-disk.
7 MDISK 1AA MR - Primary transaction history files (optional).
8 MDISK 2AA MR - Optional secondary history files.
```

Figure 8. DATAMOVE DASD Directory Statements for New Customers

The Dirmaint product code and data files should be on conventional minidisk space rather than SFS space. However, any of these DIRMAINT MDISKS, except the 15D disk, may in fact reside in a directory in a shared file system.

Note: Although this is allowed, it is not recommended. It would be possible to get into a situation where the shared file servers are out of available DASD space and are not operational; DirMaint would be unable to allocate additional DASD space because the SFS severs are not operational.

If you choose to use shared file pool space, IBM recommends that a separate DIRCONTROL directory be used in the place of each *disk*. You will need to enter GRANT AUTHORITY command to all of the DIRMSAT for READ access (or READ and NEWREAD access if you chose to use FILECONTROL directories) to the directories that replace the 1DF and 2DF disks. You will need to update the DVHPROFA DIRMAINT and the DVHPROFA * files for each of the DIRMSAT machines accordingly. For more information, see “DVHPROFA DIRMAINT” on page 43

Ensure that access to minidisks is controlled by either passwords or explicit link authorization, as determined by the minidisk owner. Minidisk passwords are now optional for controlling minidisk directory links. None of the DirMaint MDISKs should have any directory passwords (except for the 11F disk, which must have a read password of ALL, or if an ESM is installed, must be defined as UACC (READ) or PUBLIC (READ)). Any virtual machine needing access to these disks should do so through a LINK directory statement. Using passwords on any of these disks increases the risk of unauthorized access to your system.

Access to backup tapes of any of these disks must be carefully controlled to prevent unauthorized access to your system.

The logon password of any virtual machine with either read or write access to any of these disks or with access to the backup tapes for any of these disks must be carefully chosen and controlled, to minimize the risk of unauthorized access to your system. DirMaint also supports control of minidisk links by an ESM. The following notes are provided to help you with your Figure 8.

1 - 8 If you have chosen to locate any of the MAINT or P748XE4M “disks” being linked by the DATAMOVE machine into shared file system space, omit the LINK or MDISK statements for them from the DATAMOVE machine directory entry.

LINK statements for the MAINT disks have been omitted from the DATAMOVE machine directory entry. If they are not contained in the

included PROFILE, they will have to be added here. If you have installed the optional national language Help files, you should also include a LINK statement for those disks, either in the DATAMOVE machine directory entry or in the included profile.

5 Specifies the read/write scratch space, A-disk.

6 Specifies the spool file staging space, Z-disk.

7 The 1AA disk is optional. The most complete log of DATAMOVE activity is the DATAMOVE machine console spool file. By default, these console spool files are kept for nine days, for nine invocations of the DVHNDAY EXEC. The activity archive files on the 1AA disk contain less detail than the console spool file, but are retained until the disk becomes nearly full. Alternatively, if your system has an ESM with the ability for authorized virtual machines to write log records into the ESM audit trail, the DATAMOVE machine can use the ESM audit trail instead of the 1AA disk. Depending on the ESM in use, it may be either easy or difficult to isolate the DATAMOVE records from the other data in the ESM audit trail. Either form of recording involves a certain amount of overhead.

Note: IBM recommends use of an audit trail other than the console spool file. If your system is not using an ESM or is not including DATAMOVE records in the ESM audit trail, then you should use the 1AA disk. If you are recording the DATAMOVE activity in the ESM audit trail, you may omit the duplicate recording to the 1AA disk and obtain a slight improvement in system performance.

8 The 2AA disk is optional. If used, the contents are a duplicate of the 1AA disk.

Note: IBM recommends that the 2AA disk be defined and CMS formatted, but recommends that you do not enable the duplicate logging. This avoids the additional processing required when another type of logging occurs. However, this provides for continuous logging in the event of a hardware error that makes the 1AA disk unavailable.

If the 1AA **7** and 2AA **8** disks are both used, it is recommended that they reside on different physical DASD volumes attached to different physical control units connected to different physical channels or adapters. This allows one to remain available in the event of a hardware error that makes the other unavailable, which may make the difference between whether the system remains operational pending repairs.

DATAMOVE DASD Directory Statements for Migrating Customers

```
1 MDISK 491 RR - The Release 4 191 disk.
2 LINK *        491 191 M  - To run release 4.
3 LINK DIRMAINT 191 193 RR - As in Release 4.
3 LINK DIRMAINT 1A5 1A0 RR - As in Release 4.
3 LINK P748XE4M 491 591 RR - Product code, primary.
3 LINK P748XE4M 492 592 RR - Product code, alternate.
3 LINK P748XE4M 11F 11F RR - Interface code, primary.
3 LINK P748XE4M 41F 21F RR - Interface code, alternate.
3 MDISK 155 MR - Read/write scratch space, A-disk.
3 MDISK 1FA MR - Spool file staging space, Z-disk.
4 MDISK 1AA MR - Primary transaction history files (optional).
4 MDISK 2AA MR - Optional secondary history files.
```

Figure 9. DATAMOVE DASD Directory Statements for Migrating Customers

The Dirmaint product code and data files should be on conventional minidisk space rather than SFS space. However, any of these DIRMAINT MDISKS, except the 15D disk, may in fact reside in a directory in a shared file system.

Note: Although this is allowed, it is not recommended. It would be possible to get into a situation where the shared file servers are out of available DASD space and are not operational; DirMaint would be unable to allocate additional DASD space because the SFS severs are not operational.

If you choose to use shared file pool space, IBM recommends that a separate DIRCONTROL directory be used in the place of each *disk*. You will need to enter GRANT AUTHORITY command to all of the DIRMSAT for READ access (or READ and NEWREAD access if you chose to use FILECONTROL directories) to the directories that replace the 1DF and 2DF disks. You will need to update the DVHPROFA DIRMAINT and the DVHPROFA * files for each of the DIRMSAT machines accordingly. For more information, see “DVHPROFA DIRMAINT” on page 43

Ensure that access to minidisks is controlled by either passwords or explicit link authorization, as determined by the minidisk owner. Minidisk passwords are now optional for controlling minidisk directory links. None of the DirMaint MDISKs should have any directory passwords (except for the 11F disk, which must have a read password of ALL, or if an ESM is installed, must be defined as UACC (READ) or PUBLIC (READ)). Any virtual machine needing access to these disks should do so through a LINK directory statement. Using passwords on any of these disks increases the risk of unauthorized access to your system.

Access to backup tapes of any of these disks must be carefully controlled to prevent unauthorized access to your system.

The logon password of any virtual machine with either read or write access to any of these disks or with access to the backup tapes for any of these disks must be carefully chosen and controlled, to minimize the risk of unauthorized access to your system. DirMaint also supports control of minidisk links by an ESM. The following notes will help you with the example shown in Figure 9.

1 The old MDISK 191 has been changed to an MDISK 491 RR.

2 A LINK directory statement has been added to * 491 as 191 M.

3 A “LINK P748XE4M 491 591 RR” has been added, which will become the new 191 disk.

You can temporarily switch from one release level to the other or switch between test and production or between primary and backup levels by detaching the present 191 disk, linking to the other disk as 191, and re-IPLing CMS. For a permanent change, update the LINK directory statement to link to the desired disk as 191. For more information, see the *DirMaint Program Directory*. The 1AA and 2AA disks are optional. You must customize the DVHPROFM DATADVH file to use them if you need to archive DATAMOVE activities. To do this you must remove the comment or slash from the history transaction statement file:

Example—History Transaction File:

```
/PURPOSE= PTH FM= H ACC= 1AA
```

The notes in the “DATAMOVE DASD Directory Statements for New Customers” on page 25 are also applicable for the migrating user.

Directory Statements for the DIRMSAT Virtual Machine

The DIRMSAT directory example is divided into two parts, with two alternatives for each part. The first part contains all of the non-DASD directory statements, with alternatives for 370 feature systems and ESA feature systems. The second part contains all of the DASD directory statements, with alternatives for new customers and migrating customers.

For a new customer, IBM recommends using the default of DIRMSAT as the user ID of your cluster satellite service machines. For migrating users, you can:

- Rename your existing satellite servers; for example, use **DSATR4** and reuse the DIRMSAT user ID for DirMaint Release 5.
- Use a different user ID; for example, use **DSATR5** for the new release and keep your existing DIRMSAT user IDs for your old release.
- Use the DIRMSAT user ID for both releases, and allow them to coexist throughout the migration period.

Notes:

1. If you are a migrating customer and want to keep the two releases on separate virtual machines, use the example for new customers.
2. If you are defining DIRMSAT in a multiple-system CSE cluster, IBM recommends that you define DIRMSAT on all systems in the cluster so that it can be brought up on an alternate system if the primary system is unavailable. The use of shared spool files is recommended. The input and output spool files for DIRMSAT should not be placed on the CSE exclusion list.
3. In a multiple system cluster you may want to have multiple virtual machines performing the DIRMSAT function active at the same time on different systems in the cluster. It is possible to accomplish this although having the same user ID for each of those virtual machines. IBM recommends that you avoid this, and use a different user ID for each of the virtual machines performing the DIRMSAT function.

   If you find that your system startup procedures cause a DIRMSAT machine to be autologged on more than one system at a time, you will find that the first DIRMSAT server with any given user ID to be autologged should start as usual, and that any DIRMSAT servers with that same user ID that are subsequently started will issue an error message and immediately LOGOFF. This is because of the inability to obtain write access to the necessary minidisks. To suppress these messages, you can modify the DVHXLVL EXEC (an exit routine called by the PROFILE EXEC). For more information, see the *DirMaint Program Directory*.

   Any other DIRMSAT server with a different user ID should not encounter this difficulty and should start as usual.

DIRMSAT Non-DASD Directory Statements for 370 Feature Systems

```
USER DIRMSAT 1  NOLOG 2  16M 3  16M 3  BG 4  64 5
     6 INCLUDE COMMON
     7 ACCOUNT SYSTEM SYSPROG
     8 IUCV ANY PRIORITY MSGLIMIT 100
     9 OPTION BMX CONCEAL ECMODE REALTIME
```

Figure 10. DIRMSAT Non-DASD Directory Statements for 370 Feature Systems

These notes will help you with the example shown in Figure 10.

1 Identifies the user ID of the DIRMSAT virtual machine.

2 Identifies the password for the DIRMSAT virtual machine.

3 Specifies the required virtual storage size. DIRMSAT will function on as little as 6MB of storage for small source directories. IBM recommends using the largest virtual machine size available. If you chose to use a default size less than 16MB, you should leave the maximum size at 16MB.

4 Specifies the privilege classes based on VM-provided defaults.

This virtual machine is authorized to use:

- Privilege class B; this is required to do the following:
  - Allows DIRMSAT to suppress CP message headers, by using the CP MSGNOH command.
  - Allows DIRMSAT to issue the DIAGNOSE code X'84'.
  - Allows DIRMSAT to issue the DIAGNOSE code X'3C'.

  Note: By default, these are all CP privilege class B functions. If your site has changed these to other privilege classes, then DIRMSAT must be authorized for the privilege classes containing these functions.
- Privilege class G; this is for general user commands.

5 Specifies the dispatching priority for the DIRMSAT virtual machine.

6 Specifies the directory PROFILE.

7 Specifies the appropriate account to charge for the DIRMSAT virtual machine installation, and to route the printed output to the system programmer responsible for DIRMSAT operation.

8 The IUCV statement is optional unless your system is running with RACF or another ESM, and chooses to record commands received by DIRMSAT and messages sent by DIRMSAT in the ESM audit log. If ESM audit logging is enabled, an initial value of 100 is suggested for the MSGLIMIT.

Note: The ESM audit logging capability must be enabled if your system needs to comply with the TCSEC for either a class C2 or class B1 TCB.

9 Specifies required options for the DIRMSAT virtual machine.

DIRMSAT Non-DASD Directory Statements for ESA Feature Systems

```
USER DIRMSAT 1 NOLOG 2 16M 3 16M 3 BG 4
     5 INCLUDE COMMON
     6 ACCOUNT SYSTEM SYSPROG
     7 D8ONECMD FAIL LOCK
     8 IUCV ANY PRIORITY MSGLIMIT 100
     9 OPTION CONCEAL D84NOPAS IGNMAXU
```

Figure 11. DIRMSAT Non-DASD Directory Statements for ESA Feature Systems

These notes will help you with the example shown in Figure 11.

1 Identifies the user ID of the DIRMSAT virtual machine.

2 Identifies the password for the DIRMSAT virtual machine.

3 Specifies the required virtual storage size. DIRMSAT will function on as little as 6MB of storage for small source directories. IBM recommends using the largest virtual machine size available. If you chose to use a default size less than 16MB, you should leave the maximum size at 16MB.

4 Specifies the privilege classes based on VM-provided defaults.

This virtual machine is authorized to use:

- Privilege class B; this is required to do the following:
  - Allows DIRMSAT to suppress CP message headers, by using the CP MSGNOH command.
  - Allows DIRMSAT to issue the DIAGNOSE code X'84'.
  - Allows DIRMSAT to issue the DIAGNOSE code X'3C'.

  Note: By default, these are all CP privilege class B functions. If your site has changed these to other privilege classes, then DIRMSAT must be authorized for the privilege classes containing these functions.
- Privilege class G; this is for general user commands.

5 Specifies the directory PROFILE.

6 Specifies the appropriate account to charge for the DIRMSAT virtual machine installation, and to route the printed output to the system programmer responsible for DIRMSAT operation.

7 The D8ONECMD statement is an optional statement that ensures all DIRM SATELLITE CP and DIRM SATELLITE CMS commands are properly audited.

8 The IUCV statement is optional unless your system is running with RACF or another ESM, and chooses to record commands received by DIRMSAT and messages sent by DIRMSAT in the ESM audit log. If ESM audit logging is enabled, an initial value of 100 is suggested for the MSGLIMIT.

Note: The ESM audit logging capability must be enabled if your system needs to comply with the TCSEC for either a class C2 or class B1 TCB.

9 Specifies the directory options required by the DIRMSAT user ID.

The D84NOPAS option is optional for a standalone system without an ESM, or for a standalone system with RACF as the ESM. It may also be optional for systems with other ESMs. This is recommended for systems in a multiple system cluster. It is required in a multiple system cluster if any

directory entries contain the SYSAFFIN EXIST_AT, SYSAFFIN LOGON_AT, or SYSAFFIN NOLOG_AT statements.

DIRMSAT DASD Directory Statements for New Customers

```
 1 LINK P748XE4M 491 591 RR - Product code, primary.
 2 LINK P748XE4M 492 592 RR - Product code, alternate.
 3 LINK P748XE4M 11F 11F RR - Interface code, primary.
 4 LINK P748XE4M 41F 21F RR - Interface code, alternate.
 5 LINK MAINT    123 123 MW - Object directory disk.
 6 LINK DIRMAINT 1DF 1DF RR - Primary directory files.
 7 LINK DIRMAINT 2DF 2DF RR - Optional secondary directory files.
 8 LINK DIRMAINT 15D 15D RR - Intersystem locking disk.
 9 MDISK 155 MR - Read/write scratch space, A-disk.
10 MDISK 1FA MR - Spool file staging space, Z-disk.
11 MDISK 1AA MR - Primary transaction history files (optional).
12 MDISK 2AA MR - Optional secondary history files.
```

Figure 12. DIRMSAT DASD Directory Statements for New Customers

The Dirmaint product code and data files should be on conventional minidisk space rather than SFS space. However, any of these DIRMAINT MDISKS, except the 15D disk, may in fact reside in a directory in a shared file system.

Note: Although this is allowed, it is not recommended. It would be possible to get into a situation where the shared file servers are out of available DASD space and are not operational; DirMaint would be unable to allocate additional DASD space because the SFS severs are not operational.

If you choose to use shared file pool space, IBM recommends that a separate DIRCONTROL directory be used in the place of each *disk*. You will need to enter GRANT AUTHORITY command to all of the DIRMSAT for READ access (or READ and NEWREAD access if you chose to use FILECONTROL directories) to the directories that replace the 1DF and 2DF disks. You will need to update the DVHPROFA DIRMAINT and the DVHPROFA * files for each of the DIRMSAT machines accordingly. For more information, see “DVHPROFA DIRMAINT” on page 43

Ensure that access to minidisks is controlled by either passwords or explicit link authorization, as determined by the minidisk owner. Minidisk passwords are now optional for controlling minidisk directory links. None of the DirMaint MDISKs should have any directory passwords (except for the 11F disk, which must have a read password of ALL, or if an ESM is installed, must be defined as UACC (READ) or PUBLIC (READ)). Any virtual machine needing access to these disks should do so through a LINK directory statement. Using passwords on any of these disks increases the risk of unauthorized access to your system.

Access to backup tapes of any of these disks must be carefully controlled to prevent unauthorized access to your system.

The logon password of any virtual machine with either read or write access to any of these disks or with access to the backup tapes for any of these disks must be carefully chosen and controlled, to minimize the risk of unauthorized access to your system. DirMaint also supports control of minidisk links by an ESM. The following notes will help you with the example shown in Figure 12.

1 - 7 and 9 - 12. If you have chosen to locate any of the MAINT, P748XE4M or DIRMAINT "disks" being linked by the DIRMSAT machine into shared file system space, omit the LINK or MDISK statements for them from the DIRMSAT machine directory entry.

LINK statements for the MAINT disks have been omitted from the DIRMSAT machine directory entry. If they are not contained in the included PROFILE, they will have to be added here. If you have installed the optional national language Help files, you should also include a LINK statement for those disks, either in the DIRMSAT machine directory entry or in the included profile.

5 The choice of 123 as the address of the object directory disk is arbitrary. It must match the address used on the DIRECTORY statement in the USER INPUT file for the system affinity being processed by the particular DIRMSAT service machine. The DirMaint Release 4 documentation uses BC4. The PLANINFO file shows this as a link to the MAINT 123 disk. In practice, it appears that most customers use either the real system residence volume disk address (the system IPL address), or the virtual address used by the MAINT user ID to refer to the system residence volume (usually 123). The choice is yours.

The use of link mode MW on the object directory disk is correct. This is necessary for the MAINT user ID to update the CP nucleus on the system residence volume without shutting DirMaint down. This is one of the rare situations where MW is appropriate for use with a CMS application program.

6 Specifies the primary directory files.

7 The 2DF disk is optional. Its use ensures that the DIRMSAT machine can continue operations without loss of data in the event of a hardware or human error that prevents use of the 1DF disk.

Note: If the DIRMAINT machine has a 2DF disk defined, IBM recommends that DIRMSAT machines link it.

8 Specifies the intersystem locking disk.

9 Specifies the read/write scratch space, A-disk.

10 Specifies the spool file staging space, Z-disk.

11 The 1AA disk is optional. The most complete log of DIRMSAT activity is the DIRMSAT machine console spool file. By default, these console spool files are kept for nine days, for nine invocations of the DVHNDAY EXEC. The activity archive files on the 1AA disk contain less detail than the console spool file, but are retained until the disk becomes nearly full. Alternatively, if your system has an ESM with the ability for authorized virtual machines to write log records into the ESM audit trail, the DIRMSAT machine can use the ESM audit trail instead of the 1AA disk. Depending on the ESM in use, it may be either easy or difficult to isolate the DIRMSAT records from the other data in the ESM audit trail. Either form of recording involves a certain amount of overhead.

Note: IBM recommends use of an audit trail other than the console spool file. If your system is not using an ESM or is not including DIRMSAT records in the ESM audit trail, then you should use the 1AA disk. If you are recording the DIRMSAT activity in the ESM audit trail, you may omit the duplicate recording to the 1AA disk and obtain a slight improvement in system performance.

12 The 2AA disk is optional. If used, the contents are a duplicate of the 1AA disk.

Note: IBM recommends that the 2AA disk be defined and CMS formatted, but recommends that you do not enable the duplicate logging. This avoids the additional processing required when another type of logging occurs. However, this provides for continuous logging in the event of a hardware error that makes the 1AA disk unavailable.

It is recommended that the 1AA **11** and 2AA **12** disks reside on different physical DASD volumes attached to different physical control units connected to different physical channels or adapters. This allows one to remain available in the event of a hardware error that makes the other unavailable, which may make the difference between whether the system remains operational pending repairs.

DIRMSAT DASD Directory Statements for Migrating Customers

```
1 MDISK 491 RR - The Release 4 191 disk.
2 LINK * 491 191 M - To run release 4.
  LINK DIRMAINT 191 119 RR - As in Release 4.
  LINK DIRMAINT 195 159 RR - As in Release 4.
  LINK DIRMAINT 196 169 RR - As in Release 4.
3 LINK P748XE4M 491 591 RR - Product code, primary.
  LINK P748XE4M 492 592 RR - Product code, alternate.
  LINK P748XE4M 11F 11F RR - Interface code, primary.
  LINK P748XE4M 41F 21F RR - Interface code, alternate.
4 LINK MAINT    123 BC4 MW - Object directory disk.
5 LINK DIRMAINT 1DF 1DF RR - Primary directory files.
  LINK DIRMAINT 2DF 2DF RR - Optional secondary directory files.
6 LINK DIRMAINT 15D 15D RR - Intersystem locking disk.
  MDISK 155 MR - Read/write scratch space, A-disk.
  MDISK 1FA MR - Spool file staging space, Z-disk.
  MDISK 1AA MR - Primary transaction history files (optional).
  MDISK 2AA MR - Optional secondary history files.
```

Figure 13. DIRMSAT DASD Directory Statements for Migrating Customers

These notes will help you with the example shown in Figure 13.

1 The old MDISK 191 has been changed to an MDISK 491 RR.

2 A LINK directory statement has been added to * 491 as 191 M.

3 A “LINK P748XE4M 491 591 RR” has been added, which will become the new 191 disk.

You can temporarily switch from one release level to the other or switch between test and production or between primary and backup levels by detaching the present 191 disk, linking to the other disk as 191, and re-IPLing CMS. For a permanent change, update the LINK directory statement to link to the desired disk as 191. For more information, see the *DirMaint Program Directory*.

4 Specifies the object directory disk is linked as address BC4 rather than 123.

The choice of BC4 as the address is arbitrary. However, it must match the address used on the DIRECTORY statement in the USER INPUT file. The DirMaint Release 4 documentation uses BC4. In practice, it appears that most customers use either the real system residence volume disk address (the system IPL address) or the virtual address used by the MAINT user ID to refer to the system residence volume (usually 123). The choice is yours. In the Release 4 documentation this was shown as an MDISK BC4. This overlapped the MAINT 123 disk, sometimes causing confusion. IBM suggests that the MDISK BC4 entry be changed to a LINK to the MAINT disk that it formerly overlapped, usually 123. The PLANINFO file shows this as a link to the MAINT 123 disk.

5 The 1DF disk is optional. You may customize the DVHPROFA DIRMSAT file to reuse the 169 disk instead.

6 The 15D disk is optional. You may customize the DVHPROFA DIRMSAT file to reuse the 194 disk instead. The DirMaint machine needs to link to the 15D disk before the DIRMSAT machine otherwise it is possible that no Dirmaint commands will get processed.

The notes in the “DIRMSAT DASD Directory Statements for New Customers” on page 33 are also applicable for the migrating user.

Directory Entry for the RSCS Virtual Machine

If you are running DirMaint in a multiple system cluster, but are not using CSE shared spool files, or if your DIRMAINT service machine must accept transactions from the network, then you should enable the RSCS network machines to use DIAGNOSE code X'F8' to set the secure spool file origin. This is done by specifying the SETORIG keyword on the OPTION statement in the RSCS machine directory entry.

For more information, see *VM RSCS: Planning and Installation*.

Chapter 3. Tailoring the DIRMAINT Service Machine

This chapter provides guidance for tailoring the various data files used by the DIRMAINT service machine. Addressing only those statements in the CONFIG* DATADVH file(s) that you must check to ensure that DirMaint is initially in fail safe mode before performing the Installation Verification Procedures (IVP).

You must tailor several Release 4 files so they can be used with Release 5.0. After you make these changes, however, the files are updated by the DirMaint commands. For more information on these commands, see the *Directory Maintenance VM/ESA: Command Reference*. The files will be described in the sections that follow, along with those few files that require tailoring and have no command to manipulate them.

Data Files

To aid you in tailoring your files:

- The IN2PROD SAMP has placed these tailorable files on the P748XE4M 492 minidisk:
  - DVHNAMES DATADVH
  - DIRMAINT DATADVH
  - DIRMSAT DATADVH
  - DATAMOVE DATADVH
  - DVHPROFM DATADVH
  - DVHPROFA DIRMAINT
  - DVHPROFA DIRMSAT
  - PROFILE EXEC
- The IN2PROD SAMP has placed these tailorable files on the P748XE4M 41F minidisk:

  - CONFIG DATADVH
  - DIRMMAIL DATADVH
  - 140CMDS DATADVH
  - 150CMDS DATADVH

  Note: These files should only be modified on the P748XE4M 492/41F test minidisks. Once IN2PROD PROD has been run to place the DirMaint code into production on the P748XE4M 491/11F production minidisks, change these tailorable files by:

  1. Request the DIRMAINT server detach the P748XE4M 492 and 41F disks, DIRMAINT's 192 and 21F disks.
  2. Make the changes to the tailorable file on the test disk
  3. Use the DIRM FILE command to send the updated file back to the DIRMAINT service machine and replace the previous copy on the appropriate production disk.
  4. Use the DIRM RLDDATA command to place the changed tailorable file into production.

- You can use a DIRM SEND command to send the current copy of the file to your reader. Receive the file onto your disk. Edit the file and file the changes back onto your disk. Use a DIRM FILE command to send the file back to the DIRMAINT service machine and replace the previous copy of the file. And finally, use a DIRM RLDDATA or DIRM RLDEXTN command to place the changed file into production.
- If you want more control and tracking of the changes to these files, you may register your changes as local modifications to VMSES/E. Make the changes using XEDIT with AUX and UPDATE files. Merge the updates using EXECUPDT. Then, use the DIRM FILE and DIRM RLDDATA or DIRM RLDEXTN commands as described above to put the updated file into production. For more information about using VMSES/E to register, edit, and merge your updates, see the *DirMaint Program Directory*.

For more information about the format of these files, see the *Directory Maintenance VM/ESA: Diagnosis Reference*.

The files you may need to tailor are addressed in the order you should perform the tailoring. Your most static, least likely to change, files should be tailored first. Volatile files that are likely to change although you are tailoring other files should be tailored last, just before beginning the Installation Verification Procedures (IVP).

Table 1 (Page 1 of 2). New Files for DirMaint Release 5.0

File Name	Description	Page
PROFILE XEDIT	This file determines the characteristics of your editing sessions. This file should be tailored first.	43
DVHPROFA DIRMAINT	This file determines what minidisks or shared file system directories are accessed at what file mode letters during initialization. This file should be tailored second.	43

Table 1 (Page 2 of 2). New Files for DirMaint Release 5.0

File Name	Description	Page
CONFIG DATADVH	This file contains most of the tailorable parameters used throughout the DirMaint Release 5 product. (This is the primary replacement for the Release 4 DIRMAINT DATA file.)	44
DIRMAINT DATADVH	This file identifies the schedule of events to the DirMaint service machine that happen at specific set dates, times, or intervals. (This replaces the time related entries from the Release 4 DIRMAINT DATA file.)	56
DVHNAMES DATADVH	This file identifies the user ID's to be notified of any significant events that happen in the various DirMaint service machines. (This replaces the DIRM_MONITOR entries from the Release 4 DIRMAINT DATA file.)	59
DIRMMAIL SAMPDVH	The file identifies a sample for the DIRMMAIL NEWFILE file.	61

The following files contain more static information. New customers may consider these files part of the previous group. Migrating customers should convert these files next.

Table 2. New Files for DirMaint Release 5.0 Containing Static Information.

File Name	Description	Page
DVHLINK EXCLUDE	This file contains a list of minidisk addresses and their owners that are excluded from the DVHLINK FILE, and are therefore not included in the results of a DIRM REVIEW command and are not delinked or moved if or when commands are processed that remove the underlying minidisk. This replaces the R4 LINKS EXCLUDE file.	61
PWMON CONTROL	This file contains a list of user ID's whose passwords do not expire, do not receive password expiration notices, or have their password expiration notices sent to an alternate user ID or node ID.	62
RPWLIST DATA	This file contains a list of prohibited passwords.	63

If you intend to immediately bring DirMaint up with DASD Management functions, you will want to tailor the following two files next. Otherwise, you may defer tailoring of these two files until after completion of the IVP.

Table 3. New Files for Bringing up DASD Management with DirMaint Release 5.0

File Name	Description	Page
EXTENT CONTROL	This file provides information needed for DirMaint's DASD Management functions. This chapter will only address how to migrate an existing Release 4 EXTENT CONTROL file into Release 5 format. For more information on the EXTENT CONTROL file, see "The Extent Control File" on page 88.	87
AUTHDASD CONTROL	This file is new for Release 5. It determines who can allocate space in what DASD groups, regions, or volumes.	87

These last two files are the most volatile of the group. You will want to save the preparation of these files for last.

Table 4. Volatile Files to Change Just Bringing Up DirMaint Release 5.0

AUTHFOR CONTROL	This file identifies what user ID's (or profile IDs) have delegated authority for another user ID to act for them, and what command sets are included in that authority. (This replaces the DIRM_STAFF, DIRM_SUBSTAFF, and OWNER entries in the DirMaint Release 4 DIRMAINT DATA file; plus the Release 4 ASSIGN FILE.)	63
USER INPUT	This file is your existing source directory file.	68

Notes:

For customers migrating from DirMaint Release 4 to DirMaint Release 5, the DVHMIGR8 EXEC may be used to convert:

- ASSIGN FILE and DIRMAINT DATA to AUTHFOR CONTROL
- EXTENT CONTROL
- LINKS EXCLUDE to DVHLINK EXCLUDE
- PWMON CONTROL

For more information on linking to the necessary disks, accessing them in the proper order, and copying the output files to the correct destination disks, see the *DirMaint Program Directory*.

Note: Do not attempt to use the DVHMIGR8 EXEC to convert data files from DirMaint Release 3 or prior releases, the results will be unusable

Accessing Disks

The following sections assume that the following disks, or the shared file system directory equivalents, have been accessed at the indicated file mode letters.

```
A - 191
B - Reserved for a VMSES/E disk
D - Reserved for a VMSES/E disk
E - If migrating, DirMaint's 196 disk
F - If migrating, DirMaint's 195 disk
G - If migrating, DirMaint's 193 disk
H - If migrating, DirMaint's 191 disk
I - If migrating, DirMaint's 1A5 disk
J - 1DF (Primary Directory Files)
K - 492 (DirMaint's Test 191)
L - 41F (DirMaint's Test 11F)
```

PROFILE XEDIT

The PROFILE XEDIT will customize your editing sessions. There is none supplied with the DirMaint product, because the product itself does not require one in order to operate. From your regular VM user ID, enter:

Step 1. SENDFILE PROFILE XEDIT * P748XE4M

Step 2. Enter, a RECEIVE command from the P748XE4M user ID.

If you don't already have a the PROFILE XEDIT file to send, you can create one, enter:

```
XEDIT PROFILE XEDIT A
SET CASE M I
INPUT /* */
INPUT Address 'XEDIT'
INPUT 'COMMAND SET CASE M I'
INPUT Exit
FILE
```

You can also enter the following command if you need to logon to the DirMaint service machine and look at any files:

```
COPYFILE PROFILE XEDIT A = = K (OLDDATE)
```

DVHPROFA DIRMAINT

This is the second file you must consider tailoring. It determines what disks or shared file system directories are accessed at what file mode letters. This file is created by running the IN2PROD SAMP. For information, see the *DirMaint Program Directory*.

If you have installed DirMaint Release 5.0 using the recommended disk addresses shown in the *DirMaint Program Directory*, then this file requires no tailoring. Otherwise, you must update the file to correspond with the disk addresses or shared file directory names you have established.

The file should be RECFM V, and must reside on the 492 disk, the format of the file is described within the file itself. The file type of this file must match the user ID name running the DIRMAINT server. If not DIRMAINT, then rename as appropriate.

CONFIG DATADVH

The CONFIG DATADVH file contains a large number of local customization options. These can be used to enable DirMaint to work with an ESM, such as IBM's RACF or an equivalent ESM available from other vendors, fine tune DirMaint for optimum performance in YOUR environment, and enable or disable selected optional capabilities.

The format of the file is described within the file itself. It should be RECFM V, and must reside on the user interface disk(s).

These files differ from DirMaint Release 4 in that multiple CONFIG* DATADVH files are allowed. There are two types of entries in these files: using single occurance entries and using all occurrences of the keyword search string.

Important

The CONFIG DATADVH file is an IBM part that should never be modified. Desired changes should be made in an override file. An override file has a file name of CONFIG* and file type of DATADVH as explained below.

Example—Using a Occurance Value Entry:

```
PASSWORD_RANDOM_GENERATOR_EXIT= DVHPXR EXEC
```

The order in which multiple CONFIG* DATADVH files are searched is significant. Files are searched in reverse alphabetical order: CONFIG99 before CONFIG0, CONFIG0 before CONFIGZZ, CONFIGZZ before CONFIGA, and CONFIGA before CONFIG. If there are two (or more) occurrences of the same file name, only the first one is used (file mode A, or the file mode letter closest to A). If there are two or more occurrences of the keyword search string:

```
PASSWORD_RANDOM_GENERATOR_EXIT=
```

in any of these files, only the first one is used.

Example—Using all Occurrences of the Keyword Search String:

```
LOADABLE_SERV_FILE= DVHWAIT  EXEC
LOADABLE_SERV_FILE= DVHRDR   EXEC
LOADABLE_SERV_FILE= DVHRQST  EXEC
LOADABLE_SERV_FILE= DVHCEXIT EXEC
LOADABLE_SERV_FILE= DVHADZ   EXEC
LOADABLE_SERV_FILE= DVHAEZ   EXEC
LOADABLE_SERV_FILE= DVHMSG   EXEC
```

Any and all CONFIG* DATADVH files are searched in the same order used for the exit routine example. However, all records that match the search string are used. In this loadable file example, the order is significant. In other cases the order may not matter.

Some of the information in these CONFIG* DATADVH files is required by the user's virtual machine to enter DirMaint commands. If you split the IBM-supplied CONFIG file into multiple files, you may keep some of the files on disks accessible only to the DirMaint service machines: DIRMAINT, DATAMOVE, and DIRMSAT; but some of the files must remain on the user interface disk.

The comments within the file describe each statement, including a description of the statement keyword, the acceptable values for that statement, the significance of each of those values where not obvious, and whether only the first occurrence of the statement found in the various CONFIG* DATADVH file(s) is used or whether all occurrences are used.

For more information, see “The CONFIG* DATADVH File” on page 120.

Step 1. Select Directory Update Options

The first statements to check are:

```
1    RUNMODE= TESTING | OPERATIONAL
2 // SRCUPDATE= NOP | DISABLED
3    ONLINE= OFFLINE | SCHED | IMMED
4    UPDATE_IN_PLACE= YES | NO
5    WRK_UNIT_ONLINE= NO | YES
6 // DIRECTXA_OPTIONS= MIXED | MIXED NOMIXMSG
7    SORT_DIRECTORY= NO | YES
8    SORT_BY_DEVICE_ADDRESS= NO | YES
9    BACKUP REBUILD= CLUSTER DVHLINK <VCONTROL> | NONE
10   CLASS LIMIT ON USER STATEMENT=      8 | 0 ... 32 | 0 ... 8
11   CLASS STATEMENT IN PROFILE CHECK = NO | YES
12   PW_MONITOR=userid
13   WRK_UNIT_CLEANUP= ERASE | RENAME
14   PW_REUSE_HASHING_EXIT=
15   PW_REUSE_INTERVAL=
```

Figure 14. Selecting Directory Update Options

1 The `RUNMODE= TESTING` statement ensures that DirMaint will not make any changes to your source directory file as the result of any commands that are issued to the DIRMAINT service machine. When you have completed the first part of your IVP, changing the statement to `RUNMODE= OPERATIONAL` will enable DirMaint to begin making changes to the source directory.

2 The `SRCUPDATE= NOP` statement ensures that DirMaint will make changes to the source directory as requested, following an IPL or a DIRM RLDDATA command, until a DIRM DISABLE command is entered. The `SRCUPDATE= DISABLED` statement ensures that DirMaint will not make any changes to your source directory file as the result of any commands, following an IPL or a DIRM RLDDATA command, until a DIRM ENABLE command is entered. IBM recommends omitting this statement from the CONFIG* DATADVH file, allowing the enable/disable state established by using DIRM DISABLE and DIRM ENABLE commands to persist across an IPL or DIRM RLDDATA command.

3 Even with `RUNMODE= OPERATIONAL` and with `SRCUPDATE= NOP` the ONLINE= OFFLINE statement will prevent DirMaint from updating the object directory.

Changes to the source directory will have no effect on your system unless and until the updated directory is placed online using the DIRECT command for a VM/ESA 370 feature system or DIRECTXA command for a VM/ESA feature system. When you have completed the remainder of the IVP, you may enable updates to the object directory by changing the:

- `ONLINE=` statement to `ONLINE= IMMED`

  or

- `ONLINE=` statement to `ONLINE= SCHED` and change the date from 12/31/94 to ==/==/== or another date value suggested for the DIRECT entry in the DIRMAINT DATADVH file. For more information, see the “DIRMAINT DATADVH” on page 56.

A small system installation would use `ONLINE= IMMED` and a large system installation would use `ONLINE= SCHED` If the users or administrators are unable to enter subsequent commands because DIRMAINT is still busy

processing the previous command, you have reached large system status, and should switch to ONLINE= SCHED.

4 The UPDATE_IN_PLACE= YES entry has no effect when the ONLINE= entry is set to OFFLINE. After the ONLINE= entry has been set to either SCHED or IMMED, you will find that use of UPDATE_IN_PLACE=YES will give better performance and response time than using UPDATE_IN_PLACE= NO. IBM recommends UPDATE_IN_PLACE= YES for all systems.

5 The WRK_UNIT_ONLINE= YES entry has no effect when the ONLINE= entry is set to OFFLINE or IMMED. If the ONLINE= entry has been set to SCHED, and the WRK_UNIT_ONLINE= is set to NO, you may find that it takes too long to complete processing the DASD management commands: AMDISK with formatting options, CMDISK, DMDISK with cleanup being performed, PURGE with cleanup being performed, and so forth. These work units can be accelerated by using WRK_UNIT_ONLINE= YES. However, this may effectively negate the difference between ONLINE= SCHED and ONLINE= IMMED. If you find that your DIRMAINT service machine is spending more time placeing the directory changes online than it is in making directory source updates, and you already have ONLINE= SCHED, then you will need to use WRK_UNIT_ONLINE= NO.

6 The DIRECTXA_OPTIONS= entry is passed along to the CP DIRECTXA command. Valid options are to leave it blank, specify MIXED, or specify both MIXED and NOMIXMSG. For more information on these parameters, see the *z/VM: CP Command and Utility Reference*.

If you already have a clean directory that gives no error messages from DIRECTXA, then IBM recommends you leave this entry blank. This will allow DIRECTXA and therefore DirMaint to perform the maximum degree of error checking before making directory updates. If you have migrated from VM/SP, VM/SP HPO, or VM/ESA 370 feature and have not removed the 370 unique statements from your directory, then IBM recommends use of both MIXED and NOMIXMSG.

This configuration file entry has no effect for a VM/ESA 370 feature system.

7 The SORT_DIRECTORY value specifies whether the USER DIRECT file is to be maintained in sorted order. Specifying YES increases the time and storage requirements for BACKUP processing.

8 The SORT_BY_DEVICE_ADDRESS value specifies whether the device statements in each user directory are maintained in sorted order by device address. Specifying YES increases the time and storage requirements for all updates to directory entries either PROFILE or USER containing device statements.

9 The BACKUP REBUILD= CLUSTER DVHLINK <VCONTROL> | NONE statement controls the balance between the time taken up to complete a BACKUP operation and the amount of clean-up that was done by the BACKUP operation and the resulting DASD utilization. The keywords used within the statement are:

Table 5. Tags in the CMS NAMES File

Keyword	Description
CLUSTER	Specifies USER DIRECT file, all CLUSTER files, and all DIRMPART files are erased and rebuilt as part of BACKUP processing. This reclaims space in existing CLUSTER files from directory entries that have been updated and are now separate DIRMPART files.
DVHLINK	Specifies the DVHLINK FILE is rebuilt to reflect any changes in the DVHLINK EXCLUDE file since the previous BACKUP run.
VCONTROL	Specifies all VCONTROL files are erased and rebuilt as part of BACKUP processing. This reclaims DASD space for any VCONTROL files that describe volumes that have been removed from the system and corrects for changes made to the EXCLUDE section of the EXTENT CONTROL file that were not followed by a RLDEXTN command. If the statement is omitted or is present with no value, the default is CLUSTER DVHLINK. The keyword value of NONE may be used.

10 The `CLASS LIMIT ON USER STATEMENT=` specifies how many CP privilege classes may be included on the USER statement. The valid range for VM/ESA in general is 0 to 32. The valid range for VM/ESA 1.1.5 370 feature is 0 to 8. The default range is 8. A new CLASS directory statement is created if the:

- Limit is set to 0
- Number of classes defined for a user entry exceeds the specified limit
- Classes have a system affinity other than the *

11 The `CLASS STATEMENT IN PROFILE CHECK=` statement specifies whether DirMaint will do the additional checking to see if an included PROFILE contains a CLASS statement. If an included PROFILE contains a CLASS statement, the USER statement **must** specify the class as an * (except on the VM/ESA 1.1.5 370 feature), regardless of the `CLASS LIMIT ON USER STATEMENT` setting and the number of privilege classes used. This statement is ignored when used on the VM/ESA 1.1.5 370 feature.

Note: Unless you are running on the VM/ESA 1.1.5 370 feature, you should experiment with:

- `CLASS_LIMIT_ON_USER_STATEMENT= 0`
- `CLASS_STATEMENT_IN_PROFILE_CHECK= NO`
- `CLASS_LIMIT_ON_USER_STATEMENT= 8 ... 32`
- `CLASS_STATEMENT_IN_PROFILE_CHECK= YES`

The right combination for performance varies from system to system, and may vary depending on whether you are using the operand ONLINE= IMMED.

12 The `PW_MONITOR= userid` statement is used when the user needs to contact someone authorized to issue a SETPW command for their user ID in the event their logon password has expired and been set to NOLOG.

13 The `WRK_UNIT_CLEANUP=` value controls whether the WORKUNIT files will be erased or renamed to WORKSAVE after the completion of the DASD management commands. In the event of a failure, they will be renamed to WUCFFAIL in either case.

14 The PW_REUSE_HASHING_EXIT routine hashes the user's password for storage in the password history file. The file type may be either EXEC or MODULE. The IBM supplied default is DVHHASH MODULE. If not specified, the passwords will be stored in the history file as hexadecimal digits.

15 The PW_REUSE_INTERVAL identifies how long an entry is kept in the password history file. It may be either a time period with a DAYS suffix, or a count with no suffix. The IBM supplied default is 365 DAYS.

Step 2. Select Restart and Recovery Characteristics

Next, enable DirMaint restart recovery capabilities.

```
1 SHUTDOWN_LOGOFF_THRESHHOLD= 3      /* Choose 2, 3, or 4.           */
2 SHUTDOWN_RESET_THRESHHOLD=  3      /* The s_r_t must be >= 1.      */
3 SHUTDOWN_REIPL_COMMAND=     CP IPL CMS PARM AUTOCR
```

Figure 15. Selecting Restart and Recovery Characteristics

1 The SHUTDOWN_LOGOFF_THRESHHOLD value specifies the number of error induced shutdown conditions that may be encountered before the service machine logs itself off, if running disconnected. The recommended values are: 2, 3, or 4.

2 The SHUTDOWN_RESET_THRESHHOLD value specifies the number of commands that must be successfully processed after one error induced shutdown before the logoff counter is reset. A successfully processed command is one that doesn't result in a shutdown condition, it does not necessarily result in a zero return code. The minimum recommended value is 2; the maximum recommended is 5.

3 Shutdown events are handled in pairs. The first shutdown, or any odd numbered shutdown, causes a re-IPL and the failing command is retried. The second shutdown, or any even numbered shutdown is *probably* the retry of the failing command. (The lower the value for the RESET threshold, the more likely this is true; a RESET value of 1 ensures this.) Even numbered shutdowns cause either a re-IPL or a LOGOFF after purging the command from the retry queue.

- After the specified number of shutdown events have occurred, a CP LOGOFF command is entered if running disconnected. If running connected the system will continue to re-IPL.
- The SHUTDOWN_REIPL_COMMAND value specifies the CP command to be entered in order to accomplish the re-IPL. The AUTOCR keyword is required. Any other keywords that are valid on the IPL command may also be used if appropriate for your system environment.
- The DISK_SPACE_THRESHHOLD_*xxxx*= value specifies warning and shutdown limitations on DASD space usage. When DASD space usage reaches the warning threshold, hourly messages will be broadcast to the support staff asking for assistance. When usage reaches the shutdown threshold, the DirMaint service machines will log themselves off.

Step 3. Select Security and Auditing Characteristics

Next, configure DirMaint to work with your ESM (if one is installed), and enable other security related options.

```
1    INTENDED_TCB_LEVEL= D | C1 | C2 | B1
2 /  ESM_PASSWORD_AUTHENTICATION_EXIT= DVHDA0 MODULE
3 /  SPOOL_FILE_SECLABEL= SYSLOW
4 /  DISK_CLEANUP= NO | YES
4 /  CYL0_BLK0_CLEANUP= NO | YES
5 /  MESSAGE_LOGGING_FILETYPE= TRANSLOG
5 /  MESSAGE_LOGGING_FILTER_EXIT= DVHXLF EXEC
5 /  MESSAGE_LOG_RETENTION_PERIOD= 3  (MONTHS)
5 /  ESM_LOG_FILTER_EXIT= DVHXLF EXEC
6 /  ESM_LOG_RECORDING_EXIT= DVHESMLR EXEC
7 /  SHUTDOWN_MESSAGE_FAILURE= LOGOFF | REIPL
8 /  POSIX_UID_AUTO_RANGE= lowerbound | upperbound
9    ADD_COMMAND_PROCESSING= FULL | SHORT
10   PURGE_COMMAND_PROCESSING= FULL | SHORT
11   SPOOL_CONSOLE= START FOR * CLASS 0 HOLD
```

Figure 16. Selecting Security and Auditing Characteristics

1 The INTENDED_TCB_LEVEL= entry is not checked by DirMaint Release 5.0. It is present to document to an auditor or your Designated Approval Authority (DAA) what criteria the DirMaint portion of your system is intended to meet. The value you choose may impose restrictions upon the valid choices available for the other security related configuration parameters.

2 If you have an ESM installed, you will need to remove the slash (/) from the ESM_PASSWORD_AUTHENTICATION_EXIT statement, and perhaps change the routine name. The IBM-supplied routine, DVHDA0 MODULE, is ready for use with RACF or other ESMs issuing the same subcode X'04' of DIAGNOSE code X'A0' for the password verification interface. Use of an ESM with logon password control is required for class C2 or class B1 TCB operation.

3 If your system is running as a class B1 TCB with MAC, and you followed the directions in Appendix A, “External Security Manager Considerations” on page 199, you will find that spool files sent by DIRMAINT will have a SECLABEL of SYSHIGH and will be inaccessible by general users. To make DIRMAINT send these spool files with a more suitable SECLABEL, remove the slash from the SPOOL_FILE_SECLABEL statement. You may choose a different SECLABEL however, it should be one available to all or most users.

4 For IVP, you will obtain better response time by leaving the DISK_CLEANUP and CYL0_BLK0_CLEANUP statements with the slash prefix.

Note: With these two statements commented out by the slash prefix, or with the default value of NO, any minidisk space that is deleted from one user and then assigned to another user will usually contain any data left there by the first user. This violates the Object Reuse criteria for a C1, C2, or B1 TCB. To prevent unauthorized access to residual data, the slash should be removed from these two statements and the keyword changed from NO to YES before enabling the DirMaint DASD management functions. For more information on enabling DASD management, see Chapter 6, “DASD Management” on page 87.

During IVP you will probably be entering quite a few DirMaint commands in a fairly short period of time, and you will probably be creating minidisks, putting a few nonsensitive scratch files on them, and deleting them. Use of the defaults during this IVP activity is satisfactory. When you have completed the IVP, IBM recommends that you remove the leading slash from the DISK_CLEANUP statement and change the keyword value to YES if your users have minidisks containing sensitive information. Depending on the nature of the minidisks your system may have defined beginning on cylinder 0 of a CKD volume or block 0 of an FB-512 volume, you should consider changing the CYL0_BLK0_CLEANUP entry to YES also.

Note: The statement DISK_CLEANUP= YES will **not** clean a minidisk that overlaps another minidisk; although it will clean a minidisk that is overlapped by another minidisk. Thus deleting a full volume minidisk containing many other minidisks will not harm any of the other minidisks, but deleting one of the many smaller minidisks will clean that small minidisk without harm to the full volume minidisk.

5 IBM recommends use of the MESSAGE_LOGGING_FILETYPE during IVP. Simply remove the slash prefix from this statement to enable logging.

As you review the entries in the TRANSLOG files, you may find many messages that are of no interest to you. If so, you may use the MESSAGE_LOGGING_FILTER_EXIT, the IBM-supplied exit, to suppress future collection of these messages.

A MESSAGE_LOG_RETENTION_PERIOD of 3 months is suggested. This value may need to be adjusted up or down, depending on the amount of DirMaint activity on your system and the size of the minidisk you have allocated for the transaction history files.

The MESSAGE LOG RETENTION PERIOD value in the IBM supplied CONFIG SAMPDVH file specifies that the interval may be MONTHS (the default) or DAYS (or DAY). If:

- Set to DAYS or DAY, the TRANSLOG file will be closed daily, using a file type of TL*yymmdd*
- Specified as anything other than DAYS or DAY, the TRANSLOG file will continue to be closed monthly, using a file type of TLOG*yymm* there is no change
- The interval is specified as DAYS or DAY, the valid range is between 1 and 730 days, and a non-numeric value will be treated as 90 days.
- The interval is not DAYS or DAY, the valid range is 1 to 24 months, and a non-numeric value will be treated as 3 months.

6 If you have an ESM installed with the necessary capabilities, you may choose to record DirMaint activity in the ESM log files. To do so, enable the ESM_LOG_RECORDING_EXIT. The IBM-supplied exit DVHESMLR EXEC, calls the DVHRACLR MODULE to record activity into the RACF log. If you are using another ESM, you may tailor either the DVHESMLR EXEC to call the appropriate logging module for your ESM, if supplied by your ESM vendor, or you may tailor the DVHRACLR ASSEMBLE file to communicate with your ESM using the interfaces documented by the ESM.

Use of an ESM with application audit logging capability is required if your system is intended to meet either class C2 or class B1 TCB criteria, and the ESM_LOG_RECORDING_EXIT must be enabled.

As you review the entries in the ESM log, you may find many DirMaint messages that are of no interest to you. If so, you may use the ESM_LOG_FILTER_EXIT to suppress future collection of these messages. If your system is intended to meet either class C2 or class B1 TCB criteria, you must obtain the approval of your DAA for any messages you choose to filter out.

7 It is possible that an error condition may arise in the message handling routines. If your system is not required to meet any TCB criteria, your best alternative is to set the SHUTDOWN_MESSAGE_FAILURE= entry to REIPL. If running disconnected, this will cause the service machines: DIRMAINT, DATAMOVE, DIRMSAT to re-IPL CMS and attempt an automatic restart. If your system is intended to meet any TCB criteria, DirMaint must shut itself down if unable to issue or log a message for any reason. This is done by setting the SHUTDOWN_MESSAGE_FAILURE= entry to LOGOFF. If running disconnected, this will cause the service machines to LOGOFF if and when an error is encountered in the message handling routines. In either case, if running logged on, the service machine will not LOGOFF, but will re-IPL CMS, run through the PROFILE EXEC to access all necessary disks, then wait at CMS Ready for manual problem diagnosis and restart.

Example—Using the SHUTDOWN_REIPL_COMMAND

Enter:

```
CP IPL 190 PARM AUTOCR NOSPROF FILEPOOL SERVERX
```

The SHUTDOWN_MESSAGE_FAILURE value identifies the action to be taken if the failure causing the shutdown occurred in the message handler. If your system is intended to meet TCB criteria, the correct value is LOGOFF. Otherwise you may choose either LOGOFF or REIPL for this value. Because of the TCB relevance, the SHUTDOWN_MESSAGE_FAILURE

entry is located with the other security related configuration parameters.

8 The POSIX_IUD_AUTO_RANGE= entry specifies a UID range for use during automatic assignment of POSIX UIDs to users during DIRM ADD and DIRM POSIXINFO operations:

- The valid range for POSIX UIDS is 0 to 4294967295. This field will be considered null if nonnumeric data is provided in this field:
  - lowerbound < 0
  - upperbound > 4294967295
  - upperbound < lowerbound
- When this setting is found, DirMaint operations will sequentially assign a UID to the target user. The next UID to use is saved in the POSIXUID CONTROL file on the DIRMAINT server's primary directory disk (1DF by default).
- DIRM ADD operations done with directory entries that already have a POSIX UID are not affected by this setting.
- If the range is exhausted, the operation will continue but a warning message will be issued indicating that the automatic addition of a UID did not take place.
- DIRM POSIXINFO, DIRM ADD, and DIRM REPLACE operations that explicitly set a UID that falls between lowerbound and upperbound will receive a warning message and operation will continue. DIRM

REPLACE will only receive the warning message if the UID changes during the DIRM REPLACE operation.

- Should your installation exhaust the range of UIDs established, it is recommended that the upperbound be advanced or that the entire range be advanced beyond the exhausted range. This is due to the fact that DirMaint does not catalog used UIDs as it assigns them. It simply advances through the range assigning each UID sequentially.
- If you chose to use a lower range you should delete the POSIXUID CONTROL file after establishing the new range. This will cause DirMaint to start at the new lowerbound. Care should be taken to ensure that the new range does not overlap any used UIDs as DirMaint sequentially assigns them and does not check to ensure the UID has not been previously used.

9 **and** **10** These entries support the PRIVILEGE CONTROL:

- ADD_COMMAND_PROCESSING
- PURGE_COMMAND_PROCESSING

Notes:

1. These entires may be given as either FULL or SHORT.
2. The ADD_COMMAND_PROCESSING and PURGE_COMMAND_PROCESSING entires could also affect the OBJECT REUSE policy. Use of the ADD_COMMAND_PROCESSING SHORT will bypass the extent overlap checking for the ADD command and use of the PURGE_COMMAND_PROCESSING= SHORT will bypass the disk cleanup. Both options must be set or defaulted to FULL, for the TCB levels C1, C2, or B1.

If FULL is specified or defaulted then all LINK and MDISK statements are removed from the directory entry being added or purged, and are separately processed as a batch file. Full authentication and authorization checking is done for all commands in the batch file, and all appropriate exit routines are called. The ADD and PURGE commands may be left in command set A, allowing use of the ADD and PURGE commands to be delegated more widely.

If SHORT is specified, then all LINK and MDISK processing for the ADD and/or PURGE commands are processed in line, like REPLACE, with no authorization checking performed for the use of the LINK, AMDISK, or DMDISK commands that would have been included in the batch file. Calls to the following exits are bypassed for ADD and PURGE processing:

- DASD_AUTHORIZATION_CHECKING_EXIT
- LINK_AUTHORIZATION_CHECKING_EXIT
- MINIDISK_PASSWORD_SYNTAX_CHECKING_EXIT

This makes the ADD and PURGE commands comparable to REPLACE in privilege, and requires them to be moved from command set A to command set S along with REPLACE; unless use of command set A is not delegated to anyone who does not already have REPLACE authority. Use of SHORT must be approved by your auditor or DAA; and any delegation of ADD or PURGE authority may require approval of your auditor or DAA also.

Note: When SHORT is specified, the FULL processing is done if any of the these exits are used:

- DASD_OWNERSHIP_NOTIFICATION_EXIT

- LINK_NOTIFICATION_EXIT
- MINIDISK_PASSWORD_NOTIFICATION_EXIT

11 The SPOOL_CONSOLE= entry identifies the USER id to receive the console spool files from the various DirMaint service machines. The data following the equals sign (=) is usually the command syntax after the CP SPOOL CONSOLE command.

Note: When the DIRM GETCONSOLE command is issued, a copy of the spool file is sent to the command issuer and a copy is sent to the user ID identified. If the user ID's are the same, only one copy is sent. The same action will occur if the DIRM GETCONSOLE command is to retrieve a spool file residing in the virtual printer.

Step 4. Select Password Control Characteristics

If your system has an ESM installed, the ESM probably controls logon passwords and minidisk access. If so, you may keep the defaults for the following entries, or you may delete them from the CONFIG* DATADVH file(s). If your system does not have an ESM installed, or if by some chance your ESM does not control either logon passwords or minidisk access, then you need to select your password control characteristics.

```
1 PW_INTERVAL_FOR_GEN= 0 0
2 PW_INTERVAL_FOR_PRIV= 0 0
3 PW_INTERVAL_FOR_SET=
4 PW_WARN_MODE= MANUAL | AUTOMATIC
5 PW_LOCK_MODE= MANUAL | AUTOMATIC
6 PW_NOTICE_PRT_CLASS= A | 1 letter A-Z | NONE
7 PW_NOTICE_RDR_CLASS= A | 1 letter A-Z | NONE
8 MDPW_INTERVAL= 0 0
9 LINK_MAX_INDIRECT=
```

Figure 17. Selecting Password Control Characteristics

1 The PW_INTERVAL_FOR_GEN= entry indicates how long a general user may keep a given logon password (in days) before DirMaint begins sending password expiration warning notices, and how long the user may keep that password before having the password changed to NOLOG to deny access to the system. The default values are 0 and 0, indicating that notices are not sent and users are not locked out. If nonzero, the number of days before warning must be less than the number of days before lockout.

2 The PW_INTERVAL_FOR_PRIV= entry identifies how long privileged users may keep a given logon password (in days) before DirMaint begins sending password expiration warning notices, and how long the user may keep that password before having the password changed to NOLOG to deny access to the system. The default values are 0 and 0, indicating that notices are not sent and users are not locked out. If nonzero, the number of days before warning must be less than the number of days before lockout.

All users are considered to be general users, unless a CHECK_USER_PRIVILEGE_EXIT= record identifies an exit routine that determines which users are privileged. For more information, see “Check User Privilege (DVHXCP)” on page 165.

3 Unless specified otherwise, a password that is set by using an: ADD, CHNGID, or SETPW command; will be valid for the full duration specified on the respective PW_INTERVAL_FOR_GEN= or PW_INTERVAL_FOR_PRIV= entry. If you choose to make users change their password in a shorter time after having their password set by the administrator, you may specify an alternate lockout period by using the PW_INTERVAL_FOR_SET= entry. The first value specifies the number of days a password is valid following one of the commands that set the password for a general user, the second value specifies the number of days a password is valid for a privileged user.

4 The PW_WARN_MODE= entry identifies whether DirMaint will send password warning notices automatically at the time scheduled in the DIRMAINT DATADVH file (AUTOMATIC), or whether password warning notices are sent only when the administrator enters the PWMON MONITOR command (MANUAL).

5 The PW_LOCK_MODE= entry identifies whether DirMaint will change expired passwords to NOLOG automatically at the time scheduled in the DIRMAINT DATADVH file (AUTOMATIC), or whether expired passwords are only changed to NOLOG when the administrator enters the PWMON LOCKOUT command (MANUAL).

Note

Before setting PW_LOCK_MODE= AUTOMATIC, you should ensure that:

- PW_WARN_MODE= AUTOMATIC
- The PW_INTERVAL_FOR_GEN= and PW_INTERVAL_FOR_PRIV entries specify reasonable periods for your installation,
- The disconnected service machines have a surrogate designated in the PWMON CONTROL file to receive their notices,
- Critical system resource user ID's for example, the DIRMAINT service machine itself, MAINT, OPERATOR, PVM, and RSCS are listed in the PWMON CONTROL file as being exempt from lockout.

For more information on the PWMON CONTROL file, see “PWMON CONTROL” on page 62.

If you comply with these rules, your system should be safe from becoming unusable through having all user ID's on your system getting their password set to NOLOG.

6 The PW_NOTICE_PRT_CLASS= entry identifies the spool file class to be used for printed password expiration notices. A value of NONE indicates that password expiration notices will not be printed.

7 The PW_NOTICE_RDR_CLASS= entry identifies the spool file class to be used for password expiration notices sent to a user's reader. A value of NONE indicates that password expiration notes are not to be sent to the user's reader.

Note: This entry has no effect on systems prior to CMS level 11 on VM/ESA 1.2.2 (CMS level 7 on VM/ESA 1.1.5 370 feature and CMS level 10 on VM/ESA 1.2.1).

8 The MDPW_INTERVAL= entry determines how old a minidisk password may become before entering a WARNING period, and before entering the EXPIRED period. The first value must be less than the second value, the second value must be less than or equal to 373 (one year plus one week

grace), use of 0 0 disables checking. DirMaint takes no action for old passwords, but does flag them appropriately on the MDAUDIT report.

9 The LINK_MAX_INDIRECT= entry determines how deeply links to links may be nested. If set to zero, directory links are disabled. If set to 1, a lINK to an MDISK is allowed. If set to 2 or more, a LINK to a LINK is allowed. If left blank, the default value is the same as the CP limits: 2 for a 370 feature system and 50 for an ESA feature system.

DIRMAINT DATADVH

This DIRMAINT WAKEUP TIMES file controls time-driven events that take place in the virtual machines. A sample of this file (RECFM V) is supplied with the product code. As part of DIRMAINT's initialization, it will be copied to the virtual machine's A-disk. The file name will always be called DIRMAINT, regardless of the user ID of the DIRMAINT service machine.

DIRMAINT WAKEUP Times File

```
1 ==/==/== 00:00:05 00/00/00 CMS EXEC DVHNDAY
2 ==/==/== 00:01:00 00/00/00 CMS EXEC DVHDAILY
3 ==/==/== 00:02:00 00/00/00 BACKUP NOTAPE
4 ==/==/== 00:03:00 00/00/00 ELINK    CLEAN    ALL
5 ==/==/== +01:00:0 00/00/00 CMS EXEC DVHOURLY
6 12/31/94 +01:00:0 00/00/00 DIRECT
7 12/31/94 01:00:00 00/00/00 MDAUDIT  ALLCHECK AUTOMAIL
8 12/31/94 02:00:00 00/00/00 PWMON    MONITOR +15
9 12/31/94 12:00:00 00/00/00 BACKUP   TAPE BOT DIRMTAPE  DVHBCK
10 ==/==/== 23:59:00 00/00/00 CP SLEEP 2 MIN
```

These notes will help you with your DIRMAINT WAKEUP Times file.

1 The DVHNDAY EXEC is run after Midnight, every day. This is an IBM-supplied housekeeping routine. IBM recommends running this EXEC at this time. If you choose to retain your console spool files for only four or five days rather than the default, nine days, you can schedule a second invocation at or near Noon.

2 The DVHDAILY EXEC is run after Midnight each day, after the DVHNDAY EXEC has been run. This is an IBM-supplied housekeeping routine. IBM recommends that this routine be run at least once per day, or more often if you choose. You may adjust the time or times to suit your needs.

3 The DIRM BACKUP NOTAPE command is processed each day, after the DVHDAILY EXEC has been run. If you have not allocated space for the primary directory backup disk or shared file system directory, you should delete this entry. IBM recommends that you do allocate space for a primary directory backup disk, and that you run the BACKUP command daily. You may adjust the time to suit your needs. Ideally, this should be scheduled to occur when users are least likely to be issuing DirMaint commands and waiting for the result.

Note: Users may enter commands while the backup is processed, but those commands will not be processed until the backup is complete, the length of the delay depends upon the size of your directory.

4 An DIRM ELINK CLEAN ALL command is processed once each day. When a user has made too many attempts to use the DIRM LINK command with incorrect passwords, that user will be prevented from using the DIRM LINK command until a site specified number of days has elapsed. The ELINK CLEAN ALL command checks for users whose ability to use DIRM LINK can be re-enabled.

5 The DVHOURLY EXEC is run every hour, every day. This is an IBM-supplied housekeeping routine.

6 An DIRM DIRECT command is automatically performed every hour. This places your directory changes online. If you are running with ONLINE= IMMED specified in your CONFIG DATADVH file, you may omit the DIRECT line. If your installation is a large processing center you may want to replace this line with a specific schedule of times throughout the day. For example:

```
==/==/== 00:05:00 00/00/00 DIRECT
==/==/== 06:00:00 00/00/00 DIRECT
==/==/== 12:00:00 00/00/00 DIRECT
==/==/== 18:00:00 00/00/00 DIRECT
```

The IBM-supplied file uses a date of 12/31/94 for the DIRECT entries to disable DirMaint from placing changes to the source directory online. This is recommended for starting IVP. When you are satisfied with your DirMaint tailoring, you may change the date to ==/==/== for production, unless you want all directory changes placed online immediately. If you want directory changes placed online immediately, leave the date on the DIRECT entries set to 12/31/94, and change CONFIG DATADVH file to ONLINE= IMMED.

7 An implicit DIRM MDAUDIT command is processed once each month. This command checks your MDISK statements to ensure that minidisk passwords are in compliance with your site policy. Depending upon the size of your source directory, this IBM recommends that it be scheduled at a time of day when users are least likely to be issuing DirMaint commands and waiting for the result.

Note: Users may enter commands although the MDAUDIT is being taken but the commands will not be processed until the MDAUDIT is complete; the length of the delay depends upon the size of your directory.

The IBM-supplied file uses a date of 12/31/94 for the MDAUDIT entry to disable DirMaint from sending notices about expired minidisk passwords before completion of the IVP.

Note: If you have an ESM, such as RACF, installed and controlling minidisk links on your system, then you may be able to delete this MDAUDIT entry. Be aware of the following:

- If your ESM is functioning with DISKP=DEFER (for RACF, or the equivalent for your particular ESM), minidisk passwords are useful for controlling write and multiple access to minidisks above and beyond the Discretionary Access Control (DAC) and perhaps even the mandatory access control (MAC) provided by the ESM.
- Whether your ESM is functioning with DISKP=ALLOW or DISKP=DEFER (for RACF, or your ESM's equivalent), minidisk passwords can still be used to establish a directory link to a minidisk, unless that capability has been suppressed by use of the LINK_AUTHORIZATION_EXIT.

8 The DIRM PWMON MONITOR command is processed once each weekday, Monday through Friday. Depending upon the size of your source directory, IBM recommends that it be scheduled at a time of day when users are least likely to be issuing DirMaint commands and waiting for the result.

Note: Users may enter commands although the PWMON MONITOR is being taken but those commands will not be processed until the PWMON command is complete; the length of the delay depends upon the size of your directory.

The IBM-supplied file uses a date of 12/31/94 for the PWMON entry to disable DirMaint from sending notices about expired logon passwords before completion of the IVP. For more information on changing the date for automatic minidisk password monitoring, see "The Date Field (Columns 1–8)" on page 234.

Note: If you have an ESM, such as RACF, installed and controlling logon passwords on your system, then you should delete this PWMON entry, or leave it disabled with the 12/31/94 date.

9 Once each week, an automatic DIRM BACKUP TAPE command is performed. This is optional. If used, this event should be scheduled on a day of the week and at a time of day when:

- Your site has operators and tape librarians on duty
- When users are least likely to be issuing DirMaint commands and waiting for the result.

Although there may be no time that satisfies both criteria, the IBM-supplied default has selected Noon on Friday.

Notes:

1. Although this is another relatively lengthy process, depending on the size of your source directory, the delay in user responses includes only the time for tape positioning and actually dumping files from disk to tape. DirMaint is responsive to user requests although waiting for a tape to be mounted.
2. The BACKUP TAPE option requires BACKUP_TAPE_MOUNT_EXIT routine. For more information, see "Backup Tape Mount (DVHXTP)" on page 186.

The IBM-supplied file uses a date of 12/31/94 for the BACKUP TAPE entry to disable DirMaint from making tape backups. If you choose to enable automatic tape backup processing, you may change the date to ==/==/== for daily tape backups. For more information on changing the date daily see "The Date Field (Columns 1–8)" on page 234.

10 An event is **REQUIRED** to be scheduled before Midnight each day, with an action that will not be completed until after Midnight. This is necessary to ensure that events scheduled for the next day are recognized. The omission of this entry causes the service machine to hang up and never wake up as scheduled and may or may not respond to incoming user requests. The action performed is arbitrary; you may schedule one of the BACKUP, DVHDAILY, or DVHNDAY events at this time if you are sure the action will not complete until after Midnight.

Note: Do not try to schedule two or more events at or near this specific time of day. If the first does not complete until after Midnight, the other event may not be processed at all.

For more information, see "The WAKEUP Times File" on page 233.

DVHNAMES DATADVH

The DVHNAMES DATADVH file becomes the NAMES file for each of the DirMaint service machines. It is used for sending messages to designated users when events occur that require their action or awareness.

This file is in a standard CMS NAMES file format, and the file name is RECFM V. Each entry contains:

Table 6. Tags in the CMS NAMES File

Tag	Function
:NICK.	Identifies the nickname for the assigned user ID/node ID pair or to a distribution list.
:USERID.	Identifies the user to be notified.
:NODE.	Identifies the node ID where the user is located. Alternatively, a nickname may refer to a distribution list of other nicknames.
:LIST.	Specifies the nicknames in the distribution list.

These entries are required in your DVHNAMES DATADVH file for the following nicknames:

Table 7 (Page 1 of 2). DVHNAMES DATADVH Nickname Entries

Nickname	Function
DVHALL	The distribution list to be notified when DirMaint starts up or shuts down. This distribution list usually a list of the other distribution lists, possibly excluding the DVHCERT list. `:nick.DVHALL :list.DVHCERT DVHHELP DVHOPER DVHPWMON DVHSUPT`
DVHCERT	The distribution list to be notified when a situation occurs that **may** indicate that a hacker is attempting to gain unauthorized access to your system.
DVHHELP	The distribution list to be notified when the DirMaint service machine becomes available to respond to user initiated transactions and when the DirMaint service machine encounters a problem or begins a lengthy task that makes it unavailable to respond to user requests.
DVHOPER	The distribution list to be notified when events occur that require physical interaction with the DirMaint service machines, such as mounting a backup tape.
DVHPWMON	The distribution list to be notified when the PWMON command has completed, to alert the appropriate people that the data files are ready and available for manipulation.

Table 7 (Page 2 of 2). DVHNAMES DATADVH Nickname Entries

Nickname	Function
DVHSUPT	The distribution list to be notified when the DirMaint service machine encounters a problem within the DirMaint product code, the DirMaint installation, or the DirMaint tailoring that renders DirMaint unable to complete the transaction requested by the user. DirMaint will usually continue to run, however certain functions may be limited or nonoperational until the problem is resolved.

Depending upon your system configuration, certain events may happen quickly while others may take a longer to complete. When these events occur, you may have DirMaint notify, the:

- Users on the system
- Key staff personnel
- Server to perform the task quietly.

The entries in DVHNAMES for these events are:

Table 8. DVHNAMES Event Entries

Event Name	Function
DVHDAILY	The distribution list to be notified when potentially time-consuming events begin and end their daily run. This distribution list is usually the same as DVHHELP, for example: `:nick.DVHDAILY :list.DVHHELP` This includes the DVHNDAY, DVHDAILY, and BACKUP events.
DVHDRCT	The distribution list to be notified when a potentially time-consuming call to the DIRECT or DIRECTXA command is needed to update the object directory from the current source directory file. This distribution list is usually the same as DVHHELP, for example: `:nick.DVHDRCT  :list.DVHHELP`
DVHOURLY	The distribution list to be notified when a potentially time-consuming task begins or ends its periodically scheduled processing. This distribution list is usually the same as DVHHELP, for example: `:nick.DVHDRCT  :list.DVHHELP`

Notes:

1. If you omit one or more of these entries from your DVHNAMES DATADVH file, your DirMaint service machine console files will contain CP error messages about user DVH*xxxxx* not logged on. For the optional entries, these messages can be ignored.
2. If you have a user ID on the system that is the same as one of these nicknames, then this user ID will be getting all of these messages.

DIRMMAIL SAMPDVH

The file is DIRMMAIL SAMPDVH file is a sample for a DIRMAINT NEWMAIL file. The IBM supplied sample provides a description of the differences between Release 4 and Release 5 that may be of interest to the general user community.

DVHLINK EXCLUDE

The file is DVHLINK EXCLUDE file is maintained by using the DIRM USEROPTN command. This file contains a listing of the global or public minidisks for example, the MAINT 190, 19D, and 19E disks for which links to that disk should be omitted from the DVHLINK FILE. The DVHLINK EXCLUDE file is RECFM V, and resides on the primary directory file mode. If you are using a secondary directory disk or directory, an identical copy of the DVHLINK EXCLUDE file will be maintained for you.

Note: If a minidisk is listed in the DVHLINK EXCLUDE file, links to that disk are omitted from the DVHLINK FILE file. Links that are omitted from the DVHLINK FILE file are not:

- Included in the output from DIRM REVIEW
- Changed to point to the new device address when a CHVADDR is done for the minidisk
- Changed to point to the new user ID when a CHNGID is done for the user ID owning the minidisk
- Changed to point to the new user ID and address when a TMDISK is done for the minidisk
- Deleted when the minidisk is deleted or when the user ID owning the minidisk is purged
- Deleted when the owner of the minidisk uses the DIRM DLINK command

If you need any of these operations to process an excluded minidisk, the minidisk must first be removed from the DVHLINK EXCLUDE file (preferably using the DIRM USEROPTN LINKS EXCLUDE CANCEL command), and the DASD Management control files rebuilt by using the DIRM RLDEXTN command.

Customers migrating from DirMaint Release 4 can use the DVHMIGR8 EXEC to convert the LINKS EXCLUDE file to the following format:

Columns 1-8 A minidisk owner's user ID.
Column 9 Blank.
Columns 10-17 The minidisk's system affinity, or an asterisk.
Column 18 Blank.
Columns 19-22 The minidisk's virtual address.
Column 23 Blank.
Columns 24-27 The link modes to be excluded. The valid link modes are:

- **R** Read links (R and RR) should be excluded.
- **RW** Read and Write links (R, RR, W and WR) should be excluded.
- **RWM** Read, Write and Multi Write links (all except S and E links) should be excluded.
- **S** Stable links (any link using the S suffix) should be excluded.
- **SR** Stable Read links (any read link using the S suffix) should be excluded.

SRW Stable Read and Write links (any read or write link using the S suffix) should be excluded.
SRWM Stable Read, Write and Multi Write links (all except Exclusive links) should be excluded.
E Exclusive or Stable links (any link using the S or E suffix) should be excluded.
ER Exclusive or Stable Read links (any read link using the S or E suffix) should be excluded.
ERW Exclusive or Stable Read and Write links (any read or write link using the S or E suffix) should be excluded.
ERWM Exclusive or Stable Read, Write and Multi Write links (all links) should be excluded.
ALL All links should be excluded.

Note: IBM recommends only using R. Use of an exit routine is suggested to enforce compliance. For more information, see "Link Authorization (DVHXLA)" on page 171.

Column 28 Blank.

Note: In actuality, the file is composed of blank delimited fields. The relative position of the fields is critical, the specific columns for those fields is not critical. If you look at this file after DIRMAINT has been in operation for any length of time, you may find more than one blank between fields. You do not need to correct the file; if you are making an addition to the file, just align the fields under the existing entries.

PWMON CONTROL

The PWMON CONTROl file is maintained using the DIRM PWMON GET CONTROL, RECEIVE, XEDIT PWMON CONTROL, and DIRM PWMON REPLACE CONTROL commands. It contains a list of user ID's whose passwords are exempt from being changed to NOLOG when they expire (such as the OPERATOR, MAINT, and the DIRMAINT user ID itself), and a list of disconnected service virtual machines whose passwords may be allowed to expire but need warning notices sent to a human being for intervention rather than to the service machine itself. The file is RECFM V, and resides on the primary directory file mode. If you are using a secondary directory disk or directory, an identical copy of the PWMON CONTROL file will be maintained for you.

Customers migrating from DirMaint Release 4 can use the DVHMIGR8 EXEC to convert the PWMON CONTROL file to the following format:

Columns 1-8 A local user ID.
Column 9 Blank.
Columns 10-12 The keyword YES if the user ID is subject to lockout, or the keyword NO if the user ID is exempt from lockout. Any other value is treated the same as YES.
Column 13 Blank.
Columns 14-21 The alternate user ID to be notified in place of the local user ID.
Column 22 Blank.
Columns 23-30 The node ID of the alternate user to be notified.
Column 31 Blank.

Note: In actuality, the file is composed of blank delimited fields. The relative position of the fields is critical, the specific columns for those fields is not critical. If you look at this file after DIRMAINT has been in operation for any length of time,

you may find more than one blank between fields. You do not need to correct the file; if you are making an addition to the file, just align the fields under the existing entries.

RPWLIST DATA

The RPWLIST DATA file contains a list of logon passwords that are not allowed to be used on your system. When the DIRECT or DIRECTXA program is run to put the source directory on-line, any user with one of these restricted passwords as a logon password in the source directory will have it changed to NOLOG in the object directory. The user will be unable to logon to the system until the password is changed to a value not in the RPWLIST DATA file.

Notes:

1. The VM/ESA product tape contains a sample of this file. This sample file contains many of the sample passwords published in IBM documents. You should use this as a starting point for your tailoring. You will want to add obvious passwords such as your company's name or any password that you think unauthorized persons may know.
2. If you already have a copy of this file for use with your present method of directory maintenance, you should be able to continue using that copy without change.

The RPWLIST DATA file must be a RECFM F LRECL 80 file, with each record in the following format:

Columns 1-8	A character string whose use as a logon password is to be restricted.
Column 9	Must be left blank.
Columns 10-80	Comments.

Note: This particular file is **NOT** composed of blank delimited fields. The format of this file is dictated by the DIRECT MODULE. For more information on the DIRECTXA MODULE for ESA feature systems, see the *z/VM: CP Command and Utility Reference*. The file is not altered by DirMaint in any way, and you must maintain it in exactly the format documented in the VM publications.

Note: If you rename or erase the RPWLIST DATA file, a warning message will be issued by the DIRECT or DIRECTXA program and passwords will not be checked. However, the object directory will be updated.

IBM recommends placing this file on the primary directory file disk, where it will be automatically shadowed to the secondary directory file disk if you have defined one.

AUTHFOR CONTROL

The AUTHFOR CONTROL file is maintained using the AUTHFOR and DROPFOR commands. It contains a list of user ID's who are authorized to act for other user ID's, and the privileges that have been delegated to them. The file is RECFM V, and resides on the primary directory file mode. If you are using a secondary directory disk or directory, an identical copy of the AUTHFOR CONTROL file will be maintained for you.

Customers migrating from DirMaint Release 4 can use the DVHMIGR8 EXEC to convert the ASSIGN FILE, and the privilege information from the DIRMAINT DATA file (OWNER, DATAMOVE, DIRM_STAFF, SYS_OPER, DIRM_SUB_STAFF, PW_MONITOR, and SYS_SERVER_UID), to the following format:

Columns 1-8 A target user ID or profile name, or the keyword ALL.

Column 9 Blank.

Columns 10-17 A user ID authorized to act for the target ID.

Column 18 Blank.

Columns 19-26 The network node ID from which the authorized user may submit requests for the target ID. An asterisk allows the authorized user ID to enter commands from any system within the local CSL cluster.

Column 27 Blank.

Columns 28-31 The command level for which the authorized user may submit requests for the target ID. Valid values are 140A or 150A. A command level of 140A allows the authorized user to enter commands using DirMaint Release 4 compatibility syntax. A command level of 150A allows the authorized user to enter commands using the DirMaint Release 5 full function syntax. You may, and will probably want to, include records for both 140A and 150A command levels for each target ID / authorized user pair.

You must be authorized for both levels when issuing an ADD request in 150A level for a directly entry containing 140A format MDISK statements, and when issuing an ADD request in 140A level for a directory entry that contains 150A format MDISK statements. The 150A format is identified by use of one or more of the keywords: BLKSIZE, LABEL, or PWS; if none of these keywords is present the statement is 140A format.

Column 32 Blank.

Columns 33-68 The command sets identifying which commands the authorized user may use on behalf of the target id. The valid command sets are determined by the command level. The IBM defined default command sets are:

- **A** Non-DASD user directory Administrator commands.
- **D** DASD management user directory administrator commands.
- **G** General user commands.
- **H** Help Desk commands. Allows looking at things without allowing them to be changed.
- **M** Monitoring commands. Allows use of MDAUDIT, PWGEN, PWMON, and SETPW commands.
- **O** Operational support commands, such as BACKUP, NOTAPE, or SHUTDOWN.
- **P** Commands needed by automated administration Programs, such as: CLAS, DFSMS, DSO, IPF, NV/AS, RACF.
- **S** Commands needed by the DirMaint owner and Support programmer.
- **Z** Commands needed by the DirMaint service machines to communicate with each other.

For entries being migrated from the Release 4 ASSIGN FILE, specify command set G only.

Column 69 Blank.

Example

```
*TARGETI ORIGUSER ORIGNODE CMDL CMDSETS
ALL DIRADMIN * 140A ADGHMOPS
ALL DIRADMIN * 150A ADGHMOPS
ALL DIRADMIN DVHTEST1 140A ADGHMOPS
ALL DIRADMIN DVHTEST1 150A ADGHMOPS
ALL DIRADMIN DVHTEST2 140A ADGHMOPS
ALL DIRADMIN DVHTEST2 150A ADGHMOPS
ALL DIRADMIN DVHTEST3 140A ADGHMOPS
ALL DIRADMIN DVHTEST3 150A ADGHMOPS
ALL DIRMAINT * 140A ADGHMOPSZ
ALL DIRMAINT * 150A ADGHMOPSZ
ALL DIRMAINT DVHTEST1 140A ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
ALL DIRMAINT DVHTEST1 150A ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
ALL DIRMAINT DVHTEST2 140A ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
ALL DIRMAINT DVHTEST2 150A ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
ALL DIRMAINT DVHTEST3 140A ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
ALL DIRMAINT DVHTEST3 150A ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890
ALL DIRMSERV GDLVME 140A ADGHMOPS
ALL DIRMSERV GDLVME 150A ADGHMOPS
ALL DVHTEST GDLVM7 140A ADGHMOPS
ALL DVHTEST GDLVM7 150A ADGHMOPS
ALL MARKERME GDLVME 140A ADGHMOPS
ALL MARKERME GDLVME 150A ADGHMOPS
ALL MARKERME GDLVM7 140A ADGHMOPS
ALL MARKERME GDLVM7 150A ADGHMOPS
ALL DRB1 GDLVM7 140A ADGHMOPSZ

ALL DRB1 GDLVM7 150A ADGHMOPSZ
DRB1 DOUGHART GDLVM7 140A ADG
DRB1 DOUGHART GDLVM7 150A ADG
ALL MAINT * 140A ADGHMOPSZ
ALL MAINT * 150A ADGHMOPSZ
ALL SYSMAINT * 140A ADGHMOPS
ALL SYSMAINT * 150A ADGHMOPS
ALL SYSMAINT DVHTEST1 140A ADGHMOPS
ALL SYSMAINT DVHTEST1 150A ADGHMOPS
ALL SYSMAINT DVHTEST2 140A ADGHMOPS
ALL SYSMAINT DVHTEST2 150A ADGHMOPS
ALL SYSMAINT DVHTEST3 140A ADGHMOPS
ALL SYSMAINT DVHTEST3 150A ADGHMOPS
ALL SYSOPER * 140A ADGHMOPS
ALL SYSOPER * 150A ADGHMOPS
ALL SYSOPER DVHTEST1 140A ADGHMOPS
ALL SYSOPER DVHTEST1 150A ADGHMOPS
ALL SYSOPER DVHTEST2 140A ADGHMOPS
```

```
ALL SYSOPER DVHTEST2 150A ADGHMOPS
ALL SYSOPER DVHTEST3 140A ADGHMOPS
ALL SYSOPER DVHTEST3 150A ADGHMOPS
ALL DATAMOVE * 140A GHMADPS
ALL DATAMOVE * 150A GHMADPS
ALL DIRECTOR * 140A GHMADPS
ALL DIRECTOR * 150A GHMADPS
ALL MNTBAT1  *        140A GHMADPS
ALL MNTBAT1  *        150A GHMADPS
ALL DOUGB    *        140A GHMADPS
ALL DOUGB    *        150A GHMADPS
ALL NANCYM   *        140A ADGMPS
ALL NANCYM   *        150A ADGMPS
DOUGHART DOUGHART * 140A ADG
DOUGHART DOUGHART * 150A ADG
DOUGHART MARKERME * 140A G
DOUGHART MARKERME * 150A G
```

Notes:

1. In actuality, the file is composed of blank delimited fields. The relative position of the fields is critical, the specific columns for those fields is not critical. If you look at this file after DIRMAINT has been in operation for any length of time, you may find more than one blank between fields. You do not need to correct the file; if you are making an addition to the file, just align the fields under the existing entries.
2. The default command sets used by DVHMIGR8 are:

   ```
   1......8   10....17  19....26   28..31  33.............68
   TargetId   AuthedId  FromNode   CmdLvl  CmdSets
   --------   --------  --------   ------  ----------------
   owner      owner     nodeid      140A   GS<HMADPOZ>
   owner      owner     nodeid      150A   GS<HMADPOZ>
   ALL        staffid   nodeid      140A   GHMADP
   ALL        staffid   nodeid      150A   GHMADP
   ALL        pwmonitor nodeid      140A   GHM
   ALL        pwmonitor nodeid      150A   GHM
   ALL        substaff  nodeid      140A   GH
   ALL        substaff  nodeid      150A   GH
   operator   operator  nodeid      140A   GO
   operator   operator  nodeid      150A   GO
   datamove   datamove  nodeid      150A   GZ
   dirmsat    dirmsat   nodeid      150A   GZ
   ```

   If the same user ID appears in more than one role (OWNER and STAFF for example), you will need to directly edit the resulting AUTHFOR CONTROL file and combine the entries.

   Placing an asterisk in the FromNode field authorizes the specified user ID for the command set on the local system or, if using an CSE cluster, it authorizes the specified user ID for the command set on any system defined within the CSE cluster.

USER INPUT

The USER INPUT file must be a RECFM F LRECL 80 file, located on the primary directory disk (the 1DF disk, file mode F, by default). If you are migrating from DirMaint Release 4, you should do a DIRM BACKUP (using DirMaint Release 4) and copy or rename the resulting USER BACKUP file from DIRMAINT's 193 disk to USER INPUT on the 1DF disk. The first time you type **DVHBEGIN** to start-up the DIRMAINT machine, the USER INPUT file will be clustered, and mirrored onto the secondary directory disk (the 2DF disk by default) if the secondary directory disk is defined in the DVHPROFA DIRMAINT file.

Note: If a USER DIRECT file is in existence it must be erased so that DVHBEGIN will build a new file from the USER INPUT file.

In general, if the source directory file is acceptable to CP, then it is acceptable to DirMaint. There are a few exceptions:

- Each profile name and user ID in the directory must be unique. You may not have two profiles with the same name, two virtual machines with the same user ID, or a profile and a virtual machine with the same name/user ID.
- Each profile name and user ID must consist of valid CMS file name characters. Not all CMS file name characters are allowed, however. They must be upper case alphabetic letters (A-Z), numeric (0-9), or one of the national language special characters:

  $ = X'5B'

  # = X'7B'

  @ = X'7C'

  Specifically disallowed are lower case alphabetics (a-z) and the underscore character (_=X'6D'), the vertical bar (| = X'4F'), the slant bar (/ = X'61'), and the question mark (? = X'6F').

  Note: The following user ID's are reserved for DirMaint's exclusive use: $DIRCTL$, and $DIRGRP$. DirMaint uses the following nicknames for broadcasting messages: DVHALL, DVHCERT, DVHDAILY, DVHDRCT, DVHHELP, DVHOPER, DVHOURLY, DVHPWMON, and DVHSUPT; IBM recommends that you avoid making real directory entries with these names.
- It addition to the maximum length limitation of 6 characters, the volume identification for all MDISK statements must comply with the same character set rule as PROFILE and USER names. Each volume ID must consist of valid CMS file name characters. Not all CMS file name characters are allowed, however. They must be upper case alphabetic letters (A-Z), numeric (0-9), or one of the national language special characters:

  $ = X'5B'

  # = X'7B'

  @ = X'7C'

  Specifically disallowed are lower case alphabetics (a-z) and the underscore character (_ = X'6D'), the vertical bar (| = X'4F'), the slant bar (/ = X'61'), and the question mark (? = X'6F').
- All MDISKs allocated on a given volume must be of the same device type.

- If your directory contains comment records, they must follow the PROFILE or USER statement to which they apply. DirMaint considers a directory entry to begin with the PROFILE or USER statement (or an external format of the SYSAFFIN statement), and includes all records up to the next PROFILE or USER statement (or external SYSAFFIN statement).

In addition, there are a few rules you should be aware of that affect how DirMaint handles your directory:

- Comments and blank lines within a continued directory statement are completely discarded by DirMaint. For example:

```
POSIXINFO UID 123 GNAME Example ,
This comment line will be deleted by DirMaint.
        FSROOT MyFSRoot ,
*  So will these three comment lines,

*  and the blank line in between,
*  as well as the blank line below.
           IWDIR  'This is my sample IW Directory' ,

           IUPGM  MyPgm
```

  will result in or an equivalent to:

```
POSIXINFO UID 123 GNAME Example FSROOT MyFSRoot IWDIR ,
'This is my sample IW Directory' IUPGM  MyPgm
```

- Comments in a non-System Affinity source directory (a directory that does not use the SYSAFFIN keyword in its internal form) must **follow** the directory statement to which they apply. DirMaint will re-order the sequence in which directory statements are placed, keeping comments associated with the previous real statement. For example, given the following directory segment:

```
MDISK 0197 3380 DEVNO 00AF .....
* This comment is associated with the MDISK 0197 statement.
* So is this comment.
MDISK 0191 3380 DEVNO 00AA .....
* This comment is associated with the MDISK 0191 statement.
* So is THIS comment.
```

  After the directory is manipulated and sorted by address (a selectable option) the same directory segment will appear as follows:

```
MDISK 0191 3380 DEVNO 00AA .....
* This comment is associated with the MDISK 0191 statement.
* So is THIS comment.
MDISK 0197 3380 DEVNO 00AF .....
* This comment is associated with the MDISK 0197 statement.
* So is this comment.
```

  Notes:

  1. When DirMaint removes any directory statement, the comments that follow that statement are **not** removed. This may be of particular interest when processing a CMDISK command, as the MDISK is transferred to the DATAMOVE machine (removing it from the user's directory) and then transferred back to the user (but not associating it with any set of comments).
  2. Blank lines are treated as comments and follow all the same rules.

- All device addresses are expanded to 4 digits when the directory entry is processed by DirMaint. For example, given the following directory segment:

  ```
  LINK HOWLAND 191 0222 RR
  MDISK 0191 3380 DEVNO 00AA .....
  MDISK 197 3380 DEVNO AF .....
  ```

  After any statement in this directory entry has been updated the same directory segment will appear as follows:

  ```
  LINK HOWLAND 0191 0222 RR
  MDISK 0191 3380 DEVNO 00AA .....
  MDISK 0197 3380 DEVNO 00AF .....
  ```

  Note: For VM/ESA 370 feature systems, all device addresses are converted to 4 digits for DirMaint's internal processing; but are converted back to 3 digits in the external form. For example:

  ```
  CONSOLE 9 3215 T
  ```

  becomes

  ```
  CONSOLE 009 3215 T
  ```

  for a VM/ESA 370 feature system, but becomes

  ```
  CONSOLE 0009 3215 T
  ```

  for a VM/ESA feature system.

- To allow compression of statements when dealing with System Affinity, most statements are upper cased and multiple blanks between tokens are eliminated. Comments are always excluded from being upper cased and having excess blanks removed. Other statements may be excluded by setting a pair of control variables in DVHBBSET. The POSIXGROUP and POSIXINFO statements are set to allow mixed case by default. For example, given the following directory segment:

  ```
  PosixInfo UId 42 GName g32g FSRoot  /g32g/42 IWDir Mark
  Link  Howland  191 193 rr
  * Attempt a link   to the test disk

  MDisk 197 3380 Devno af .....
  ```

  After any statement in this directory entry has been updated the same directory segment will appear as follows:

  ```
  POSIXINFO UId 42 GName g32g FSRoot  /g32g/42 IWDir Mark
  LINK HOWLAND 0191 0193 RR
  * Attempt a link   to the test disk

  MDISK 0197 3380 DEVNO 00AF .....
  ```

  Observe that the:

  – LINK and MDISK statement have been completely upper cased and tokenized
  – POSIXINFO statement name has been upper cased but the remainder of the statement has not been changed
  – Blank line and comment were not effected.

- Currently CP allows and ignores multiple copies of NOPDATA in the user directory. DirMaint allows only one copy of NOPDATA per system affinity group; extra copies are discarded.

Overriding and Supplementing DirMaint Commands and Messages

You may add new commands to DirMaint, or modify the way DirMaint processes existing commands, by adding a local override file. For example, the REPLACE request is in command set S by default for command set level 150A. If you wish to make the REPLACE command available to all administrators (command set A), you could modify the existing entry in the 150CMDS DATADVH file, changing the ....S. to A......S.. However, a better way is to create a LCLCMDS DATADVH file containing just that one line:

```
REPLACE      DVHFILE   DVHREP    Y A......S.
```

Where:

REPLACE is the command name, DVHFILE is the file name of the handling routine for the DirMaint code that runs in the user's virtual machine, DVHREP is the name of the handling routine in the DIRMAINT service machine, Y indicates that the invoker's password is required to authenticate the request (unless the invoker has issued a DIRM NEEDPASS NO command), and the A......S. string are the command sets that contain the REPLACE command.

You may have more than one command in an override file, and you may have more than one override file. The file name and file type of your override files are specified on the COMMANDS_140A= and COMMANDS_150A= records in your CONFIG* DATADVH file(s).

For example, if you want to change the MAXSTORE command from command set A (administrator) to command set G (general user), but want the administrator notified if or when any request exceeds 32M. You could use the REQUEST_AFTER_PROCESSING_EXIT= entry to accomplish the task, but it would be called for EVERY request processed. The alternative is to create an override entry in LCLCMDS DATADVH:

```
MAXSTORE     DVHXMIT   LCLMAXST  Y A.G....S.
MAXSTORAGE   DVHXMIT   LCLMAXST  Y A.G....S.
```

and create a LCLMAXST EXEC to handle your requirements:

```
/* LCLMAXST EXEC - Send msg to admininstrator for requests > 32M. */
Address 'COMMAND'
Parse Upper Arg new_maxst .
'GLOBALV SELECT DVH15 GET ORIG_USER ORIG_NODE' ,
                          'SYSAFFIN TARGETID TRACE'
orig_user = Strip(orig_user)
orig_node = Strip(orig_node)
sysaffin  = Strip(sysaffin)
targetid  = Strip(targetid)
trace     = Strip(trace)
Select
   When Pos(' LCLMAXST=',' 'trace' ') <> 0
        Then Parse Var trace . 'LCLMAXST=' trace_optn
   When Pos(' LCL*=',' 'trace' ') <> 0
        Then Parse Var trace . 'LCL*=' trace_optn
   Otherwise trace_optn = ''
End
If trace_optn <> ''
   Then Do
        Say 'LCLMST2161I LCLMAXST called with' new_maxst
        Trace Value trace_optn
        End
'EXEC DVHMAXST'  new_maxst
If rc = 0
   Then Do
        size = Substr(new_maxst,1,Length(new_maxst)-1)
        units = Right(new_maxst,1)
        Select
           When units = 'K' & size < 32*1024
                Then Nop
           When units = 'M' & size < 32
                Then Nop
           Otherwise Do
                     Push 'COMMAND CMS SENDFILE (NOTE'
                     Push 'COMMAND SAVE'
                     Push 'COMMAND INPUT FROM' orig_user'@'orig_node ,
                                  'FOR' targetid 'AT' sysaffin ,
                                  'MAXSTOR' new_maxst 'is > 32M'
                     'EXEC NOTE SYSADMIN'
                     End
        End
If trace_optn <> ''
   Then Say 'LCLMST2162I LCLMAXST ending with RC='rc
Exit rc
```

Overriding and Supplementing DirMaint Messages

You may add new messages to DirMaint, or modify the way DirMaint processes existing messages, by adding a local override file. For example, you might want to modify the PASSWORD_SYNTAX_CHECKING_EXIT (DVHPXV EXEC) to look in your company's internal phone directory to prevent use of first or last names, employee serial numbers, telephone extensions, etc. as passwords; and would want to include messages to explain these violations. Instead of modifying the IBM supplied 150AUSER MSGADVH and/or 150ASERV MSGADVH repositories, you can create your own LCLAUSER MSGADVH and/or LCLASERV MSGADVH repositories. Alternatively, you may choose any file name and file type for your supplemental repositories, as long as they are listed in the

<lang>_USER_MSGS_<cmdl> and <lang>_SERV_MSGS_<cmdl> records in the CONFIG* DATADVH file(s).

Each record in the message repositories begins with a control field, consisting of a 4 digit message number, a 2 digit format number, a 2 digit line number, and a severity indicator. The calling routine identifies the message number and format number to be issued, and DirMaint's message handling routine finds the specified format of the message in the first available repository (in the order specified in the CONFIG* DATADVH file(s)) and issues each line of that message.

You may add new messages to the repository, add new formats of existing message numbers, modify the text of existing message formats (by using the same message and format number in your override file), or completely eliminate a message (by specifying the message and format number in your override file with no text).

The digit line number *zero* is optional for each message format, and provides special override information, most notably the return code. Usually, DirMaint's message handler return code is the same as the message number, unless modified by a *line zero* override.

Chapter 4. Tailoring the DATAMOVE Service Machine

This chapter provides guidance for bringing up the DATAMOVE service machine(s) to format and copy minidisks.

The DirMaint functions are performed by two permanently disconnected virtual machines equipped with an automatic restart facility. The DIRMAINT virtual machine owns and manages the directory; the DATAMOVE virtual machine copies and formats of CMS minidisks. Users invoke DATAMOVE functions by submitting commands to the DIRMAINT virtual machine.

DirMaint supports load balancing among multiple DATAMOVE machines, up to a maximum limit of 9999 DATAMOVE machines.

Defining the DATAMOVE Service Machines

Each DATAMOVE service machine must be defined to CP, to an ESM if one is installed, and to DIRMAINT.

Although it is possible to have the same user ID for service machines on different systems within a cluster, this imposes restrictions on how spool files and SMSGs can be sent. Even with the same user ID, each DATAMOVE service machine requires its own Read/Write disk space. Therefore, IBM recommends that each DATAMOVE service machine have a unique user ID within the cluster. Except for the name of the virtual machine, each one should be defined to CP as described under “Step 1. Define a DATAMOVE Service Machine to DIRMAINT,” below. If an ESM is installed each DATAMOVE machine must be defined to the ESM as described under Appendix A, “External Security Manager Considerations” on page 199.

Step 1. Define a DATAMOVE Service Machine to DIRMAINT

To define a DATAMOVE service machine to DIRMAINT, add an entry to the CONFIG DATADVH file. For more information on the CONFIG DATADVH file, see “CONFIG DATADVH” on page 44. The format is:

```
DATAMOVE_MACHINE= userid nodeid sysaffin
```

Where:

userid
Identifies the DATAMOVE service machine you are logging on.

nodeid
Identifies the DATAMOVE machine you are running on.

sysaffin
Identifies an optional system affinity codes that can be processed. If one DATAMOVE machine can process multiple system affinities, an entry is needed for each system affinity. At least one entry is required using system affinity of asterisk (*).

Example—DirMaint Cluster without CSE Shared Spooling Definitions

```
DATAMOVE_MACHINE= DATAMOV1 DVHTEST1 DVHTEST1
DATAMOVE_MACHINE= DATAMOV2 DVHTEST2 DVHTEST2
DATAMOVE_MACHINE= DATAMOV3 DVHTEST3 DVHTEST3
DATAMOVE_MACHINE= DATAMOV4 DVHTEST1 DVHTEST1
DATAMOVE_MACHINE= DATAMOV5 DVHTEST2 DVHTEST2
DATAMOVE_MACHINE= DATAMOV6 DVHTEST3 DVHTEST3
DATAMOVE_MACHINE= DATAMOV1 DVHTEST1 *
DATAMOVE_MACHINE= DATAMOV2 DVHTEST2 *
DATAMOVE_MACHINE= DATAMOV3 DVHTEST3 *
DATAMOVE_MACHINE= DATAMOV4 DVHTEST1 *
DATAMOVE_MACHINE= DATAMOV5 DVHTEST2 *
DATAMOVE_MACHINE= DATAMOV6 DVHTEST3 *
```

Example—DirMaint Cluster with CSE Shared Spooling Definitions

```
DATAMOVE_MACHINE= DATAMOV1 DATAMOV1 DVHTEST1
DATAMOVE_MACHINE= DATAMOV2 DATAMOV2 DVHTEST2
DATAMOVE_MACHINE= DATAMOV3 DATAMOV3 DVHTEST3
DATAMOVE_MACHINE= DATAMOV4 DATAMOV4 DVHTEST1
DATAMOVE_MACHINE= DATAMOV5 DATAMOV5 DVHTEST2
DATAMOVE_MACHINE= DATAMOV6 DATAMOV6 DVHTEST3
DATAMOVE_MACHINE= DATAMOV1 DATAMOV1 *
DATAMOVE_MACHINE= DATAMOV2 DATAMOV2 *
DATAMOVE_MACHINE= DATAMOV3 DATAMOV3 *
DATAMOVE_MACHINE= DATAMOV4 DATAMOV4 *
DATAMOVE_MACHINE= DATAMOV5 DATAMOV5 *
DATAMOVE_MACHINE= DATAMOV6 DATAMOV6 *
```

Example—DATAMOVE Server on a Stand-Alone System:

```
DATAMOVE_MACHINE= DATAMOVE * *
```

Step 2. Identify the Communication Path

In addition to declaring the existence of each of the DATAMOVE servers, you must identify the communications path between DIRMAINT and each of the DATAMOVE machines, and the return path between the DATAMOVE machines and DIRMAINT.

Note: This is not required for a DATAMOVE machine running on the same node ID as the DIRMAINT machine. This is done with network routing entries in the CONFIG DATADVH file. The format is:

FROM= *fromnode* DEST= *tonick* S= *spoolid* T= *destnode*

Where:

fromnode
Identifies the network *nodeid* where the transaction originates.

tonick
Identifies the nickname for the *nodeid* where the transaction is being sent.

spoolid
Identifies the *userid* of the machine where punch output should be sent to reach the specified destination. If cross system spooling is enabled, this is the *userid* of the DirMaint service machine, either DIRMAINT or DATAMOVE, at that node; otherwise it is the *userid* of an RSCS network machine.

destnode
Identifies the network *nodeid* where the transaction is processed.

Without CSE shared spooling, the routing definitions for the DATAMOVE machines should be the same as those for the DIRMSAT machines.

Example—DirMaint Cluster Without CSE Shared Spooling:

```
FROM= DVHTEST1 TO= T2 S= DIRMNET1 T= DVHTEST2
FROM= DVHTEST2 TO= T1 S= DIRMNET2 T= DVHTEST1
FROM= DVHTEST1 TO= T3 S= DIRMNET1 T= DVHTEST3
FROM= DVHTEST3 TO= T1 S= DIRMNET3 T= DVHTEST1
FROM= DVHTEST2 TO= T3 S= DIRMNET2 T= DVHTEST3
FROM= DVHTEST3 TO= T2 S= DIRMNET3 T= DVHTEST2
```

Where:

DIRMNET1, *DIRMNET2*, **and** *DIRMNET3*
Identifies the *userid* of the private RSCS network machines that carry only DirMaint traffic in our cluster. Use of a private network may be significantly faster than putting your DirMaint traffic on the general RSCS network.

Example—DirMaint Cluster With CSE Shared Spooling:

```
FROM= DVHTEST1 TO= DATAMOV2 S= DATAMOV2 T= DATAMOV2
FROM= DVHTEST1 TO= DATAMOV3 S= DATAMOV3 T= DATAMOV3
FROM= DVHTEST1 TO= DATAMOV4 S= DATAMOV4 T= DATAMOV4
FROM= DVHTEST1 TO= DATAMOV5 S= DATAMOV5 T= DATAMOV5
FROM= DVHTEST1 TO= DATAMOV6 S= DATAMOV6 T= DATAMOV6
FROM= DVHTEST2 TO= DATAMOV1 S= DATAMOV1 T= DATAMOV1
FROM= DVHTEST2 TO= DATAMOV3 S= DATAMOV3 T= DATAMOV3
FROM= DVHTEST2 TO= DATAMOV4 S= DATAMOV4 T= DATAMOV4
FROM= DVHTEST2 TO= DATAMOV5 S= DATAMOV5 T= DATAMOV5
FROM= DVHTEST2 TO= DATAMOV6 S= DATAMOV6 T= DATAMOV6
FROM= DVHTEST3 TO= DATAMOV1 S= DATAMOV1 T= DATAMOV1
FROM= DVHTEST3 TO= DATAMOV2 S= DATAMOV2 T= DATAMOV2
FROM= DVHTEST3 TO= DATAMOV4 S= DATAMOV4 T= DATAMOV4
FROM= DVHTEST3 TO= DATAMOV5 S= DATAMOV5 T= DATAMOV5
FROM= DVHTEST3 TO= DATAMOV6 S= DATAMOV6 T= DATAMOV6
FROM= DVHTEST1 TO= DVHTEST2 S= DIRMAINT T= DVHTEST2
FROM= DVHTEST1 TO= DVHTEST3 S= DIRMAINT T= DVHTEST3
FROM= DVHTEST2 TO= DVHTEST1 S= DIRMAINT T= DVHTEST1
FROM= DVHTEST2 TO= DVHTEST3 S= DIRMAINT T= DVHTEST3
FROM= DVHTEST3 TO= DVHTEST1 S= DIRMAINT T= DVHTEST1
FROM= DVHTEST3 TO= DVHTEST2 S= DIRMAINT T= DVHTEST2
```

Notes:

1. This allows DIRMAINT to run on any of the three systems in the cluster without having to redefine the routings.
2. For a full 16 system cluster with 32 or more DATAMOVE machines, this would not be entirely practical.

Step 3. Define the DATAMOVE Retry Limit

If a DATAMOVE machine is unable to link to a minidisk because a user user is linked to a disk for which a CMDISK command has been issued, or the directory change to transfer the minidisk to DATAMOVE has not been placed online, the FORMAT/COPY/CLEAN request will be placed into the DATAMOVE machine's retry queue. The DM_MAXIMUM_RETRIES value determines the maximum size of this retry queue. It has no effect on the number of times one request will be retried. After DIRMAINT has been notified that this limit has been reached, DIRMAINT will not assign any more work to that particular DATAMOVE machine. The default value for DM_MAXIMUM_RETRIES= is 10. The format of the entry is

```
DM_MAXIMUM_RETRIES= integer
```

Step 4. Enabling DATAMOVE Exits

When adding a new minidisk, it can either be given to the user unformatted, or it can be given to DATAMOVE for formatting as a CMS minidisk before making it available to the user. When removing a minidisk from a user, any residual data may be left on that disk space, or the minidisk can be assigned to DATAMOVE for cleaning before making that space available for reuse. These functions are automatic if one or more DATAMOVE machines have been defined to DIRMAINT as shown in “Step 1. Define a DATAMOVE Service Machine to DIRMAINT” on page 75.

When changing a minidisk definition, DATAMOVE can usually format the new minidisk extent as a CMS minidisk and copy the existing CMS files from the old extent to the new minidisk extent. DATAMOVE can not copy files from OS or DOS formatted disks, or other non-CMS formatted space, nor can it correctly RECOMP the new minidisk and copy an IPLable nucleus from an existing reCOMPed minidisk, nor can it correctly copy sparce files from a RESERVEd minidisk. Some installations have customer written or vendor provided utilities that may be able to handle some of the situations that DATAMOVE can't handle. DATAMOVE can make use of these utilities by way of the DATAMOVE_NONCMS_COPYING_EXIT.

To enable use of this exit routine, specify the file name and file type; EXEC or MODULE. The format of this entry recorded in the CONFIG* DATADVH file is:

```
DATAMOVE_NONCMS_COPYING_EXIT=
```

DATAMOVE DATADVH

The DATAMOVE WAKEUP TIMES file controls time-driven events that take place in the virtual machines. A sample of this file (RECFM V) is supplied with the product code. As part of DATAMOVE initialization, it will be copied to the virtual machine's A-disk. The file name will always be called DATAMOVE, regardless of the user ID of the DATAMOVE service machine.

DATAMOVE WAKEUP Times File

```
1 ==/==/== 00:00:05 00/00/00 CMS EXEC DVHNDAY
2 ==/==/== 00:01:00 00/00/00 CMS EXEC DVHDAILY
3 ==/==/== +01:00:0 00/00/00 CMS EXEC DVHOURLY
4 ==/==/== 23:59:00 00/00/00 CP SLEEP 2 MIN
5 ==/==/== +00:mm:0 00/00/00 DMVCTL WAKEUP
```

These notes will help you with your DATAMOVE WAKEUP Times file.

1 The DVHNDAY EXEC is run after Midnight, every day. This is an IBM-supplied housekeeping routine. IBM recommends running this EXEC now. If you choose to retain your console spool files for only four or five days rather than the default, nine days, you can schedule a second invocation at or near Noon.

2 The DVHDAILY EXEC is run after Midnight each day, after the DVHNDAY EXEC has been run. This is an IBM-supplied housekeeping routine. IBM recommends that this routine be run at least once per day, or more often if you choose. You may adjust the time or times to suit your needs.

3 The DVHOURLY EXEC is run every hour, every day. This is an IBM-supplied housekeeping routine.

4 An event is **REQUIRED** to be scheduled before Midnight each day, with an action that will not be completed until after Midnight. This is necessary to ensure that events scheduled for the next day are recognized. The omission of this entry causes the service machine to hang up and never wake up as scheduled and may or may not respond to incoming user requests. The action performed is arbitrary; you may schedule one of the DVHDAILY, or DVHNDAY events at this time if you are sure the action will not complete until after Midnight.

Attention

Do not try to schedule two or more events at or near this specific time of day. If the first does not complete until after Midnight, the other event may not be processed at all.

5 The DMVCTL WAKEUP will cause the DATAMOVE server to review the DVHDMCTL QUEUE file for any pending work needed to be processed.

Where:

mm
Specifies the time interval in minutes, which best meets the performance and usability characteristics for your system. The sample shipped with the product code has the time set to 30 minutes, adjust this as required.

For more information, see the “The WAKEUP Times File” on page 233.

Chapter 5. Tailoring the DIRMSAT Service Machine

This chapter provides guidance for bringing up the DIRMaint SATellite service machine(s) to synchronize multiple object directories from a single source directory.

A multiple system CSE cluster contains multiple CPUs with shared DASD. This allows a single DIRMAINT service machine to maintain a single source directory that can be used by each of the systems in the cluster. DirMaint can only maintain a single object directory, and each system in the cluster needs its own object directory. A satellite service machine may be used to maintain a duplicate object directory as protection against a hardware error preventing use of the primary system residence DASD volume.

Defining the DIRMSAT Service Machines

Each DIRMSAT service machine must be defined to CP, to an ESM if one is installed, and to DIRMAINT.

Although it is possible to have the same user ID for service machines on different systems within a cluster, this imposes restrictions on how spool files and SMSGs can be sent. Even with the same user ID, each satellite service machine requires its own Read/Write disk space. Therefore, IBM recommends that each satellite service machine have a unique user ID within the cluster. Except for the name of the virtual machine, each one should be defined to CP as described under “Step 1. Define a Satellite Service Machine to DIRMAINT,” and if an ESM is installed each satellite must be defined to the ESM as described under Appendix A, “External Security Manager Considerations” on page 199.

Step 1. Define a Satellite Service Machine to DIRMAINT

To define a satellite service machine to DIRMAINT, add an entry to the CONFIG DATADVH file, or one of its auxiliaries. For more information on the CONFIG DATADVH file, see “CONFIG DATADVH” on page 44. The format is:

```
SATELLITE_SERVER= userid nodeid
```

Where:

userid
Identifies the satellite service machine you are logging on.

nodeid
Identifies the satellite machine you are running on.

Example—DirMaint Cluster Without CSE Shared Spooling:

```
SATELLITE_SERVER= DIRMSAT1 DVHTEST1 121150
SATELLITE_SERVER= DIRMSAT2 DVHTEST2 122150
SATELLITE_SERVER= DIRMSAT3 DVHTEST3 210150
SATELLITE_SERVER= DIRMSAT4 DVHTEST1 444444
SATELLITE_SERVER= DIRMSAT5 DVHTEST2 555555
SATELLITE_SERVER= DIRMSAT6 DVHTEST3 666666
```

Example—DirMaint Cluster With CSE Shared Spooling:

```
SATELLITE_SERVER= DIRMSAT1 DIRMSAT1 121150
SATELLITE_SERVER= DIRMSAT2 DIRMSAT2 122150
SATELLITE_SERVER= DIRMSAT3 DIRMSAT3 210150
SATELLITE_SERVER= DIRMSAT4 DIRMSAT4 444444
SATELLITE_SERVER= DIRMSAT5 DIRMSAT5 555555
SATELLITE_SERVER= DIRMSAT6 DIRMSAT6 666666
```

Note: Each satellite becomes its own *nodeid*.

Example—DirMaint Single Satellite Sever on a Stand-Alone System:

```
SATELLITE_SERVER= DIRMSAT  DIRMNODE *
```

Step 2. Identify the Communication Path

In addition to declaring the existence of each of the satellite servers, you must identify the communications path between DIRMAINT and each of the DIRMSAT machines, and the return path between the DIRMSAT machines and DIRMAINT.

Note: This is not required for a DIRMSAT machine running on the same node ID as the DIRMAINT machine for the purpose of maintaining a duplicate object directory. This is done with network routing entries in the CONFIG DATADVH file. The format is:

```
FROM= fromnode DEST= tonick  S= spoolid T= destnode
```

Where:

fromnode
: Identifies the network *nodeid* where the transaction originates.

tonick
: Identifies the nickname for the *nodeid* where the transaction is being sent.

spoolid
: Identifies the user ID of the machine where punch output should be sent to reach the specified destination. If cross system spooling is enabled, this is the user ID of the DirMaint service machine (either DIRMAINT or the DIRMSAT machine's user ID at that node ID); otherwise it is the *userid* of an RSCS network machine.

destnode
: Identifies the network *nodeid* where the transaction is processed.

Example—DirMaint Cluster Without CSE Shared Spooling:

```
FROM= DVHTEST1 TO= T2 S= DIRMNET1 T= DVHTEST2
FROM= DVHTEST2 TO= T1 S= DIRMNET2 T= DVHTEST1
FROM= DVHTEST1 TO= T3 S= DIRMNET1 T= DVHTEST3
FROM= DVHTEST3 TO= T1 S= DIRMNET3 T= DVHTEST1
FROM= DVHTEST2 TO= T3 S= DIRMNET2 T= DVHTEST3
FROM= DVHTEST3 TO= T2 S= DIRMNET3 T= DVHTEST2
```

Where:

DIRMNET1, *DIRMNET2*, **and** *DIRMNET3*
: Identifies the *userid* of the private RSCS network machines that carry only DirMaint traffic in our cluster. Use of a private network will be significantly faster than putting your DirMaint traffic on the general RSCS network.

Example—DirMaint Cluster With CSE Shared Spooling:

```
FROM= DVHTEST1 TO= DIRMSAT2 S= DIRMSAT2 T= DIRMSAT2
FROM= DVHTEST1 TO= DIRMSAT3 S= DIRMSAT3 T= DIRMSAT3
FROM= DVHTEST1 TO= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST1 TO= DIRMSAT5 S= DIRMSAT5 T= DIRMSAT5
FROM= DVHTEST1 TO= DIRMSAT6 S= DIRMSAT6 T= DIRMSAT6
FROM= DVHTEST2 TO= DIRMSAT1 S= DIRMSAT1 T= DIRMSAT1
FROM= DVHTEST2 TO= DIRMSAT3 S= DIRMSAT3 T= DIRMSAT3
FROM= DVHTEST2 TO= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST2 TO= DIRMSAT5 S= DIRMSAT5 T= DIRMSAT5
FROM= DVHTEST2 TO= DIRMSAT6 S= DIRMSAT6 T= DIRMSAT6
FROM= DVHTEST3 TO= DIRMSAT1 S= DIRMSAT1 T= DIRMSAT1
FROM= DVHTEST3 TO= DIRMSAT2 S= DIRMSAT2 T= DIRMSAT2
FROM= DVHTEST3 TO= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST3 TO= DIRMSAT5 S= DIRMSAT5 T= DIRMSAT5
FROM= DVHTEST3 TO= DIRMSAT6 S= DIRMSAT6 T= DIRMSAT6
FROM= DVHTEST1 TO= DVHTEST2 S= DIRMAINT T= DVHTEST2
FROM= DVHTEST1 TO= DVHTEST3 S= DIRMAINT T= DVHTEST3
FROM= DVHTEST2 TO= DVHTEST1 S= DIRMAINT T= DVHTEST1
FROM= DVHTEST2 TO= DVHTEST3 S= DIRMAINT T= DVHTEST3
FROM= DVHTEST3 TO= DVHTEST1 S= DIRMAINT T= DVHTEST1
FROM= DVHTEST3 TO= DVHTEST2 S= DIRMAINT T= DVHTEST2
```

Notes:

1. With CSE shared spooling available, a private network is not required.
2. This allows DIRMAINT to run on any of the three systems in the cluster without having to redefine the routings.
3. For a full 16 system cluster with 32 or more DIRMSAT machines, this would not be entirely practical.

DirMaint Emergency Coverage: You will want to enable DIRMAINT to run on two or three of the systems to give yourself emergency coverage in case the system where DIRMAINT usually runs is down, but you probably don't need all 15 systems in a full cluster as backup.

To use a single satellite server to maintain a duplicate object directory on a stand-alone system, no network routing information is required.

If you have not delegated authority to another user ID to use DIRM SEND and DIRM FILE commands, then you must make your changes to the CONFIG DATADVH file by logging on to the DIRMAINT machine.

If DIRMAINT is already up and running, enter:

```
CMS XEDIT CONFIG DATADVH
```

DIRMAINT will be nonresponsive to requests from other users although you are making these changes, but will queue them and begin processing them when you type:

```
FILE
```

If you are logged on as DIRMAINT although it is up and running, enter:

```
RLDDATA
```

without the *DIRM* prefix. If you are logged on to the DIRMAINT machine and want to leave it up and running, enter:

```
CP DISC
```

If DIRMAINT is not and running, enter:

```
XEDIT CONFIG DATADVH
```

Logoff after you have made and filed your changes.

DIRMSAT DATADVH

The DIRMSAT WAKEUP TIMES file controls time-driven events that take place in the virtual machines. A sample of this file is (RECFM V) is supplied with the product code. As part of DIRMSAT initialization, it will be copied to the virtual machine's A-disk. The file name will always be called DIRMSAT, regardless of the user ID of the satellite service machine.

DIRMSAT WAKEUP Times File

```
1 ==/==/== 00:00:05 00/00/00 CMS EXEC DVHNDAY
2 ==/==/== 00:01:00 00/00/00 CMS EXEC DVHDAILY
3 ==/==/== +01:00:0 00/00/00 CMS EXEC DVHOURLY
4 ==/==/== 23:59:00 00/00/00 CP SLEEP 2 MIN
```

These notes will help you with your DIRMSAT WAKEUP Times file.

1 The DVHNDAY EXEC is run after Midnight, every day. This is an IBM supplied housekeeping routine. IBM recommends running this EXEC now. If you choose to retain your console spool files for only four or five days rather than the default 9 days, you may which to schedule a second invocation at or near Noon.

2 The DVHDAILY EXEC is run after Midnight each day, after the DVHNDAY EXEC has been run. This is an IBM-supplied housekeeping routine. IBM recommends that this routine be run at least once per day, or more often if you choose. You may adjust the time or times to suit your needs.

3 The DVHOURLY EXEC is run every hour, every day. This is an IBM-supplied housekeeping routine.

4 An event is **REQUIRED** to be scheduled before Midnight each day, with an action that will not be completed until after Midnight. This is necessary to ensure that events scheduled for the next day are recognized. The omission of this entry causes the service machine to hang up and never wake up as scheduled and may or may not respond to incoming user requests. The action performed is arbitrary; you may schedule one of the DVHDAILY or DVHNDAY events at this time if you are sure the action will not complete until after Midnight.

Attention

Do not try to schedule two or more events at or near this specific time of day. If the first does not complete until after Midnight, the other event may not be processed at all.

For more information, see the “The WAKEUP Times File” on page 233.

Chapter 6. DASD Management

This chapter is intended to give a system administrator an understanding of how DirMaint can be used to perform DASD administration. It includes an overview of the methods used by DirMaint to perform these tasks and the required steps that need to be done to get your system running. In addition, some control structures used by DirMaint to handle DASD requests are discussed and the methods used for error recovery are explored. The relationship between the DIRMAINT machine and the product servers will also be explained. For more information on the product server types, see "What is a Server?" on page 6.

Preparing Your DIRMAINT Machine

Before you start, you should be aware of the several tasks you must accomplish to define your DIRMAINT machine. The following sections describe the tasks you must perform to prepare for the DASD requests.

Defining a DATAMOVE Machine to the DIRMAINT Server

A single configuration file entry will define your DATAMOVE server to the DIRMAINT machine. The format is:

```
DATAMOVE_MACHINE= machname machnode sysaffin
```

Where:

machname
Identifies the user ID of the DATAMOVE machine.

machnode
Identifies the RSCS node name of the DATAMOVE machine.

sysaffin
Identifies the system affinity associated with the DATAMOVE machine. For non-CSE systems this value is usually an *.

Usage Notes

1. One entry for each DATAMOVE machine in your system is required.
2. All fields on the entry are required.
3. Duplicate entries or entries with an incorrect format will be rejected with the appropriate error messages during initialization.
4. These entries are consulted when the DirMaint server is initializing, when the DVHBEGIN command is entered or the DirMaint server is AUTOLOGed. The appropriate control structures are built for each DATAMOVE machine and they are considered ready for work.
5. For more information on DirMaint configuration files and specifying configuration file entries, see “CONFIG DATADVH” on page 44.

Example—Segment of the Configuration File: This entry defines four DATAMOVE servers: DATAMOVA, DATAMOVB, DATAMOVC and DATAMOVD on GDLVM7 with a system affinity of *. Enter:

```
DATAMOVE_MACHINE= DATAMOVA GDLVM7 *
DATAMOVE_MACHINE= DATAMOVB GDLVM7 *
DATAMOVE_MACHINE= DATAMOVC GDLVM7 *
DATAMOVE_MACHINE= DATAMOVD GDLVM7 *
```

The Extent Control File

The EXTENT CONTROL file defines any volume that is being used for minidisk allocation and provides a template, or layout, of how the space should be used. In addition, it also contains system and device default values used during allocation operations.

Note: Explicitly defined volumes done through the AUTOV operand of the AMDISK command need not be defined in the extent control file.

An example of an EXTENT CONTROL file is shown in Figure 18 on page 89. This example has only an abbreviated list of autoblock and default entries, because of the size of these extent control file sections.

The extent control file contains several sections; each section may occur only once. Each section starts and ends with an identifying tag. To enhance readability in the extent control file, you can add blank lines and use the asterisk (*) to annotate your console sheet or display screen.

The extent control file must exist on the DIRMAINT 1DF disk prior to use. During installation the IN2PROD EXEC places the extent control file on the 1DF disk.

```
* ------------------------------------------------------ *
* Any header comments are placed here.                   *
* All records starting with an asterisk are ignored.     *
* ------------------------------------------------------ *
1 :REGIONS.
* ------------------------------------------------------ *
* Regions are mapped in this section.                    *
* RegionId     VolSer     RegStart   RegEnd       Type   *
* ------------------------------------------------------ *
* Note:  The 'Type' field device in the :DEFAULTS. *
* section must be set to the correct max cylinder value. *
* ------------------------------------------------------ *
RegionA        Myvol1     1          200          3380-02
RegionB        Myvol1     201        400          3380-02
RegionAll      Myvol1     START      END          3380-02
:END.
2 :GROUPS.
* ------------------------------------------------------ *
* Groups are mapped in this section.                     *
* GroupName RegionList                                   *
* ------------------------------------------------------ *
MyGroup1 RegionA RegionB RegionAll
:END.
3 :AUTOBLOCK.
* ------------------------------------------------------ *
* All autoblock allocation parameters are placed into    *
* this section.                                          *
* DASDType BlockSize Blocks/Unit Alloc_Unit Architecture *
* ------------------------------------------------------ *

⋮
3375       4096         96          1     CKD
3380        800        540          1     CKD
3380        512        690          1     CKD

⋮
:END.
4 :EXCLUDE.
* ------------------------------------------------------ *
* All excluded users and user devices are                *
* placed into this section                               *
* ------------------------------------------------------ *
MAINT 0190
MAINT 02*
:END.
```

Figure 18 (Part 1 of 2). EXTENT CONTROL File

```
5 :DEFAULTS.
  * -------------------------------------------------------
  * All default capacities are placed into this section.
  * The device type must be the same as selected in
  * the :REGIONS. section.
  * -------------------------------------------------------

⋮
  3375          959
  3380-01       885
  3380-02      1770
  3380-03      2655
  3380          885
  3390-01      1113

⋮
6 :END.
```

Figure 18 (Part 2 of 2). EXTENT CONTROL File

Extent Control File Sections

A brief explanation of the extent control file sections is shown in Table 9. The section name is also used as the identifying tag in the extent control file.

Table 9. Summary of Extent Control File Sections

Section	Function
1 :REGIONS.	Defines an area or region on your DASD volume for use during DirMaint automatic allocation.
2 :GROUPS.	Defines a grouping of regions for use during DirMaint automatic allocation.
3 :AUTOBLOCK.	Defines blocking factors and device architectures for various device types.
4 :EXCLUDE.	Defines users or user/device combinations that should be considered as excluded by the DirMaint DASD subsystem.
5 :DEFAULTS.	Defines the default maximum size for various DASD devices.
6 :END.	Defines the ending tag for all sections.

Notes:

1. Only a single occurrence of any section may occur in the EXTENT CONTROL file. If multiple occurrences of a single section do occur, the first is used.
2. Sections may be presented in any order within the EXTENT CONTROL file.
3. All tags may be specified in mixed case. DirMaint translates the EXTENT CONTROL file to upper case as it is read.

:REGIONS. Section

A region (1) is the basic unit of DASD segmentation used by DirMaint. It defines a single, contiguous area (a region) on a single DASD volume. Multiple region definitions are allowed within the :REGIONS. section of the extent control file. The region entry is similar to the MDISKS entry that existed in prior releases of DirMaint. The format is:

```
regionid volser start end ttttmmmm
```

Where:

regionid
Specifies the name of this region entry. Specification of this field is subject to the following rules.

- Region names must be unique. If a region entry shares the same name with another region entry, the first record is used, the second entry is ignored.
- Region names may consist of the characters A-Z, 0-9, #, @ and $. DirMaint requires that this field be eight characters or less.

volser
Specifies the volume ID of the region where the region is located.

- This value represents the value placed into volser field on any minidisks generated from this region.
- Volume IDs supported by DirMaint consist of the characters A-Z, 0-9, #, @ and $.

start
Specifies the starting block or cylinder (inclusive) of the region. A keyword of START can be used to define the start of a region. This translates to cylinder 1 of a CKD device and block 32 of an FBA device.

end
Specifies the ending block or cylinder (inclusive) of the region. A keyword of END can be used to define the end of a region. This translates to the largest cylinder or block available on the volume. This value will differ with device type and model.

Attention

Specification of an ending value that exceeds the physical volume maximum cylinder or block is allowed in a region definition, but allocation requests will not be granted beyond the physical limitations of the volume.

ttttmmmm
Specifies the DASD device type and an associated model number information.

The *ttttmmmm* value specified on the :REGIONS: entry must exactly match the value specified in the :DEFAULTS. section. If unable to locate an exact match, the device type (*tttt*) determines the maximum allocatable block or cylinder.

tttt Specifies the device type associated with this region. This value is placed into the directory source on the MDISK statement when allocations take place on a volume defined by this region.

mmmm

Specifies a string used when cross checking a :REGIONS. entry with a :DEFAULTS. entry. Generally this consists of model number information:

- The model number is optional.
- The *mmmm* value, if specified, is not limited to a specific size. It may be a small as a single character or as many characters as will fit on a single line.
- Supported characters are A-Z, 0-9, #, @, $, or - (dash).
- Imbedded blanks are not allowed.
- Determines the maximum allocatable block or cylinder from the :DEFAULTS. section of the EXTENT CONTROL file.

Usage Notes

1. Region entries can be specified in mixed case but case is not respected. DirMaint translates all entries to upper case as they are read.
2. Any data following the last required field on each region entry is ignored.
3. DirMaint does not imposed a limit on the number of region entries.
4. Regions may overlap; however, that allocation requests that target a specific region must occur entirely within that region to be considered successful.
5. Areas of DASD volumes that are not defined through region entries are not eligible for the various automatic allocation methods supported by DirMaint. Except, AUTOV which can allocate the region. Extents in these areas must be allocated using specific extent information.
6. DirMaint does not allow the use of &SYSRES for a volume identification on an MDISK directory statement. The value of +VMRES is supported, with some restrictions.
   - The use of +VMRES is reserved by CP and can not be used as the real volume label of a physical DASD volume. (If +VMRES is a real volume label, the pseudo label can be changed by including the &SYSRES parameter on the DIRECTXA_OPTIONS= entry in the CONFIG* DATADVH file(s).
   - When using either AUTOV or VBLK*nnnn* allocation for an MDISK on an IPLable system residence volume, the administrator must ensure that all MDISK statements for that volume are defined the same way, either using +VMRES (or alternate synonym) for all or using the real volume label for all; mixing the two forms is not supported.
   - When allocating space by specifying an absolute starting cylinder or block, the system administrator must ensure that the volume identification used is consistent with the adjacent space, either using +VMRES (or alternate synonym) or the real volume label consistently; mixing the two forms is not supported.

:GROUPS. Section

A group (**2**) is a collection of one or more regions. The format is:

```
groupid region1 region2 ...regionn
```

Where:

groupid
Specifies the name of this group entry. Specification of this field is subject to the following rules.

- If a group entry shares the same name with another group entry, it is considered a single group. For instance, consider the following groups segment from an EXTENT CONTROL file:

  ```
  MyGroup   Mike1 Mike2
  MyGroup2  Mark1
  MyGroup   Mike3
  ```

  This series of statements actually defines two groups, MyGroup and MyGroup2. The same information could have been represented as:

  ```
  MyGroup   Mike1 Mike2 Mike3
  MyGroup2  Mark1
  ```

  This gives the user the ability to define large groups without using excessively long records.
- Group names must not start with a valid EXTENT CONTROL file tag.
- Group names may consist of the characters A-Z, 0-9, #, @, and $.
- DirMaint requires that this field be eight characters or less.

regionn
Specifies a region that exists within this group. The region must be defined in the :REGIONS. section.

Usage Notes

1. Regions within a group are searched, in order, for a valid location for DASD allocation.
2. Group entries can be specified in mixed case but case is not respected. DirMaint translates all entries to upper case as they are read.
3. The default scanning method when allocating DASD from a group is to scan from the first region defined within the group to the last region defined within the group each time an allocation request is made. An alternate scanning method can be used. This method is referred to as wrapping or rotating. To employ this method, an additional group definition line is required for the group using the alternate scanning method. The format of the entry is:

   ```
   GRPNAME (ALLOCATE ROTATING)
   ```

   This entry is in addition to the group statements.

Example—Defining a Group Name: The following group defines a group name called GDLVM7 that contains four regions (REG1, REG2, REG3 and REG4)

```
GDLVM7 REG1 REG2
GDLVM7 REG3 REG4
```

By default each allocation attempt will attempt to allocate in REG1 before attempting to allocate in REG2, REG3 and finally REG4 in that order. By altering the group definition to:

```
GDLVM7 (ALLOCATE ROTATING)
GDLVM7 REG1 REG2
GDLVM7 REG3 REG4
```

DirMaint will track the name of the region that was last allocated on. The next allocation attempt will take place on the following region name.

Example—A Successful Allocation: If a successful allocation took place on REG2, the next allocation attempt will be attempted on REG3, REG4, REG1 then REG2, in that order.

:AUTOBLOCK. Section

AutoBlock (**3**) entries are used by the DirMaint machine to calculate the number of cylinders or blocks to allocate with some automatic allocation methods. The specific architecture type for supported DASD types is also obtained from this section. The format is:

```
type blksize blkperunit allocunit architecture
```

Where:

type
Specifies the DASD type associated with each entry.

blksize
Specifies the block size of this entry.

blkperunit
Specifies the blocks per unit for this entry.

allocunit
Specifies the allocation unit for this entry.

architecture
Specifies the device architecture associated with the entry.

Usage Notes

1. This section is shipped fully defined. Modifications should only be made to this section if your installation is using an unlisted device type.
2. When initializing the volume control files, the device architecture is taken from the AUTOBLOCK section.
3. The following automatic allocation methods use this table to determine the actual number of cylinders or blocks to allocate.
   - GBLK*nnnn*
   - RBLK*nnnn*
   - TBLK*nnnn*

- VBDS*nnnn*
- VBLK*nnnn*

Where:

nnnn
Specifies the combination of the device type of volid being targeted for allocation and the blocking factor. The automatic allocation keyword determines which entry to use.

This section may be altered if some device types do not apply to your installation. For instance, some device types do not apply in a VM/ESA 1.5 370 environment. You may choose to delete or comment out the appropriate device types from this section. Should a user attempt to use the device type, DirMaint will then reject the attempt without having to resort to calling DIRECT.

Allocation Formula

```
actualalloc = rndup(allocsize / blkperunit) * allocunit
```

Where:

actualalloc
Specifies the number of cylinders or blocks actually obtained during the allocation request.

allocsize
Specifies the value passed during the allocation request.

BlkPerUnit
Is obtained from the AutoBlk entry.

AllocUnit
Is obtained from the AutoBlk entry.

:EXCLUDE. Section

DirMaint represents extents within its control structures as either extents or excluded (**4**) extents. Excluded extents are excluded from extent checking during DirMaint DASD allocation operations. This includes the full volume overlays for backup operations.

The EXCLUDE section of the EXTENT CONTROL file gives the system administrator the ability to map some or all of a user's extents as excluded. Multiple entries are also allowed. The format is:

```
userid Address
```

Where:

userid
Identifies owner of the extent or extents to be excluded. A user ID can be followed by an * to act as a wild card character.

address
Specifies the address or set of addresses to be excluded.

- The address is an optional field. If it is not provided, or an asterisk is

specified, all minidisk specifications owned by the user ID are considered excluded.

- If the address is provided it may consist of 1 to 4 digits, with an optional trailing asterisk.
  - DirMaint considers all addresses to consist of four digits.
  - The specified digits are considered the left most digits of the four digit address.
  - A trailing asterisk or specification of less than four digits implies that all addresses starting with the specified digits are to be considered excluded.

Example—Wild Card Character: If you, Enter:

```
HOWLAND* 0191
```

This excludes the 0191 device on any *user ID* starting with HOWLAND, is shown as:

```
HOWLAND1, HOWLAND2, ...
```

If you specify an * without an *user ID*, the entry is ignored and is treated as a comment statement.

Example—Segment of the EXCLUDE section:

```
CAMUT               * Every disk owned by CAMUT is excluded.
HOWLANDM 3*         * HOWLANDM's 3000 - 3FFF are excluded
HOWLANDM 01*        * HOWLANDM's 0100 - 01FF are excluded
HOWLANDM 1199       * HOWLANDM's 1199 is excluded
```

Any extent owned by HOWLANDM with an address that matches the addresses listed will be considered excluded. All of the following MDISK statements would be considered excluded if they occurred in the HOWLANDM directory entry:

```
MDISK 0199 3380 1 END SYSPAK      (matched '01*')
MDISK 177 3380 1 END SYSPAK       (matched '01*')
MDISK 3199 3380 1 END SYSPAK      (matched '3*')
MDISK 1199 3380 1 END SYSPAK      (matched '1199')
```

Attention

Defining a user or a user's device as excluded forces the extent to be considered as an EXCLUDED extent when DirMaint builds its volume control files. Excluded extents are not consulted before allocating new extents. This allows a new extent to overlap an EXCLUDED extent.

:DEFAULTS. Section

As discussed in ":REGIONS. Section" on page 91, the *ttttmmmm* field of your region entries determines the maximum allocatable block or cylinder when building the volume control structures within the DIRMAINT machine. When this is done, the :DEFAULTS. (**5**) section of the EXTENT CONTROL file is consulted to find the required value. If an extent is encountered during initialization that does not have a region entry that can be used to determine the maximum allocatable block or cylinder, the device type from the MDISK statement determines the maximum block or cylinder. The format is:

```
tttmmmm Cyl|Blk
```

Where:

ttttmmmm
: Specifies the DASD device type and an associated model number information associated with this default entry.

  The *ttttmmmm* value specified on the :DEFAULTS. entry must exactly match the value specified in the :REGIONS. section. If unable to locate an exact match, the device type (*tttt*) determines the maximum allocatable block or cylinder.

  tttt Specifies the device type associated with this default entry.

  mmmm
  : Specifies a string used when cross checking a :REGIONS. entry with a :DEFAULTS. entry. Generally this consists of model number information.

    - The model number information is optional.
    - The *mmmm* value, if specified, is not limited to a specific size. It may be a small as a single character.
    - Supported characters are A-Z, 0-9, #, @, $, or -(dash).
    - Imbedded blanks are not supported.
    - Is used during initialization to determine the maximum allocatable block or cylinder for a :REGIONS. entry from the :DEFAULTS. section of the EXTENT CONTROL file.

cyllblk
: Specifies the maximum allocatable cylinder (for CKD devices) or block (for FBA devices) on this device.

Example—Fragment from the EXTENT CONTROL File:

```
3380          885
3380-01       885
3380-02      1770
3380-03      2655
```

Notice that if a region entry specifies a *ttttmmmm* entry of:

```
3380-03
```

DirMaint will consider cylinder 2654 as the maximum allocatable cylinder. If there are no region entries on this system and an extent using device type 3380 is found, DirMaint will default to the

```
3380          885
```

entry and consider 884 as the maximum cylinder. The last example shown, if all devices in your system are Model 03, the default value for 3380 may be altered from 885 to 2655. After doing this, all devices without a corresponding region entry will use a default value of 2655.

Attention

If DirMaint cannot determine the maximum block/cylinder from an explicit region entry, the default value for the device will be used. This is usually the smallest model. This renders extents above the limit as unallocatable. To correct this, define an explicit region or change the default value for the device and rebuild the volume control files using the RLDEXTN command.

:END. Tag

The :END. tag is used to denote the end of the :REGIONS., :GROUPS., :AUTOBLOCK., :EXCLUDE., and :DEFAULTS. sections. A single :END. tag should follow each section.

The AUTHDASD File

DirMaint allows the local system to implement a DASD allocation authorization system through an exit call. For more information, see “DASD Authorization Checking (DVHXDA)” on page 166. If the exit is not located or if the exit defers, DirMaint defaults to its native DASD allocation authorization scheme.

DASD allocation requests are provided with two levels of control under DirMaint:

- The first level is the command authority required to issue the command. For more information on command classes, see “Command Classes” on page 135.
- The second level is the protection in the entries of the AUTHDASD DATADVH control file. This file is located on the primary directory disk.

The AUTHDASD DATADVH Control File

The format is:

```
userid node allocclass name1 name2 ... nameN
```

Where:

userid
Identifies the user ID issuing the allocation request.

- A value of * is valid in this field. When this value is used it indicates that all user's on the specified node are authorized for the given allocation type.

node
Identifies the node of the user issuing the allocation request.

- A value of * is valid in this field. When this value is used it indicates that the specified user on any node is authorized for the given allocation type.

allocclass
Specifies one of the following allocation classes authorized for the specified user ID.

Table 10. Allocation Classes

Class	Allocation Types	Explanation
VOLUME	AUTOV or VLBK*xxxx*	Allocations requests that involve volume level authority. The values following this field are the authorized vol IDs or an * indicating that the user is authorized for volume level authorization on all volumes within the system.
REGION	AUTOR or RBLK*xxxx*	Allocations requests that involve region level authority. The values following this field are the authorized regions or an * indicating that the user is authorized to allocate within any defined region. **Note:** DASD areas not contained within a defined region can not be allocated using this authority and explicit extents can not be provided. The user is restricted to AUTOR and RBLK*xxxx*.
GROUP	AUTOG or GBLK*xxxx*	Allocations requests that involve group level authority. The values following this field are the authorized groups or an * indicating that the user is authorized to allocate within any defined group. **Note:** DASD areas not contained within a defined group can not be allocated using this authority and explicit extents can not be provided. The user is restricted to AUTOG and GBLK*xxxx*.
SPECIFIC	Numeric Value	Allocation requests that involve specific extent information on the command invocation. Allocation requests requiring this level of authority involve those requests that supply the starting cylinder, cylinder or block count and the volume ID of the intended allocation.
*	Any	Unlimited allocation requests. Any valid allocation method is allowed, including DEVNO requests. **Note:** This authority is required to process allocation requests involving DEVNO. Authorization at this level authorizes the user to use any available method to allocate.

namen

Specifies the REGION, GROUP or VOLUME ID to which this authorization applies.

- A value of * is valid in this field. When this value is used, all instances of the allocation type are considered authorized.

Usage Notes

1. Some allocation requests require no authority and are always authorized. V-DISK, T-DISK, TBLK*xxxx* and VDBS*xxxx* are examples of allocation methods that are always authorized.
2. Your installation may rely exclusively on the command privilege classes of the commands that allocate DASD to protect your disk resources. In this case, you may consider granting all users global authority. If you choose to permit all users, who have the command authority to enter DASD allocation commands, using any method, then place this entry in the AUTHDASD file:

   ```
   *      *      *
   ```

   This is the IBM shipped default setting for the AuthDASD file. If this record is being used, ensure that only authorized users have been given the command authority to enter commands that allocate DASD.

Automatic Allocation Algorithms

The DirMaint product supports two automatic allocation algorithms. The algorithm is selected by placing an entry in an accessed configuration file. The format is:

```
DASD_ALLOCATE= method
```

Where:

method
Indicates one of the following allocation methods.

FIRST_FIT
Specifies that allocation attempts within a defined DASD region be conducted on a first fit basis. The first gap found of sufficient size within the specified allocation area is allocated.

EXACT_FF
Specifies that allocation attempts will utilize an exact fit algorithm followed by first fit. Allocation will be attempted in any gap that exactly matches the size being allocated. Should there be no gap that matches this size, then a first fit algorithm is employed.

Usage Notes

1. If allowed to default, or if an unknown value is entered, the FIRST_FIT algorithm is used.
2. The FIRST_FIT algorithm should yield better performance as it only searches a region for a gap that is large enough to contain the new extent. The EXACT_FF algorithm, although slower, should minimize DASD fragmentation over time.
3. DirMaint will never allocate an extent that forms an overlap with another nonexcluded extent unless specific extent information is provided and extent checking is OFF. Extent checking can be set by using a configuration file entry. The format is:

   ```
   EXTENT_CHECK= ON|OFF
   ```

   Where:

   ON
   Is the default setting.

OFF
If a value other than ON or OFF is entered, the default value of ON is used.

4. The number of unassigned work units is controlled by the entry in the CONFIG* DATADVH file.

   The format is:

   ```
   MAXIMUM_UNASSIGNED_WORKUNITS= nnnn
   ```

   Where:

   nnnn
   Is an unsigned integer and defaults to 0 if not specified. If left to default, all DASD transactions will be rejected.

   For more information on the Work Unit Control File (WUCF), see "Work Unit Control File" on page 104.

5. Unassigned work units may build up on your system for several reasons. The most common may be a busy DATAMOVE machine(s). When a DASD request requiring DATAMOVE interaction is received, a work unit is created and placed on the unassigned queue if all DATAMOVE machines are currently active. The work units are removed from the unassigned queue and assigned to a DATAMOVE machine as each DataMove becomes available.
6. Setting this value too low may result in DASD commands being rejected if a large influx of DASD commands are received. A value of 25 is recommended for general use. You may choose to set this value higher if DirMaint is being used in an environment where a large volume of DASD allocation commands in a short period of time is expected (a university installation for instance).

Protecting System Areas on DASD

DirMaint has the ability to use the CP QUERY ALLOC command to display the number of cylinders or pages that are allocated, in use, and available for DASD volumes attached to the system.

DirMaint will use the following operands of the CP QUERY ALLOC command:

CP QUERY ALLOC Area	DirMaint Extent Owner
DRCT	.DIRECT.
PAGE	.PAGE.
SPOOL	.SPOOL.
TDISK	.TDISK.

Notes:

1. The CP QUERY ALLOC command is not valid on all releases and requires that DIRMAINT machine be authorized for privilege class D where the command is valid.
2. The mapping of system areas is only done on volumes known to DirMaint at the time the facility is invoked. If the command is not valid or the DIRMAINT machine is not authorized the mappings will not take place. The mapping is invoked during initialization and when the ALL option of the RLDEXTN command is used. For more information about the RLDEXTN command, see *Directory Maintenance VM/ESA: Command Reference*.
3. Known volumes include any volume specified in the :REGIONS. section of the EXTENT CONTROL file and any volume referenced by an MDISK statement within the source directory, at the time the volume control files are built.
4. If an allocation attempt is made on a volume unknown to DirMaint it is important to note that any system areas resident on that volser have not been explicitly protected by DirMaint. It is recommended that any volume with system areas be specified in the :REGIONS. section of the EXTENT CONTROL file.
5. For more information about the CP QUERY ALLOC command, see *z/VM: CP Command and Utility Reference*.
6. In a CSE cluster, DirMaint only maps and protects the CP owned space on the system where the DIRMAINT machine is running.

Volume Control File

The volume control files are used during an allocation attempt to locate a free area for allocation. These files are also consulted when building free and used maps of DASD space.

A volume control file is built for each known volume on the system. This includes volumes that are not used in the directory but mentioned in the :REGIONS. area of the EXTENT CONTROL file. Volume control files are built with the volser as the file name and VCONTROL as the file type. These files reside on the primary directory file mode and are built and maintained automatically by DirMaint.

You can place statements in the volume control file however, if you specify more than one statement with the same operands the last operand definition overrides any previous specifications. For more information on the automatic mapping of system areas see “Protecting System Areas on DASD” on page 101. This only takes place on volumes known to DirMaint at the time, the facility is invoked.

Volume Control File Example

An example of a volume control file fragment:

```
⋮
1 DEVTYPE= 3390
2 MAXBLK= 1113
3 ARCH= CKD
4 ENTRY= 11 20 HOWLANDM 0192 *
4 ENTRY= 21 40 HOWLANDM 0193 *
4 ENTRY= 41 50 RITTERME 0191 *
5 EXCLD=  6 100  7 105  8 RITTERME  9 0111  10 *
⋮
```

Figure 19. Volume Control File

The following notes are to help you with your "Volume Control File Example":

1 DEVTYPE specifies the device type associated with this volume id. This value is the *ttttmmmm* field as mentioned in ":REGIONS. Section" on page 91. If this volume control file was not generated by a region entry, this field is the device type from the MDISK statement. Note that any future extents allocated on this volume with dynamic device type allocation use the first four digits from this field.

2 MAXBLK specifies the maximum allocatable block or cylinder.

3 ARCH specifies the architecture associated with this volume. Currently this value is FBA or CKD.

4 ENTRY specifies an extent entry.

5 EXCLD specifies an excluded extent entry.

Example—ENTRY and EXCLD Multiple Field Records: ENTRY and EXCLD are multiple field records representing a single contiguous extent. The format is:

```
type start end owner address sysaffin
```

Type **5**
Specifies the type of extent entry. EXCLD and ENTRY represent an excluded entry and a typical entry respectively.

Start **6**
Specifies the starting extent block or cylinder, inclusive.

End **7**
Specifies the ending extent block or cylinder, inclusive.

Owner **8**
Specifies the user ID that owns the extent.

Address **9**
Specifies the address the owner references this extent with.

sysaffin **10**
Specifies the system affinity associated with this extent.

Operation

The operation of the DASD subsystem involves two steps: directory initialization and manipulating extents.

Directory Initialization

When DirMaint is initialized, the DirMaint machine builds several control structures to represent the current state of allocation for each known volume. The volume control files can also be rebuilt by using the RLDEXTN command with the ALL option. For more information on the RLDEXTN command, see *Directory Maintenance VM/ESA: Command Reference*. This operation is done automatically by DirMaint when it initializes for the first time.

Manipulating Extents

DirMaint provides several commands designed to allocate, change, delete and manipulate directory MDISK statements.

AMDISK	Add (allocate) a new extent.
CMDISK	Alter the size of an existing extent.
DMDISK	Deallocate an existing extent.
MMDISK	Mirror the extent information from one extent to another.
RMDISK	Redefine the extent information on an existing extent.
MDISK	Alter the mode and passwords of an existing extent.

For more information on commands, see *Directory Maintenance VM/ESA: Command Reference*.

Work Unit Control File

When DirMaint receives a DASD request of AMDISK, CMDISK, DMDISK, or TMDISK, a transaction file, known as a WUCF, is built and placed on disk. The WUCF is checked to see if asynchronous processing is required. If the transaction does not require DATAMOVE activity, it is immediately acted on. WUCFs requiring asynchronous processing are assigned to an open DATAMOVE machine or placed on a queue to await the next available DATAMOVE machine.

The WUCF is a transaction file representing a DASD command, created on the primary directory disk. The format is:

```
nnnnnnnn WORKUNIT
```

Where:

nnnnnnnn
Specifies a random 8-digit number representing the specific work unit.

Transaction File Example

```
 1 DMM: &DMM.NAME &DMM.NODE
 2 DEV.ONE: &DEV.ONE
 3 DEV.TWO: &DEV.TWO
 4 ORIGNODE: GDLVM7
 5 ORIGUSER: MNTDASD1
 6 ORIGSEQ#: 12
 7 ORIGCMD: AMDISK 0306 3390 AUTOG 500 GDLVM7 MR
 8 SYSAFFIN: *
 9 TARGETID: DSSERV
10 LANG: AMENG
11 CMDLEVEL: 140A
12 ASUSER: MNTDASD1
13 REQUEST: 12
14 ORIGEXTENT: N/A
15 BEGINCMDS:
16 NTRIED  17 WORKUNIT  18 ENABLE
16 NTRIED  17 AMDISK  18 FOR DSSERV 0306 33        90 AUTOG 500 GDLVM7 MR
16 NTRIED  17 UNLOCK  18 0306 DSSERV NOMSG
16 NTRIED  17 WORKUNIT  18 RESET
16 NTRIED  17 DIRECT
```

Figure 20. Transaction File

The “Transaction File Example” contains two separate areas.

Prefix Area

The prefix area establishes the context the command was entered in. This also lists the DATAMOVE machine that owns the WUCF and any devices currently in use by this WUCF. The following notes are to help you with your “Transaction File Example.”

1 This section contains the user ID and node ID of the DATAMOVE machine that was assigned to act on behalf of this command. In this example, the &DMM.NAME and &DMM.NODE indicate that this work unit has not been assigned to a specific DATAMOVE. This situation is common and may indicate that all DATAMOVE machines were busy when the work unit was received. If DATAMOVE interaction is not required, this field will remain &DMM.NAME and &DMM.NODE.

2 This field contains the first device being used on the DATAMOVE machine that is associated with this work unit. In this example, the &DEV.ONE is a further indication that the work unit has not been assigned. After the work unit is assigned to a DATAMOVE this field is changed to reflect a virtual address on the DATAMOVE machine. If DATAMOVE interaction is not required, this field will remain &DEV.ONE.

3 This field contains the second device being used on the DATAMOVE machine that is associated with this work unit. In this example, the &DEV.TWO is a further indication that the work unit has not been assigned. After the work unit is assigned to a DATAMOVE this field is changed to reflect a virtual address on the DATAMOVE machine. If DATAMOVE interaction is not required, this field will remain &DEV.TWO. Some requests only require a single device.

4 Reflects the node from which the command originated. In this example, the command originated from node GDLVM7.

5 Reflects the user that originated the command. In this example, the user MNTDASD1 entered the command.

6 Is the sequence number given in the original command.

7 Is the original command. All parameters are presented as provided from the user.

8 Is the system affinity associated with the command invocation.

9 Is the target user ID, which will be altered by the command.

10 Is the language associated with this command.

11 Is the command level in which the command was entered.

12 Is the setting of the prefix ASUSER when the command was entered.

13 Is the request number associated with this command.

14 Is the original extent information or N/A. This field is only set if the command being processed is a CMDISK.

15 Is a token representing the start of the commands area and the end of the prefix area.

Command Area

The command area defines the subcommands required to accomplish the task. There are usually several subcommands within each WUCF. The exact sequence of commands depends on the command that generated the WUCF. Regardless of the command, each subcommand consists of three basic parts:

16 Part 1

Status
Identifies the status of this specific command. The first part may have the following values:

NTRIED
The command has not been tried.

DONE*nnnncccc*
The command has returned from DATAMOVE or returned from a subcommand handler with the specified status. The field is zero padded.

Where:

nnnn
Specifies the DATAMOVE return code.

For more information for specific meanings, see *Directory Maintenance VM/ESA: Messages*. If this status is reflecting the return code from a subcommand handler, this field will always be zeros.

cccc
Specifies the CMS return code of a failing condition (if we have a failing condition) or zeros if all worked well.

ACTIVE
The command has been sent to DATAMOVE to process. No status has returned from the DATAMOVE machine.

RETRY
The command has been sent to DATAMOVE to process. A recoverable error was returned and this step should be retried.

17 Part 2

Command Name
Denotes the subcommand name.

18 Part 3

Command Parameters
Denotes the specific parameters associated with the command. This list will be different for each command.

Error Recovery

When a failure occurs although a work unit is being processed, automatic rollback processing is attempted before deallocating the work unit. If rollback is possible, a batch file is created with the required commands to rollback the work unit. The batch file is submitted automatically. In either case, the Work Unit Control File is copied to a file for administrative review. The format is:

```
nnnnnnnn WUCFFAIL
```

To understand the specific rollback processing done by DirMaint requires some background information on the method used by DirMaint to handle DASD requests.

As many events that require the DATAMOVE server are handled asynchronously, DirMaint DASD operations are subject to asynchronous failures. The DASD subsystem has been designed to meet three types of asynchronous failures:

Soft Failures

Soft failures generally occur when the assigned DATAMOVE machine is unable to obtain a link to the required device. This can occur when the object directory has not been placed online or when a virtual machine still has a link to the required device. These errors are handled by DirMaint as retry events. The DATAMOVE virtual machine will periodically attempt the operation and, if links are obtained, complete the required operation. This failure will leave no residual files indicating that it ever occurred.

Hard Failure—Recoverable

If possible, the DirMaint machine will rollback the transaction and return the system to a state that existed before the command was entered. In either case, a residual file is produced to help the administrator determine the cause of the failure.

Automated Rollback

Automatic rollback processing involves:

- Transferring resources from the DATAMOVE machine back to the user
- Releasing any obtained, but unused, extents.
- Removing locks on devices

There are three scenarios where this can be performed by DirMaint:

- Failure of AMDISK subcommand. Several commands create a WUCF where one of the first operations involves the subcommand AMDISK. If this command should fail, DirMaint is able to release any device locks obtained by this transaction.
- Failure of DATAMOVE COPY request. The WUCF created for a CMDISK request involves using an AMDISK subcommand to create a new extent, transferring the original extent to a DATAMOVE machine, and requesting that the DataMove machine COPY the information from the old extent to the new extent. The new extent is then returned to the user and the old extent is released. Should the COPY request fail, DirMaint is able to:
  - Transfer the old extent back to the user
  - Release the new extent
  - Release any device locks obtained by this transaction
- Failure of DATAMOVE format. The WUCF created for an AMDISK that requires formatting involves the allocation of a new extent on a DATAMOVE machine, formatting it and finally transferring it to the user. If the format should fail, DIRMAINT is able to:
  - Release the obtained extent
  - Release any locks obtained by this transaction

Hard Failure—NonRecoverable

If a WUCF failure does not meet the criteria for automatic rollback, the WUCF may require administrative intervention to clean up after the failure. The original WUCF, now renamed to *nnnnnnnn* WUCFFAIL, remains as a history of what commands were performed and which commands failed. Note that device locks and extents all remain exactly as they were when the WUCF failed.

Manual Rollback

The manual steps required to rollback a WUCF will vary with the command that created the WUCF. This can be determined from the ORIGCMD: tag in the prefix area. The specific actions will depend on where the WUCF failed. As explained in “Work Unit Control File” on page 104, the command status field will indicate the failing command.

For more information on the specific steps generated in the WUCF files, see “Error Recovery Scenarios” on page 109.

Error Recovery Scenarios

These scenarios discuss the types of work units that may be found and the steps performed during their execution. Use these scenarios as an aid to problem solving. For information on the error messages and return codes to determine the cause of the failure, see the *Directory Maintenance VM/ESA: Messages*.

AMDISK With No DATAMOVE Interaction

These devices may have been locked on behalf of this transaction. If the failure has occurred before the UNLOCK step (and it was not a candidate for automatic rollback), the device may still be locked.

```
DMM: &DMM.NAME &DMM.NODE
DEV.ONE: &DEV.ONE
DEV.TWO: &DEV.TWO
ORIGNODE: GDLVM7
ORIGUSER: MNTDASD1
ORIGSEQ#: 12
ORIGCMD: AMDISK 0306 3390 AUTOG 500 GDLVM7 MR
SYSAFFIN: *
TARGETID: DSSERV
LANG: AMENG
CMDLEVEL: 140A
ASUSER: MNTDASD1
REQUEST: 12
ORIGEXTENT: N/A
BEGINCMDS:
1 NTRIED WORKUNIT ENABLE
2 NTRIED AMDISK FOR DSSERV 0306 3390 AUTOG 500 GDLVM7 MR
3 NTRIED UNLOCK 0306 DSSERV NOMSG
4 NTRIED WORKUNIT RESET
5 NTRIED DIRECT
```

Figure 21. AMDISK With No DATAMOVE Interaction

The following steps are provided to help you with your “AMDISK With No DATAMOVE Interaction” error recovery.

1 WORKUNIT ENABLE causes object directory updates to be suspended. Failure of this step may indicate a problem using GLOBALV.

2 AMDISK adds an extent to the DATAMOVE machine. Failure on this step probably resulted from an authorization failure. It should have been handled by automatic rollback processing. If the failure occurs after this step, DirMaint has allocated an extent on the DATAMOVE machine that may need to be deallocated.

3 UNLOCK releases the locks on the user's device. Note that you may still have a device lock pending for this user if the failure occurs here.

4 WORKUNIT RESET causes object directory updates to be enabled. Failure of this step may indicate a problem using GLOBALV.

5 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

AMDISK With DATAMOVE Interaction

These devices may have been locked on behalf of this transaction. If the failure has occurred before the UNLOCK step (and it was not a candidate for automatic rollback), the device may still be locked.

```
   DMM: DATAMOV1 GDLVMK1
   DEV.ONE: 100
   DEV.TWO:
   ORIGNODE: GDLVMK1
   ORIGUSER: DIRMAINT
   ORIGSEQ#: 92
   ORIGCMD: AMDISK 6543 3380 1995 1 K1CP04 LABEL MJH191
   SYSAFFIN: *
   TARGETID: HOWLAND9
   LANG: AMENG
   CMDLEVEL: 150A
   ASUSER: DIRMAINT
   REQUEST: 92
   ORIGEXTENT: N/A
   BEGINCMDS:
 1 NTRIED WORKUNIT ENABLE
 2 NTRIED AMDISK FOR DATAMOV1 100 3380 1995 1 K1CP04
 3 NTRIED WORKUNIT RESET
 4 NTRIED DIRECT
 5 NTRIED DMVCTL DATAMOV1 GDLVM7 DMVCTL FORMAT DIRMAINT GDLVMK1 32027026 1 HOWLAND9 6543 * 100 = MJH191
 6 NTRIED WORKUNIT ENABLE
 7 NTRIED TMDISK FOR DATAMOV1 100 HOWLAND9 6543
 8 NTRIED WORKUNIT RESET
 9 NTRIED DIRECT
10 NTRIED UNLOCK 6543 HOWLAND9 NOMSG
```

Figure 22. AMDISK With DATAMOVE Interaction

The following steps are provided help you with your “AMDISK With DATAMOVE Interaction” error recovery.

1 WORKUNIT ENABLE causes object directory updates to be suspended. Failure of this step may indicate a problem using GLOBALV.

2 AMDISK adds an extent to the DATAMOVE machine. Failure on this step probably resulted from an authorization failure. It should have been handled by automatic rollback processing. If the failure occurs after this step, DirMaint has allocated an extent on the DATAMOVE machine that may need to be deallocated.

3 WORKUNIT RESET causes object directory updates to be enabled. Failure of this step may indicate a problem using GLOBALV.

4 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

5 DMVCTL FORMAT requests that the new extent on DATAMOVE be formatted.

6 WORKUNIT ENABLE causes object directory updates to be suspended. Failure of this step may indicate a problem using GLOBALV.

7 TMDISK transfers the extent from DATAMOVE to the user. The new extent has been allocated on DATAMOVE and is now formatted. You may choose to release the locks on the users device and issue a separate TMDISK (from DATAMOVE to the user) to give the user their new extent.

8 WORKUNIT RESET causes object directory updates to be enabled. Failure of this step may indicate a problem using GLOBALV.

9 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

10 UNLOCK releases the locks on the user's device. Note that you may still have a device lock pending for this user if the failure occurs here.

CMDISK

The CMDISK always requires DATAMOVE interaction. These devices may have been locked on behalf of this transaction. If the failure has occurred before the UNLOCK step (and it was not a candidate for automatic rollback), the device may still be locked.

```
   DMM: DATAMOV1 GDLVMK1
   DEV.ONE: 100
   DEV.TWO: 101
   ORIGNODE: GDLVMK1
   ORIGUSER: DIRMAINT
   ORIGSEQ#: 128
   ORIGCMD: CMDISK 0191 XXXX AUTOR 5 REGION1
   SYSAFFIN: *
   TARGETID: HOWLAND9
   LANG: AMENG
   CMDLEVEL: 150A
   ASUSER: DIRMAINT
   REQUEST: 128
   ORIGEXTENT: MDISK 0191 3380 2181 1 K1CP04 MR SAM DOC HARRY
   BEGINCMDS:
 1 NTRIED WORKUNIT ENABLE
 2 NTRIED AMDISK FOR DATAMOV1 100 XXXX AUTOR 5 MIKEPLAY MR SAM DOC HARRY
 3 NTRIED TMDISK FOR HOWLAND9 0191 DATAMOV1 101
 4 NTRIED WORKUNIT RESET
 5 NTRIED DIRECT
 6 NTRIED DMVCTL DATAMOV1 GDLVMK1 DMVCTL COPY DIRMAINT GDLVMK1 04003811 1 HOWLAND9 0191 * 101 100
 7 NTRIED WORKUNIT ENABLE
 8 NTRIED TMDISK FOR DATAMOV1 100  HOWLAND9 0191
 9 NTRIED UNLOCK 0191 HOWLAND9 NOMSG
10 NTRIED DMDISK FOR DATAMOV1 101 NOCLEAN KEEPLINKS
11 NTRIED WORKUNIT RESET
12 NTRIED DIRECT
```

Figure 23. CMDISK

The following steps are provided help you with your “CMDISK” error recovery.

1 WORKUNIT ENABLE causes object directory updates to be suspended. Failure of this step may indicate a problem using GLOBALV.

2 AMDISK adds an extent to the DATAMOVE machine. Failure on this step probably resulted from an authorization failure. It should have been handled by automatic rollback processing. If the failure occurs after this step, DirMaint has allocated an extent on the DATAMOVE machine that may need to be deallocated.

3 TMDISK transfers the extent from the user to DATAMOVE in preparation of the COPY step.

4 WORKUNIT RESET causes object directory updates to be enabled. Failure of this step may indicate a problem using GLOBALV.

5 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

6 DMVCTL COPY copies the old extent (the extent transferred from the user) to the new extent (the extent allocated on DATAMOVE). Failure on this step are usually the result of an attempt to copy a non-CMS disk. Failures that

occurred here are candidates for automatic rollback processing and should have been handled by DirMaint.

7 WORKUNIT ENABLE causes object directory updates to be suspended. Failure of this step may indicate a problem using GLOBALV.

8 TMDISK transfers the new extent from DATAMOVE to the user. Note that at this time both the old and new extents should still exist on the DATAMOVE machine. The data from the old extent should also exist on the new extent and could be transferred directly to the user after releasing any pending device locks.

9 UNLOCK releases the locks on the user's device. Note that you may still have a device lock pending for this user if the failure occurs here.

10 DMDISK deallocates the old extent from the DATAMOVE machine. This is the original user extent. Failure at this point may mean that the original user extent is still associated with the DATAMOVE machine. Although this will not cause problems with the DATAMOVE machine, you may choose to take steps to eliminate the extent.

11 WORKUNIT RESET causes object directory updates to be enabled. Failure of this step may indicate a problem using GLOBALV.

12 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

DMDISK With No DATAMOVE Interaction (NOCLEAN)

These devices may have been locked on behalf of this transaction. If the failure has occurred before the UNLOCK step (and it was not a candidate for automatic rollback), the device may still be locked.

```
   DMM: DATAMOVA GDLVMK1
   DEV.ONE:
   DEV.TWO:
   ORIGNODE: GDLVMK1
   ORIGUSER: DIRMAINT
   ORIGSEQ#: 35
   ORIGCMD: DMDISK 0194 NOCLEAN
   SYSAFFIN: *
   TARGETID: HOWLAND9
   LANG: AMENG
   CMDLEVEL: 150A
   ASUSER: DIRMAINT
   REQUEST: 35
   ORIGEXTENT: N/A
   BEGINCMDS:
1  NTRIED DMDISK FOR HOWLAND9 0194 NOCLEAN
2  NTRIED UNLOCK 0194 HOWLAND9 NOMSG
3  NTRIED DIRECT
```

Figure 24. DMDISK With No DATAMOVE Interaction (NOCLEAN)

The following steps are provided help you with your “DMDISK With No DATAMOVE Interaction (NOCLEAN)” error recovery.

1 DMDISK deallocates the extent from the user directory. Note that you may still have an extent associated with the user.

2 UNLOCK releases the locks on the user's device. Note that you may still have a device lock pending for this user if the failure occurs here.

3 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

DMDISK With DATAMOVE Interaction (CLEAN)

The DMDISK requests with the CLEAN option generate an auxiliary WUCF to perform the actual deallocation and cleaning. This auxiliary WUCF is handled separately from the initial DMDISK WUCF. Because of the asynchronous nature of the DirMaint DASD sub system, the target of the DMDISK may remain attached to the user. During this time device locks prevent activity on it. After the auxiliary WUCF is dispatched and completes, the device will be gone from the user's directory.

```
DMM: DATAMOVA GDLVMK1
DEV.ONE:
DEV.TWO:
ORIGNODE: GDLVMK1
ORIGUSER: DIRMAINT
ORIGSEQ#: 38
ORIGCMD: DMDISK 0194 CLEAN
SYSAFFIN: *
TARGETID: HOWLAND9
LANG: AMENG
CMDLEVEL: 150A
ASUSER: DIRMAINT
REQUEST: 38
ORIGEXTENT: N/A
BEGINCMDS:
1 NTRIED DMDISK FOR HOWLAND9 0194 CLEAN
```

Figure 25. DMDISK With No DATAMOVE Interaction (CLEAN)

The following steps are provided help you with your “DMDISK With DATAMOVE Interaction (CLEAN)” error recovery.

1 DMDISK generates an auxiliary WUCF that will transfer the extent to DATAMOVE, clean it and then deallocate the extent. Failure during this step may indicate that there were problems associated with the creation of another WUCF. Note, depending on the point of failure, there may still be an extent associated with the user at this point.

ZAPMDISK (Auxiliary DMDISK)

ZAPMDISK requests are generated on behalf of a DMDISK CLEAN request. The ZAPMDISK handles the cleaning and deallocation of the extent.

```
   DMM: DATAMOVA GDLVMK1
   DEV.ONE: 100
   DEV.TWO:
   ORIGNODE: GDLVMK1
   ORIGUSER: DIRMAINT
   ORIGSEQ#: 38.2
   ORIGCMD: ZAPMDISK HOWLAND9 0194
   SYSAFFIN: *
   TARGETID: HOWLAND9
   LANG: AMENG
   CMDLEVEL: 150A
   ASUSER: DIRMAINT
   REQUEST: 38.2
   ORIGEXTENT: N/A
   BEGINCMDS:
 1 NTRIED WORKUNIT ENABLE
 2 NTRIED TMDISK FOR HOWLAND9 0194 DATAMOVA 100
 3 NTRIED WORKUNIT RESET
 4 NTRIED DIRECT
 5 NTRIED UNLOCK 0194 HOWLAND9 NOMSG
 6 NTRIED DMVCTL DATAMOVA GDLVMK1 DMVCTL CLEAN DIRMAINT GDLVMK1 37537049 1 HOWLAND9 0194 * 100
 7 NTRIED WORKUNIT ENABLE
 8 NTRIED DMDISK FOR DATAMOVA 100 NOCLEAN KEEPLINKS HOWLAND9
 9 NTRIED WORKUNIT RESET
10 NTRIED DIRECT
```

Figure 26. ZAPMDISK (Auxiliary DMDISK)

The following steps are provided help you with your “ZAPMDISK (Auxiliary DMDISK)” error recovery.

1 WORKUNIT ENABLE causes object directory updates to be suspended. Failure of this step may indicate a problem using GLOBALV.

2 TMDISK transfers the users extent to a DATAMOVE machine in preparation for cleaning. Note that the extent may still be associated with the original user.

3 WORKUNIT RESET causes object directory updates to be enabled. Failure of this step may indicate a problem using GLOBALV.

4 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

5 UNLOCK releases the locks on the user's device. Note that you may still have a device lock pending for this user if the failure occurs here.

6 DMVCTL CLEAN cleans the extent on the DATAMOVE machine. Note that the extent may remain uncleaned and attached to the DATAMOVE machine.

7 WORKUNIT ENABLE causes object directory updates to be suspended. Failure of this step may indicate a problem using GLOBALV.

8 DMDISK deallocates the extent from the DATAMOVE machine. This DMDISK command explicitly uses the NOCLEAN option to force a simple deallocation of the associated extents. Note that the newly cleaned extent may remain attached to the DATAMOVE machine.

9 WORKUNIT RESET causes object directory updates to be enabled. Failure of this step may indicate a problem using GLOBALV.

10 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

TMDISK

These devices may have been locked on behalf of this transaction; one device represents the source user's device and the other device represents the target user's device. If the failure has occurred before the UNLOCK steps, one or both devices may still be locked.

```
   DMM: &DMM.NAME &DMM.NODE
   DEV.ONE: &DEV.ONE
   DEV.TWO: &DEV.TWO
   ORIGNODE: GDLVMK1
   ORIGUSER: DIRMAINT
   ORIGSEQ#: 8
   ORIGCMD: TMDISK 9999 TO HOWLAND9 0191
   SYSAFFIN: *
   TARGETID: HOWLAND2
   LANG: AMENG
   CMDLEVEL: 150A
   ASUSER: DIRMAINT
   REQUEST: 8
   ORIGEXTENT: N/A
   BEGINCMDS:
1  NTRIED WORKUNIT ENABLE
2  NTRIED TMDISK FOR HOWLAND2 9999 HOWLAND9 0191
3  NTRIED WORKUNIT RESET
4  NTRIED DIRECT
5  NTRIED UNLOCK 0191 HOWLAND9 NOMSG
6  NTRIED UNLOCK 9999 HOWLAND2 NOMSG
```

Figure 27. TMDISK

The following steps are provided help you with your “ZAPMDISK (Auxiliary DMDISK)” on page 116 error recovery.

1 WORKUNIT ENABLE causes object directory updates to be suspended. Failure of this step may indicate a problem using GLOBALV.

2 TMDISK transfers the source device to the target user and device.

3 WORKUNIT RESET causes object directory updates to be enabled. Failure of this step may indicate a problem using GLOBALV.

4 DIRECT causes object directory to be placed online if current settings allow. Failure on this command may indicate problem with your source directory.

5 UNLOCK target unlocks the target user's device address. Note that you may still have a device lock pending for this user if the failure occurs here.

6 UNLOCK source unlocks the source user's device address. Consult the error message and return codes to determine the exact cause of the failure. Note that you may still have a device lock pending for this user if the failure occurs here.

Chapter 7. User Tailoring

This chapter will show you how you can tailor exit routines and data files with commands. Your system administrator has already applied the IBM defaults during installation of DirMaint Release 5.0, therefore, no user tailoring is required. However, a user can customize their workstation to their needs. You should read this chapter at your terminal, reviewing the examples as you read the text.

The ACCESS DATADVH File

The ACCESS DATADVH file is a required file intended for local tailoring, but will default to the DIRMAINT 11F disk. This allows you to modify how commands and data files are routed.

The DIRMAINT EXEC obtains access to the interface files by using the *nodeid* entries found in the SYSTEM NETID and ACCESS DATADVH files. The production level of these two files generally resides on the 19E disk of the MAINT machine, although the test level of these files generally resides on the 29E disk of the P748XE4M machine.

A common error that occurs is having multiple copies of the ACCESS DATADVH file in the search order, with the wrong one accessed ahead of the other. This can be detected by use of the DIRM CHECK command. For more information on the DIRM CHECK command, see *Directory Maintenance VM/ESA: Command Reference*.

When accessing the ACCESS DATADVH file, you should choose one of these formats. Enter:

```
ON= nodeid  USE= server:owner.subdirectory
ON= nodeid  USE= owner vaddr <rpass>
```

Or you can choose to use the IBM-supplied default format. Enter:

```
ON= *  USE= P748XE4M 11F ALL
```

Where:

ON=
Specifies *nodeid*, this keyword must be followed by at least one blank column.

nodeid
Identifies the node of the user entering the allocation request.

***** Specifies a default entry for use if no other entry matches the user's *nodeid*.

USE=
Specifies the interface or the qualified path name of a shared file directory; this keyword must be followed by at least one blank column.

server:owner.subdirectory
Identifies the *userid* that owns the interface or the qualified path name of a shared file directory.

or

owner vaddr
Specifies the virtual machine *userid* and the minidisk address containing the user interface files or blanks if you are using a shared file directory.

Note: The DVHPROFX EXEC file must be updated if a disk address other than 11F or an SFS directory is used.

rpass
Specifies the read-share password needed to obtain a link for the interface disk, or blanks if you are using a shared file directory.

Notes:

1. If you are using the password of *ALL*, then *ALL* should be specified in the ACCESS DATADVH entry.
2. If an ESM is installed, the password may or may not be required, or may be required but ignored.
3. If you are using RACF as your ESM with DISKP=ALLOW on the SYSSEC macro, a password is not required in the ACCESS DATADVH entry and is ignored if specified. If you are using RACF with DISKP=DEFER, then the ACCESS DATADVH entry must supply the correct link password.

For more information, see your ESM documentation. If you are using IBM RACF, refer to *RACF Macros and Interfaces*, SC28-1345.

The CONFIG* DATADVH File

The CONFIG* DATADVH file is created with the entry keywords as shown in Table 11 on page 121. DirMaint Release 5.0 allows entries in multiple CONFIG* DATADVH files, therefore, it is not necessary to duplicate the entire file to supplement or override a single line as in prior releases.

CONFIG* DATADVH File Example

An example of a CONFIG* DATADVH file is shown in Figure 28 on page 121. Because of the size of the CONFIG* DATADVH file, the entries are abbreviated in this example.

```
 1 REQUIRED_USER_FILE= DVHCMD EXEC
 2 LOADABLE_USER_FILE= DIRMAINT EXEC    Recommended.
 3 DEFAULT_CMDSET.140A= G
 4 SAMPL_USER_MSGS_1x0A=
 5 KANJI_USER_MSGS_140A= LCLAUSER MSGKDVH
 6 KANJI_BATCH_HEADER_140A= DVHBHEAD DATAKDVH
 7 KANJI_HELP_140A= DIRM HELPDIRM
 8 KANJI_MENU_DEFS_150= DVHMENUS DATAKDVH
 9 PARSER_140A= DVHADZ   EXEC
10 COMMANDS_140A= 140CMDS  DATADVH
11 PASSWORD_RANDOM_GENERATOR_USER_EXIT= DVHPXR   EXEC   (Required)
12 FROM= VMESA210  DEST= VMESA122 S= DIRMAINT  T= VMESA122
13 PW_MIN_LENGTH= 3
14 PW_REUSE_HASHING_EXIT=
15 PW_REUSE_INTERVAL=
16 SAMPL_LINESIZE_140A= 222
17 SAMPL_LINESIZE_150A= 222
```

Figure 28. CONFIG DATADVH File*

Table 11 (Page 1 of 2). Summary of CONFIG DATADVH File Entries*

Entry Keyword	Function
1 REQUIRED_USER_FILE=	Defines the files needed in the user's virtual machine to enter any DIRMAINT commands.
2 LOADABLE_USER_FILE=	Defines the user file to be made resident or nonresident by using the EXECLOAD and EXECDROP commands.
3 DEFAULT_CMDSET.1x0A=	Identifies the general user command set for each command level if a local user has not been explicitly authorized for use of any privileged command sets.
4 SAMPL_USER_MSGS_1x0A=	The SAMPL entry provides an example of creating a custom language for a special application. The SAMPUSER message repository may reassign message numbers and severities, may rephrase the message text or suppress the message entirely, and may change the return code passed back when the message is issued.
National Language Support	Defines a set of online directions, Help files, menus, and messages by using these files: • 5 *lang*_USER_MSGS_1x0A= • 6 *lang*_BATCH_HEADER_1x0A= • 7 *lang*_HELP_1x0A= • 8 *lang*_MENU_DEFS_1x0A=
9 PARSER_1x0A=	Defines the command entered by the user, verifies it is syntactically correct, expands keyword abbreviations to their full length, extracts selected information from and about the command, and makes it available to other parts of the product.
10 COMMANDS_1x0A=	Defines the file name of the handler routine. This will determine what machine will process the command.
11 Various _USER_EXIT=	Defines alternative processing options to be performed.
12 FROM= DEST=	Defines the necessary route for a command or file from the system or to route messages or files from the DIRMAINT service machine back to the user.

Table 11 (Page 2 of 2). Summary of CONFIG DATADVH File Entries*

Entry Keyword	Function
13 PW_MIN_LENGTH=	Defines a security check of the user's password.
14 PW_REUSE_HASHING_EXIT	Defines a routine to hash the user's password for storage in the password history file. The file type may be either EXEC or MODULE. The default is DVHHASH MODULE. If not specified, the passwords will be stored in the history file as hexadecimal digits.
15 PW_REUSE_INTERVAL	Identifies how long an entry is kept in the password history file. This can be either a time period with a DAYS suffix, or a count with no suffix. The default is 365 DAYS.
16 SAMPL_LINESIZE_140A= 222 17 SAMPL_LINESIZE_150A= 222	By default, DirMaint will dynamically select a message output length of either 52 or 73 characters. User's may select a "language" whose messages are formatted for a line length other than the default. **Note:** The maximum linesize is equal to 222; because the maximum length of the CP command buffer is 240, minus 9 for the user ID and intervening blank, minus 10 for the CP MSGNOH command and another blank. The minimum value is 40.

Notes:

- Blank lines and comments (lines starting with a slash (/)) are allowed to enhance readability.

  The IBM convention is to use the delimiter **/*** before and ***/** after prologues, directions, and other readable information.
- An inactive machine readable entry will have a **/** in the prefix area.
- A common error that occurs is to have multiple copies of this file in the search order with the wrong one accessed ahead of the other. This can be detected with the DIRM CHECK command.

Example—Fragments from the CONFIG and CONFIGAA DATADVH Files

```
⋮
1 LOADABLE_USER_FILE= DVHCMD EXEC
1 LOADABLE_USER_FILE= DVHMSG EXEC
1 LOADABLE_USER_FILE= DVHXMIT EXEC
2 PASSWORD_RANDOM_GENERATOR_USER_EXIT= DVHPXR EXEC
⋮
```

Figure 29. CONFIG DATADVH File

```
⋮
1 LOADABLE_USER_FILE= DVHFILE EXEC
1 LOADABLE_USER_FILE= DVHADZ EXEC
2 USER_EXIT= MYPXR EXEC
⋮
```

Figure 30. CONFIGAA DATADVH File

The following notes are to help you with your Figure 29 and Figure 30.

1 Some entries in the CONFIG* DATADVH files appear multiple times and are cumulative. If there are three LOADABLE_USER_FILE= entries in the base CONFIG DATADVH file and two more in a CONFIGAA DATADVH file, all five files are loadable.

2 Other entries are alternatives. If specified more than once in a single CONFIG DATADVH file or in multiple CONFIG* DATADVH files, only the first entry encountered is used. If there is a PASSWORD_RANDOM_GENERATOR_USER_EXIT= entry in both a CONFIG DATADVH file and in a CONFIGAA DATADVH file, only the entry

in the CONFIGAA DATADVH file will be used. This makes the order that the files are searched important. The files are searched in REVERSE alphanumeric order: CONFIG99 before CONFIG0, CONFIG0 before CONFIGZZ, CONFIGZZ before CONFIGA, and CONFIGA before CONFIG. If two files have the same file name, only the file on the disk or directory with the lowest file mode letter is searched.

The REQUIRED_USER_FILE= Entries

The REQUIRED_USER_FILE= entries are the files that must be present for the user's virtual machine to correctly issue any DIRMAINT commands.

```
// REQUIRED_USER_FILE= ACCESS   DATADVH Already checked.
// REQUIRED_USER_FILE= CONFIG*  DATADVH Already checked.
   REQUIRED_USER_FILE= WHERETO  DATADVH
   REQUIRED_USER_FILE= 140CMDS  DATADVH
   REQUIRED_USER_FILE= 150CMDS  DATADVH
   REQUIRED_USER_FILE= DVHULVL  DATADVH
// REQUIRED_USER_FILE= DIRMAINT EXEC    Already checked.
   REQUIRED_USER_FILE= DVHADZ   EXEC
   REQUIRED_USER_FILE= DVHAEZ   EXEC
   REQUIRED_USER_FILE= DVHCMD   EXEC
   REQUIRED_USER_FILE= DVHCEXIT EXEC
// REQUIRED_USER_FILE= DVHCXB   EXEC
// REQUIRED_USER_FILE= DVHCXA   EXEC
   REQUIRED_USER_FILE= DVHFILE  EXEC
   REQUIRED_USER_FILE= DVHFNDCS EXEC
   REQUIRED_USER_FILE= DVHGLBLV EXEC
   REQUIRED_USER_FILE= DVHHELP  EXEC
   REQUIRED_USER_FILE= DVHMSG   EXEC
   REQUIRED_USER_FILE= DVHPWC   EXEC
   REQUIRED_USER_FILE= DVHPXR   EXEC
   REQUIRED_USER_FILE= DVHPXV   EXEC
// REQUIRED_USER_FILE= DVHPXA   EXEC
   REQUIRED_USER_FILE= DVHUCHK  EXEC
   REQUIRED_USER_FILE= DVHXMIT  EXEC
// REQUIRED_USER_FILE= COMPARE  MODULE
// REQUIRED_USER_FILE= EXECDROP MODULE
// REQUIRED_USER_FILE= NUCXDROP MODULE
   REQUIRED_USER_FILE= PIPE     MODULE  Do we have a supported CMS level?
// REQUIRED_USER_FILE= VMFCLEAR MODULE
// REQUIRED_USER_FILE= 140AUSER MSGADVH
   REQUIRED_USER_FILE= 150AUSER MSGADVH
```

Figure 31. REQUIRED_USER_FILE= Entries

The following notes are to help you with “The REQUIRED_USER_FILE= Entries.”

Notes:

1. The REQUIRED_USER_FILE= statements must be followed by both a file name and a file type.
2. The DIRMAINT EXEC ensures that all listed files are present before continuing with command processing. If one or more files are not found, the DIRMAINT EXEC will use the information in the ACCESS DATADVH file to link and access the minidisk with the user interface files, or to access the shared file directory

with those files. If one or more required files are still not found, the DIRMAINT EXEC issues an error message and exits with a nonzero return code.

3. The IBM-supplied required files listing includes all files supplied by IBM that are expected to reside on the user interface disk or directory. Any local user exit routines or new user machine command handling routines should be added to this listing. Performance can be improved by reducing the number of entries on the listing.
4. This listing of REQUIRED_USER_FILE= entries, may not be appropriate for everyone. You can create a separate CONFIG* DATADVH file, perhaps with the name CONFIGRU DATADVH, to contain all of the REQUIRED_USER_FILE= entries and remove them from the IBM-supplied CONFIG DATADVH file. Each virtual machine can then customize its own private copy of the CONFIGRU DATADVH file without needing to duplicate the entire base CONFIG DATADVH file.

The LOADABLE_USER_FILE= Entries

The LOADABLE_USER_FILE= entries are the user files that are made resident or nonresident by the DIRM EXECLOAD and DIRM EXECDROP commands.

```
   LOADABLE_USER_FILE= DIRMAINT EXEC    Recommended.
   LOADABLE_USER_FILE= DVHMSG   EXEC    Recommended.
   LOADABLE_USER_FILE= DVHADZ   EXEC    Recommended.
   LOADABLE_USER_FILE= DVHAEZ   EXEC    Recommended.
   LOADABLE_USER_FILE= DVHCMD   EXEC    Recommended.
   LOADABLE_USER_FILE= DVHFNDCS EXEC    Recommended.
   LOADABLE_USER_FILE= DVHCEXIT EXEC    Recommended.
/  LOADABLE_USER_FILE= DVHCXB   EXEC    Recommended.
   LOADABLE_USER_FILE= DVHCXA   EXEC    Recommended.
   LOADABLE_USER_FILE= DVHXMIT  EXEC    Recommended.
   LOADABLE_USER_FILE= DVHFILE  EXEC    Recommended.
/  LOADABLE_USER_FILE= DVHPWC   EXEC    Recommend individual tailoring.
/  LOADABLE_USER_FILE= DVHPXR   EXEC    Recommend individual tailoring.
/  LOADABLE_USER_FILE= DVHPXV   EXEC    Recommend individual tailoring.
/  LOADABLE_USER_FILE= DVHPXA   EXEC    Recommend individual tailoring.
// LOADABLE_USER_FILE= DVHEXLD  EXEC    Probably not worthwhile.
// LOADABLE_USER_FILE= DVHGLBLV EXEC    Probably not worthwhile.
// LOADABLE_USER_FILE= DVHHELP  EXEC    Probably not worthwhile.
// LOADABLE_USER_FILE= DVHVCHK  EXEC    Probably not worthwhile.
```

Figure 32. LOADABLE_USER_FILE= Entries

The following notes are to help you with “The LOADABLE_USER_FILE= Entries.”

Notes:

1. The LOADABLE_USER_FILE= statements must be followed by both a file name and file type. The file type must be either EXEC, MODULE, REXX, or XEDIT.
2. The DIRMAINT EXEC will usually release the shared file directory or release and detach the minidisk containing the user interface files on completion of a DIRM command if it did the link and access before processing the next command. However, it will leave the disk or directory accessed if all required files were found without doing the link and access before processing the command. The DIRM EXECLOAD and DIRM EXECDROP commands are exceptions to this general rule. The disk or directory will always remain

accessed following a DIRM EXECLOAD command, and will always be released and detached after a DIRM EXECDROP command.

3. The virtual machines that spend a substantial amount of their time running DirMaint commands may benefit by making parts of the DirMaint program resident, thus saving on file I/O time for each command entered. This is a trade off, as making DirMaint files resident uses storage, which may impact other programs that run in the same virtual machine either concurrently or consecutively. Files are made resident by entering a DIRM EXECLOAD command; and can be made nonresident by entering a DIRM EXECDROP command.
4. The IBM-supplied loadable files listing includes all performance critical files supplied by IBM that reside on the user interface disk or directory. These files are generally used by system administrators. Any local user exit routines or new user machine command handling routines should be added to this listing. Performance can be improved by listing all frequently used files.
5. This listing of LOADABLE_USER_FILE= entries may not be appropriate for everyone. You can create a separate CONFIG* DATADVH file, perhaps with the name CONFIGLU DATADVH, to contain all of the LOADABLE_USER_FILE= entries and remove them from the IBM supplied CONFIG DATADVH file. Each virtual machine can then customize its own private copy of the CONFIGLU DATADVH file without needing to duplicate the entire base CONFIG DATADVH file.

The DEFAULT_CMDLEVEL= Entry

The DEFAULT_CMDLEVEL value determines which messages and commands parsing files should be used when the user has not entered a DIRM DEFAULTS CMDLEVEL command to select their own default CMDLEVEL.

DirMaint Release 5.0 supports two command levels:

- The 150A level provides all of the function supported in Release 5, using the Release 5 preferred command syntax. IBM encourages users sitting in front of a terminal to use the full function 150A command level.
- The 140A level provides all of the function supported in Release 4 that remains in Release 5, using the Release 4 compatibility command syntax. Command level 140A is intended for use by programs that have not been changed to use the 150A command syntax, allowing the service virtual machines DFSMS, DSO, IPF, NVAS, RACF, and so forth to run without changes, even if the administrator and the general user population exploit the full capabilities of DirMaint Release 5.

Each virtual machine can select its own command level by issuing a DIRM DEFAULTS CMDLEVEL 1x0A command.

If multiple occurrences of this keyword are encountered in the CONFIG* DATADVH files, the first occurrence will be used.

For those virtual machines that have not issued a DIRM DEFAULTS CMDLEVEL command, the default command level is determined by a DEFAULT_CMDLEVEL= entry in the CONFIG* DATADVH file. The IBM-supplied default is:

```
DEFAULT_CMDLEVEL= 150A
```

If possible, IBM recommends that all virtual machines running programs that issue DirMaint commands using the Release 4 syntax issue a DIRM DEFAULTS CMDLEVEL 140A command. Otherwise, the DEFAULT_CMDLEVEL must be changed to 140A.

National Language Support

The national language support files identify the language dependent files needed for the user's active language.

```
AMENG_BATCH_HEADER_140A= DVHBHEAD DATAADVH
AMENG_BATCH_HEADER_150A= DVHBHEAD DATAADVH
AMENG_COPYRIGHT_NOTICE=  DVHCOPYR DATAADVH
AMENG_HELP_140A=         DIRM     HELPDIRM
AMENG_HELP_150A=         DVHAMENG HELPADVH
AMENG_MENU_DEFS_150A=    DVHMENUS DATAADVH
AMENG_USER_MSGS_140A=    LCLAUSER MSGADVH
AMENG_USER_MSGS_140A=    150AUSER MSGADVH
AMENG_USER_MSGS_150A=    LCLAUSER MSGADVH
AMENG_USER_MSGS_150A=    150AUSER MSGADVH
UCENG_BATCH_HEADER_140A= DVHBHEAD DATAUDVH
UCENG_BATCH_HEADER_150A= DVHBHEAD DATAUDVH
UCENG_COPYRIGHT_NOTICE=  DVHCOPYR DATAUDVH
UCENG_HELP_140A=         DIRM     HELPDIRM
UCENG_HELP_150A=         DVHUCENG HELPUDVH
UCENG_MENU_DEFS_150A=    DVHMENUS DATAADVH
UCENG_USER_MSGS_140A=    LCLAUSER MSGUDVH
UCENG_USER_MSGS_140A=    140AUSER MSGUDVH
UCENG_USER_MSGS_140A=    150AUSER MSGUDVH
UCENG_USER_MSGS_150A=    LCLAUSER MSGUDVH
UCENG_USER_MSGS_150A=    150AUSER MSGUDVH
KANJI_BATCH_HEADER_140A= DVHBHEAD DATAKDVH
KANJI_BATCH_HEADER_150A= DVHBHEAD DATAKDVH
KANJI_COPYRIGHT_NOTICE=  DVHCOPYR DATAKDVH
KANJI_HELP_140A=         DIRM     HELPDIRM
KANJI_HELP_150A=         DVHUCENG HELPKDVH
KANJI_MENU_DEFS_150A=    DVHMENUS DATAUDVH
KANJI_USER_MSGS_140A=    LCLAUSER MSGKDVH
KANJI_USER_MSGS_140A=    140AUSER MSGKDVH
KANJI_USER_MSGS_140A=    150AUSER MSGKDVH
KANJI_USER_MSGS_150A=    LCLAUSER MSGKDVH
KANJI_USER_MSGS_150A=    150AUSER MSGKDVH
1SAPI_BATCH_HEADER_140A= DVHBHEAD DATAADVH
1SAPI_BATCH_HEADER_150A= DVHBHEAD DATAADVH
1SAPI_COPYRIGHT_NOTICE=  DVHCOPYR DATAADVH
1SAPI_HELP_140A=         DIRM     HELPDIRM
1SAPI_HELP_150A=         DVHAMENG HELPADVH
1SAPI_MENU_DEFS_150A=    DVHMENUS DATAADVH
1SAPI_USER_MSGS_140A=    LCLAUSER MSG1DVH
1SAPI_USER_MSGS_140A=    140AUSER MSG1DVH
1SAPI_USER_MSGS_140A=    150AUSER MSG1DVH
1SAPI_USER_MSGS_150A=    LCLAUSER MSG1DVH
1SAPI_USER_MSGS_150A=    150AUSER MSG1DVH
```

Figure 33. National Language Support

The following notes are to help you with “National Language Support.”

Notes:

1. Each command level has its own set of online directions, Help files, menus, and messages. DirMaint Release 5.0 supports translation of each of these varieties of information into any left-to-right language whose character set is supported by VM. IBM provides all of these files in mixed case American English (AMENG), with instructions for conversion to upper case English (UCENG), and supplies messages in Japanese (KANJI).
2. The virtual machine can select its own language by issuing a DIRM GLOBALV LANG *xxxxx* command. This may be, but need not be, the same language chosen for CMS using the SET LANG command. If a DIRM DEFAULT LANG command has not been issued, DirMaint will use a QUERY LANG command to determine the language.
3. To avoid repeating entries in the CONFIG* DATADVH file for each language, a series of five dots, ..., identifies a default that applies to any language that does not have its own entry.

Example—Identify a Default Entry in the CONFIG* DATADVH File:

```
....._BATCH_HEADER_140A= DVHBHEAD DATAADVH
....._BATCH_HEADER_150A= DVHBHEAD DATAADVH
....._COPYRIGHT_NOTICE=  DVHCOPYR DATAADVH
....._HELP_140A=         DIRM     HELPDIRM
....._HELP_150A=         DVHAMENG HELPADVH
....._MENU_DEFS_150A=    DVHMENUS DATAADVH
....._USER_MSGS_140A=    LCLAUSER MSGADVH
....._USER_MSGS_140A=    150AUSER MSGADVH
....._USER_MSGS_150A=    LCLAUSER MSGADVH
....._USER_MSGS_150A=    150AUSER MSGADVH
```

This listing of national language choices may not be appropriate for everyone. You can create a separate CONFIG* DATADVH file, perhaps with the name CONFIGNL DATADVH, to contain all of the language related entries and remove them from the IBM-supplied CONFIG DATADVH file. Each virtual machine can then customize its own private copy of the CONFIGNL DATADVH file without needing to duplicate the entire base CONFIG DATADVH file.

The lang_BATCH_HEADER_1x0A= Entries

When editing a batch file, a set of directions will be shown. These directions describe how to submit the batch commands and how to cancel the commands if you decide not to submit them. A different set of directions may be used for each command level. Only the first entry is used for each language and command level. The IBM-supplied defaults are:

```
KANJI_BATCH_HEADER_140A= DVHBHEAD DATAKDVH
KANJI_BATCH_HEADER_150A= DVHBHEAD DATAKDVH
UCENG_BATCH_HEADER_140A= DVHBHEAD DATAUDVH
UCENG_BATCH_HEADER_150A= DVHBHEAD DATAUDVH
....._BATCH_HEADER_140A= DVHBHEAD DATAADVH
....._BATCH_HEADER_150A= DVHBHEAD DATAADVH
```

Each entry identifies the file name and file type of the file containing the directions for that combination of language and command level.

If multiple occurrences of this keyword are encountered in the CONFIG* DATADVH file, the first occurrence will be used.

The lang_COPYRIGHT NOTICE= Entries

The first time you invoke a DirMaint Release 5.0 command, and approximately once per month thereafter, a copyright notice will be displayed for several seconds. These entries identify the file name and file type of the file to be displayed for the user's selected language. If you have multiple entries, the first occurrence of the entry is the language used.

Example—lang_COPYRIGHT NOTICE= Entry

```
UCENG_C_N= DVHCOPYR DATAUDVH
KANJI_C_N= DVHCOPYR DATAKDVH
DVHCOPYR DATAADVH
```

The lang_HELP_1x0A= Entries

The lang_HELP_1x0A= identify supply the Online Help information available. The IBM-supplied Help files are in mixed case American English, with instructions available on conversion to Upper Case English. Each supported language identifies which files to use. If multiple entries are specified for the same language and command level, the first entry encountered is used. The IBM-supplied defaults are:

```
KANJI_HELP_140A= DIRM HELPDIRM
KANJI_HELP_150A= DVHAMENG HELPADVH
UCENG_HELP_140A= DIRM HELPDIRM
UCENG_HELP_150A= DVHUCENG HELPUDVH
....._HELP_140A= DIRM HELPDIRM
....._HELP_150A= DVHAMENG HELPADVH
```

Each entry specifies a file name and file type, although not for the same file. The file name is the name of the HELPMENU file used when a DIRM HELP command is entered without specifying a topic name. The file type is the file type of the online HELP file used when a DIRM HELP topic_name command is entered.

If multiple occurrences of this keyword are encountered in the CONFIG* DATADVH file, the first occurrence will be used.

The lang_MENU__DEFS_1x0A= Entries

Most DirMaint commands can be submitted by filling in a menu panel. The menu panel definitions are contained in a file for which the file name and file type must be specified. The IBM-supplied defaults are:

```
KANJI_MENU_DEFS_150A= DVHMENUS DATAADVH
UCENG_MENU_DEFS_150A= DVHMENUS DATAUDVH
....._MENU_DEFS_150A= DVHMENUS DATAADVH
```

Each entry specifies a file name and file type of the menu definition file. If multiple occurrences of this keyword are encountered in the CONFIG* DATADVH file, the first occurrence will be used.

Note: IBM has chosen not to supply menus for command level 140A. Command level 140A is intended for use by programs that have not been changed to use the 150A command syntax. IBM encourages users sitting in front of a terminal to use the full function 150A command level.

The lang_USER_MSGS_1x0A= Entries

DirMaint will search multiple message repositories when looking for the text of a message. The IBM-supplied defaults are:

```
KANJI_USER_MSGS_140A= 140AUSER MSGKDVH
KANJI_USER_MSGS_140A= 150AUSER MSGKDVH
UCENG_USER_MSGS_140A= 140AUSER MSGUDVH
UCENG_USER_MSGS_140A= 150AUSER MSGUDVH
....._USER_MSGS_140A= 140AUSER MSGADVH
....._USER_MSGS_140A= 150AUSER MSGADVH
KANJI_USER_MSGS_150A= 150AUSER MSGKDVH
UCENG_USER_MSGS_150A= 150AUSER MSGUDVH
....._USER_MSGS_150A= 150AUSER MSGADVH
```

The following notes are to help you with “The lang_USER_MSGS_1x0A= Entries.”

Notes:

1. Each file name and file type of the repositories to be searched must all be listed in the CONFIG* DATADVH file. A different set of repositories may be used for each command level.
2. The entry identifies the file name and file type of the message repository file for that combination of language and command level.

   The entries for command level 140A use both the 140AUSER and 150AUSER repositories, although the entries for command level 150A use only the 150AUSER repositories. The 140AUSER repositories contain overrides for the 150AUSER repositories. In many but not all cases, this allows messages to be issued with the same routine identification, message number, severity, message text, and return code that Release 4 used for the similar condition. There are messages issued by Release 4 for which there is no similar condition in Release 5.0; adding the message to the 140AUSER repository will not cause the message to be issued. There are messages issued by Release 5.0 for which there is no similar condition in Release 4. So the messages are issued with the Release 5.0 routine identifiers and message numbers.

   If your site needs to create overrides, enter:

   ```
   ....._USER_MSGS_140A= LCLAUSER MSGADVH
   ....._USER_MSGS_150A= LCLAUSER MSGADVH
   ```

   and insert them into the file preceding the IBM-supplied entries, or include them in a separate CONFIG* DATADVH file with a name (CONFIGZZ perhaps) that will be searched before the IBM-supplied file.

Messages and Return Codes

Messages consist of a message identifier and message text. The identifier distinguishes messages from each other. The text is a phrase or sentence that either describes a condition that has occurred or requests a response from a user. The Synchronous Application Programming Interface Language (SAPI) may be used by programs that need to interpret DirMaint's messages.

Example—Message Formats: The format of most message identifiers is:

```
DVHABC1234S MSG= 1234 FMT= 01 SEV= S RTN= DVHABC SUBS= s1 s2 s3 ...
```

If you, Enter:

```
DIRM review
```

An AMENG message response might be:

```
DVHREQ2289I Command REVIEW complete; RC = 0.
```

The corresponding 1SAPI message response would be:

```
DVHREQ2289I MSG= 2289 FMT= 01 SEV= I RTN= DVHREQ SUBS= 0 REVIEW
```

Where:

DVH
Prefix identifier

ABC
An abbreviation of the routine name of the routine for which the error occurred.

1234
The numeric message number consists of three or four digits that are associated with the condition that caused the message to be generated.

S A letter that shows a severe error message. The severity code values are:

A	User action is required.
E	Error message
I	Information message
R	User response is required.
W	Warning message
T	Terminating error message.

MSG= 1234
Identifies the message number

FMT= 01
Identifies the message format

SEV= S
Identifies the message severity code

RTN= DVHABC
Identifies the message routine name

SUBS= s1 s2 s3 ...
Identifies the message variable information elements for substitution into the message text.

The PARSER_1x0A= Entries

All DirMaint commands must be *parsed*. The parser ensures that the command entered by the user has the correct syntax expands keyword abbreviations to their full length, and extracts selected information from and about the command and makes it available to other parts of the product. Each command level has its own parser. These are identified by entries in the CONFIG* DATADVH file. The IBM-supplied defaults are:

```
PARSER_140A= DVHADZ EXEC
PARSER_150A= DVHAEZ EXEC
```

Each PARSER_1x0A= entry must be followed by both a file name and file type. The file type must be EXEC or MODULE.

If multiple occurrences of this keyword are encountered in the CONFIG* DATADVH file, the first occurrence will be used.

The COMMANDS_1x0A= Entries

After the syntax of the command has been validated, the command must be *handled*. A data file determines the file name of the *handler* routine. This occurs whether the command is sent to the DIRMAINT service machine or is completely processed in the issuing user's virtual machine. If sent to the DIRMAINT service machine, a data file determines the file name of the *handler* routine whether password validation is required and whether the command is available in the general user command set or whether authorization for a privileged command set is required for use of the particular command. Each command level has a separate data file containing this information. The IBM-supplied defaults are:

```
COMMANDS_140A=    LCLCMDS  DATADVH
COMMANDS_140A=    140CMDS  DATADVH
COMMANDS_150A=    LCLCMDS  DATADVH
COMMANDS_150A=    150CMDS  DATADVH
```

Each COMMANDS_1x0A= entry must be followed by both a file name and file type.

If multiple occurrences of this keyword are encountered in the CONFIG* DATADVH file, they will be searched in the order specified until the command definition is found or the list is exhausted. Thus, if your site changes the command set required for use of a particular command, you can include that one command in a separate file (LCLCMDS DATADVH for example) rather than modify the IBM-supplied file. You can then list that file either in the CONFIG DATADVH file before the IBM supplied entries, or in a separate CONFIG* DATADVH file (CONFIGZZ for example) that is searched before the CONFIG DATADVH file.

The Various USER_EXIT= Entries

The Various _USER_EXIT= entries are used at several points during processing of a command. There are alternative implementations that may be chosen or special site specific functions that may need to be performed. These are handled by various exit routines. The IBM-supplied defaults are:

```
COMMAND_BEFORE_PARSING_USER_EXIT=    DVHCXC EXEC
COMMAND_BEFORE_PROCESSING_USER_EXIT= DVHCXB EXEC
COMMAND_AFTER_PROCESSING_USER_EXIT=  DVHCXA EXEC  (sample)
PASSWORD_RANDOM_GENERATOR_USER_EXIT= DVHPXR EXEC  (required)
PASSWORD_SYNTAX_CHECKING_USER_EXIT=  DVHPXV EXEC  (sample)
PASSWORD_NOTIFICATION_USER_EXIT=     DVHPXA EXEC  (sample)
```

Each USER_EXIT= entry must be followed by both a file name and file type. The file type must be either EXEC or MODULE.

If multiple occurrences of this keyword are encountered in the CONFIG* DATADVH file, only the first occurrence will be used.

Note: For more information on the various exit points, see the Chapter 9, “Exit Routines” on page 141.

The PW_MIN_LENGTH= Entry

The PW or TESTPW command eventually gets routed into the PASSWORD_SYNTAX_CHECKING_USER_EXIT routine. Your site may have rules prohibiting use of *trivial* passwords. These rules are enforced by this exit routine. One of the most common rules is to prohibit short passwords. The IBM-supplied sample exit routine looks in the CONFIG* DATADVH file to determine what your site considers to be a *short* versus a *long* password. The IBM-supplied default is:

```
PW_MIN_LENGTH= 3
```

The PW_MIN_LENGTH= entry must be followed by an integer value between 1 and 8 inclusive.

If multiple occurrences of this keyword are encountered in the CONFIG* DATADVH file, only the first occurrence will be used.

Note: This entry is also used by the PASSWORD_SYNTAX_CHECKING_EXIT routine running in the DirMaint service machine. Frequently, these two exit descriptors will point to the same physical routine.

The FROM= DEST= Entries

The FROM= DEST= entries aid in a multiple system cluster environment when it is necessary to route a command or file from the system where it is entered by the user to the system where the DIRMAINT service machine is running, or to route messages or files from the DIRMAINT service machine back to the user. In many cases, systems in a multiple system cluster may be using the Cross System Extensions (CSE) shared spool file support. In other cases, systems in a multiple system cluster may not be using shared spool file, but will be communicating with each other through a private *spool file bridge* network. And in other cases, systems in a multiple system cluster may share a common source directory, but communicate with each other only through the enterprise-wide RSCS network. Entries in the CONFIG* DATADVH file identify how to accomplish this routing in each situation. The IBM-supplied default is:

```
FROM= *  DEST= *  S= *  T= *
```

This uses the RSCS networking between systems.

If you are using shared spool files between your systems, you should change the *S= * to S= DIRMAINT*, or whatever the user ID of your DIRMAINT service machine happens to be.

If you are using a dedicated *spool file bridge* network, you will need to identify your network configuration to DirMaint. You may need to completely specify each entry, which may need up to 256 entries if there are no similarities between them. For example, our DirMaint Development test system still uses the spool file bridge technique on our test system, with the network structure set up for DirMaint Release 4.0 on a VM/SP HPO Release 5 system and a pair of VM/XA™ SP Release 2.1 systems.

```
FROM= DVHTEST1  DEST= DVHTEST2  S= DVHTEST2  T= DVHTEST1
FROM= DVHTEST1  DEST= DVHTEST3  S= DVHTEST3  T= DVHTEST1
FROM= DVHTEST2  DEST= DVHTEST1  S= DVHTEST1  T= DVHTEST2
FROM= DVHTEST2  DEST= DVHTEST3  S= DVHTEST3  T= DVHTEST2
FROM= DVHTEST3  DEST= DVHTEST1  S= DVHTEST1  T= DVHTEST3
FROM= DVHTEST3  DEST= DVHTEST2  S= DVHTEST2  T= DVHTEST3
```

Although this technique still works between our current VM/ESA systems, a little rational restructuring of the network makes the entries in the CONFIG* DATADVH file a lot easier:

```
FROM= *         DEST= DVHTEST1  S= SFBRIDGE  T= *
FROM= *         DEST= DVHTEST2  S= SFBRIDGE  T= *
FROM= *         DEST= DVHTEST3  S= SFBRIDGE  T= *
```

For a maximum of 16 entries or for a single entry:

```
FROM= *         DEST= *         S= SFBRIDGE  T= *
```

DirMaint Release 4 was limited to routing commands, and results within a fairly small cluster. However, DirMaint Release 5.0 supports enterprise-wide networking. This allows an administrator on the corporate headquarters system in California, USA, to make directory changes to a system in Europe without having to log on to the system in Europe. (Of course, such remote administration must have been previously authorized by the administrators of the European system.) This requires using the RSCS network between the systems. And it requires entries in the CONFIG* DATADVH file to describe how this routing is done. The IBM-supplied default is:

```
FROM= *  DEST= *  S= *  T= *  U= DIRMAINT
```

This presumes the user ID of the network service machine is found using the system IDENTIFY command, and the tag node is the same as the destination name, which in turn is the same as the TOSYS value specified on the DIRM command. If you use a *nickname* for the TOSYS value, the nickname must be defined on a DEST= tag for the system from which the command is coming, and the correct network node ID must be specified on the corresponding T= tag. If the user ID of the DIRMAINT service machine on the destination system is not DIRMAINT, the correct user ID must be specified on the U= tag.

Example—Using the IDENTIFY Command

```
FROM= *  DEST= HQ  S= *  T= CORPHQ  U= DIRMR5
```

The order that multiple occurrences of these entries are encountered in the CONFIG* DATADVH file is very significant, and somewhat different from the ordering processing of the National Language entries. When looking for a language related entry, the first occurrence of the language specific entry is used regardless of its position relative to the ..... entry. The first ..... entry is used only if no relevant entry is found for the specific language in question. When looking for a cluster or network routing entry, all entries are considered in the order that they occur. Thus, the command may be processed using a ***** entry, even though there is a specific entry for the FROM= and DEST= nodes in the file, if the ***** entry appears first.

Chapter 8. Delegating Administrative Authority

This chapter provides guidance for delegating administrative authority and altering command classes on your system. This is necessary if you want to allow users to enter commands other than class G or General user commands.

Command Classes

DirMaint employs a command class structure similar to the CP command privilege structure. A specific command can be placed in one or more command classes and a specific user can be authorized to enter one or more command classes.

DirMaint has several layers of authorization to go through when a command is entered. Some of the logic takes place on the users machine before sending the transaction to the DirMaint server. The remainder of the logic takes place on the DirMaint server.

When a command is entered, DIRMAINT determines if the user:

- Entering the command is authorized for the command set required to enter the command.
- Is authorized to enter that level command against the target user.

Command classes are represented by alphanumeric characters.

With DirMaint Release 5.0 you can have up to 36 tailorable DirMaint command sets. A user ID cannot enter commands in a command set unless authorization is given. However, user IDs can be authorized to act on behalf of other user IDs, and may be authorized to use the command sets of the user ID they are acting on behalf of. This allows for delegation of administrative authority.

Example—Administrative Authority: Administrative authority could be delegated to class instructors over student user IDs, or to department supervisors over user IDs within their department.

By default, DirMaint Release 5.0 provides the following nine command sets at installation:

Table 12 (Page 1 of 2). Privilege Classes

Class	User and Function
A	Administration, non-DASD related
D	DASD Management
G	General users
H	Helpdesk
M	Password Monitor
O	System Operator

Table 12 (Page 2 of 2). Privilege Classes

Class	User and Function
P	DASD management automated Programs, such as DFSMS/VM®
S	Support programmer
Z	Internal communication

Notes:

1. Additional command classes can be defined if specific command subsets are required.

Command Classes on the DIRMAINT Server

When the transaction is received on the DirMaint server, the command class of the command and the issuing user is checked against the AUTHFOR CONTROL file to ensure they have authority to issue the command. In addition to using the class of the command to authorize the issuing user, the target of the operation is also consulted to ensure they have authorized the user issuing the command to use the specified command class against their user directory entry.

Defining a New Command Class

The shipped command classes may be inappropriate for your local installation. DIRMAINT allows administrators to define a custom set of commands and assign them to a locally defined command class. This command class is then treated as a shipped command class by the DIRMAINT installation.

Custom command classes are particularly useful for allowing users to use a few commands from a potentially dangerous class of commands. You may want to permit a POSIX administrator to have authority to assign specific UID values on your system. The DIRMAINT POSIXINFO command is required to perform this. The POSIXINFO command is shipped with a command class A but several other administrative commands also share the privilege class A.

One alternative is to grant your POSIX administrator class privilege class A and trust that they will only use the POSIXINFO command. A better solution is to define a new command class and place the POSIXINFO command in this new class. Then authorize the POSIX administrator to use the new class.

Command classes are established in the 150CMDS DATADVH and 140CMDS DATADVH files. Each file corresponds to the command level being used. As POSIXINFO only exists in the 150A command set. The specific format of the file is described in the prologue section of the file.

Example—150CMDS DATADVH File:

```
⋮
1 POOL         DVHXMIT   DVHPOOL   Y  2 A........
  POSIXFSROOT  DVHXMIT   DVHPOSIX  Y    ..G......
  POSIXGLIST   DVHXMIT   DVHGLIST  Y    A........
  POSIXGROUP   DVHXMIT   DVHGBGRP  Y    A........
  POSIXINFO    DVHXMIT   DVHPOSIX  Y    A........
  POSIXIUPGM   DVHXMIT   DVHPOSIX  Y    ..G......
  POSIXIWDIR   DVHXMIT   DVHPOSIX  Y    ..G......
  POSIXOPT     DVHXMIT   DVHPXOPT  Y    A........
  PRIORITY     DVHXMIT   DVHPRI    Y    A........
  PRIOSET      DVHXMIT   DVHPRI    Y    A........
⋮
```

1 Specifies the command name,

2 Specifies the command classes associated with the command. As you can see, POSIXINFO is considered a class A command. To define a new class simply place an alphanumeric character adjacent to the existing command classes. You should choose a class that is not being used as an IBM default. You should also note that the class field must be a single contiguous, with no imbedded blanks. The position you place the new class is unimportant but placing it on the end is recommended.

As all IBM-supplied defaults are alphabetic, the following example will use command class 5 to ensure that no conflicts arise. After altering the file, the results are:

```
⋮
POOL         DVHXMIT   DVHPOOL   Y    A........
POSIXFSROOT  DVHXMIT   DVHPOSIX  Y    ..G......
POSIXGLIST   DVHXMIT   DVHGLIST  Y    A........
POSIXGROUP   DVHXMIT   DVHGBGRP  Y    A........
POSIXINFO    DVHXMIT   DVHPOSIX  Y  3 A........
POSIXIUPGM   DVHXMIT   DVHPOSIX  Y    ..G......
POSIXIWDIR   DVHXMIT   DVHPOSIX  Y    ..G......
POSIXOPT     DVHXMIT   DVHPXOPT  Y    A........
PRIORITY     DVHXMIT   DVHPRI    Y    A........
PRIOSET      DVHXMIT   DVHPRI    Y    A........
⋮
```

3 Indicates you can now authorize your administrator to use privilege class 5 and they will only have authority to issue the POSIXINFO command.

DirMaint Server Authorization Procedures

The DIRMAINT server uses the AUTHFOR CONTROL file as a repository of authorization information. This file contains a listing of user IDs who are authorized to act for other user IDs and the privilege classes that have been delegated to them.

AUTHFOR CONTROL File

The AUTHFOR CONTROL file resides on the DirMaint 1DF disk. The format is:

```
tUid iUid iNode CmdLevel CmdSets
```

Where:

tUid
: Identifies the *userid* or *profileid* granting permission for a set of command classes to be used against them. A keyword of *ALL* may be used here to indicate that the following user has authority to use the specified privileges against all users.

iUid
: Identifies the *userid* that is authorized to use the command sets.

iNode
: Identifies the network *nodeid* that the issuing user is on. This allows *userids* on different nodes to be granted different command classes.

CmdLevel
: Specifies the command level that authority is granted for. Valid values are 140A and 150A.

CmdSets
: Specifies a list of command classes being authorized. The list cannot have any spaces imbedded between the classes.

  Note: The AUTHFOR CONTROL file can be maintained in one of two ways

  - Directly through the DIRM AUTHFOR command.
  - Manually by using DIRM SEND to retrieve the file. Using XEDIT to alter the file and returning the file through the DIRM FILE command.

    If the AUTHFOR CONTROL file is included in the loadable files list in the CONFIG DATADVH file, then a DIRM RLDCODE command is needed after the DIRM FILE command to make the change known to DIRMAINT. Otherwise, the change will not be noticed until the next time the DIRMAINT machine is re-IPLed.

Example—Granting a User Class A Authority: In this example, we will be granting user ID HOWLANDM at GDLVM7 authority to issue class A for all user IDs in the source directory. This authority will only be granted for command level 105A. The following steps should be done:

1. Issue the following command from the DirMaint console:

   ```
   FOR ALL AUTHFOR HOWLANDM FROM GDLVM7 CMDLEVEL 150A CMDSET A
   ```

   or, add the following line to the AUTHFOR CONTROL file using XEDIT:

   ```
   ALL      HOWLANDM GDLVM7   150A A
   ```

2	Contact user ID HOWLANDM and request that the following command be issued from their machine:
	`DIRM DEFAULTS CMDSETS GA`
	Note: This command should contain all command classes assigned to user ID HOWLANDM because this replaces the existing command set. So, if user ID HOWLANDM already had classes GDS you should adjust the command to include the original classes and the new class. The order of the command classes is not important. Enter:
	`DIRM DEFAULTS CMDSETS GADS`

Example—Revoking a User Class Authority: To revoke the authority you just gave user ID HOWLANDM at GDLVM7, the following steps should be done:

Table 13. Revoking a User Class A Authority

1	Issue the following command from the DirMaint console:
	`FOR ALL DROPFOR HOWLANDM FROM GDLVM7 CMDLEVEL 150A CMDSET A`
	or, delete the following line from the AUTHFOR CONTROL file using XEDIT:
	`ALL HOWLANDM GDLVM7 150A A`
	Note: If user ID HOWLANDM had additional classes (besides class A) you may choose to alter the line instead of deleting it. To let the user retain the *other* classes simply remove the *A* from the command set list and let the other classes remain.

2	Contact user ID HOWLANDM and request that the following command be issued from their machine:
	`DIRM DEFAULTS CMDSETS G`
	Note: If user ID HOWLANDM had additional classes (besides class A) that are not to be revoked you should include these classes in the list (along with class G) as shown below:
	`DIRM DEFAULT CMDSETS GDS`

Chapter 9. Exit Routines

This chapter describes each of the exits available with DirMaint Release 5.0. An exit is a point in a program that is designed to allow an exit routine to gain control of certain processes. Some exit routines are required, but most are optional. IBM supplies samples of all of the required DirMaint exit routines, as well as samples for some of the optional exits. Where appropriate, the supplied DirMaint exit routines are enabled for interfacing between DirMaint and other programs, such as RACF, OV/VM, or products that facilitate distributed processing.

Most DirMaint exit routines are written in the REXX programming language, and can be customized or replaced by an installation-written exit, or be called by an installation-written command. Only the IBM-supplied sample exit routines may be used in a TCB environment without the approval of your Designated Approval Authority (DAA). Any modifications to the IBM-supplied sample exit routines in a TCB environment require the approval of the DAA.

Release 4.0 Exit Support Replacements

The DirMaint Release 4.0 exits have been replaced with new exits that perform comparable function. Where appropriate, the replacement exits have been enhanced for system affinity support. Table 14 lists each Release 4.0 exit and the Release 5.0 replacement.

Table 14. Summary of Replacement Exit Routines

DirMaint Release 4.0 Exit	DirMaint Release 5.0 Exit
DVHUA1	DVHXRB
DVHUA2	DVHXRA
DVHACCT	DVHXAV
DVHACCTM	DVHXAV
DVHPWX	DVHPXV
DVHMPW	DVHXMP
DVHLOCAL	DVHXPROF (See note, below.)
DVHDATLC	DVHXPROF (See note, below.)
DVHSATLC	DVHXPROF (See note, below.)

Note: The IBM-supplied DVHXPROF exit routine is called by each of the service machines: DIRMAINT, DATAMOVE, and DIRMSAT. The name of this exit routine is not tailorable. If this exit routine is replaced by an installation written routine, it must be a REXX exec file, and it must be named DVHXPROF. This replaces:

- DVHLOCAL for DIRMAINT
- DVHDATLC for DATAMOVE
- DVHSATLC for any DIRMSATs.

Command and Exit Routine Interactions

When a DirMaint command is entered, it may interact with multiple exit routines within the various virtual machines it executes in. The number of exit routines a command interacts with is dependent upon the command being entered and any installation tailoring that was done.

User Virtual Machine

When a command is entered, the command may cause an interaction with all, or a subset of, the following exit routines:

- DVHCXC
- DVHCXB
- DVHPXR
- DVHPXV

- DVHPXA
- DVHCXV

The exit routine interactions occur in the order of the above list.

DATAMOVE Service Machine

When a command is entered, the command may cause an interaction with all, or a subset of, the following exit routines:

- DVHXRC
- DVHXRB
- DVHDXP
- DVHXRA

The exit routine interactions occur in the order of the above list.

DIRMSAT Service Machine

When a command is entered, the command may cause an interaction with all, or a subset of, the following exit routines:

- DVHXRC
- DVHXRB
- DVHXRA

The exit routine interactions occur in the order of the above list.

DIRMAINT Service Machine

When a command is entered, the command may cause an interaction with multiple exit routines being called within the DIRMAINT service machine. Authorization checking exits will be called before notification exits. The following exits will be called for each command:

ACCOUNT	DVHXAV, DVHXAN
MDISK	DVHXMP, DVHXMN
PW	DVHPXV, DVHXPN
TESTPW	DVHPXV
ADD	DVHPXV, DVHXAV, DVHXDA - for each disk, DVHXMP - for each disk, DVHXLA - for each link, DVHXUN, DVHXPN, DVHXAN, DVHXDN - for each disk, DVHXMN - for each disk, and DVHXLN - for each link
AMDISK	DVHXDA, DVHXMP, DVHXDN, DVHXMN
BACKUP	DVHXTP
CHNGID	DVHPXV, DVHXAV, DVHXMP - for each disk DVHXUN - new, DVHXPN - new, DVHXAN - new, DVHXDN - for each new disk, DVHXMN - for each new disk, DVHXLN - for each new link, DVHXMN -for each old disk, DVHXDN -for each old disk, DVHXLN - for each old link, DVHXAN - old, DVHXPN - old, DVHXUN - old
CHVADDR	DVHXMP, DVHXMN - old, DVHXDN - twice, DVHXMN - new. Or, for a changed LINK: DVHXLN - twice
CMDISK	DVHXDA, DVHXDN

DMDISK	DVHXMN, DVHXDN
PURGE	DVHXMN - for each disk, DVHXDN - for each disk, DVHXLN - for each old link, DVHXAN, DVHXPN, DVHXUN.
PWGEN	DVHPXR, DVHPXV
PWMON	DVHXCP, DVHXPP
REPLACE	DVHPXV, DVHXAV, DVHXPN, DVHXAN, DVHXMN - for each deleted disk, DVHXDN - for each deleted disk, DVHXLN - for each deleted link, DVHXDN - for each added disk, DVHXMN - for each remaining disk, DVHXLN - for each new link.
RMDISK	DVHXDA, DVHXDN
SETACNT	DVHXAV, DVHXAN
SETPW	DVHPXV, DVHXPN
SETSTAG	DVHXTA
TMDISK	DVHXFA - new, DVHXMP - new, DVHXDN - new, DVHXMN - new, DVHXMN - old, DVHXDN - old
STAG	DVHXTA

Exit Routines Summary

Table 15 summarizes the DirMaint Release 5.0 exit routines. You can find more information about each exit routine by referring to the referenced page.

Table 15 (Page 1 of 2). Exit Routines Summary

Exit Routine	Function	Environment	IBM Supplied Sample	Page
DVHCXA	Command exit, after processing	User Machine	Yes	148
DVHCXB	Command exit, after parsing, before processing	User Machine	No	149
DVHCXC	Command exit, before parsing	User Machine	Yes	151
DVHDA0	External security manager password authentication exit	DIRMAINT Machine		152
DVHDXP	DATAMOVE non-CMS disk copying exit	DATAMOVE Machine	No	153
DVHESMLR	External Security Manager log recording exit	User Machine DIRMAINT Machine DATAMOVE Machine DIRMSAT Machine	Yes	154
DVHPXA	User's logon password exit, after transmission to DIRMAINT	User Machine	Yes	156
DVHPXR	Random password generation exit for logon	User Machine DIRMAINT Machine	Yes	157 161
DVHPXV	User's logon password exit, syntax verification	User Machine DIRMAINT Machine	Yes	159
DVHXAN	Account number notification exit	DIRMAINT Machine	No	163
DVHXAV	Account number verification exit	DIRMAINT Machine	Yes	164
DVHXCP	Check user privilege exit	DIRMAINT Machine	No	165
DVHXDA	DASD authorization checking exit	DIRMAINT Machine	No	166
DVHXDN	DASD notification exit	DIRMAINT Machine	No	168
DVHXFA	FOR authorization checking exit	DIRMAINT Machine DATAMOVE Machine DIRMSAT Machine	No	169
DVHXLA	Link authorization checking exit	DIRMAINT Machine	No	171
DVHXLF	Log record filtering exit	DIRMAINT Machine DATAMOVE Machine DIRMSAT Machine	Yes	172
DVHXLN	Link notification exit	DIRMAINT Machine	No	174
DVHXMN	Minidisk password change notification exit	DIRMAINT Machine	Yes	175

Table 15 (Page 2 of 2). Exit Routines Summary

Exit Routine	Function	Environment	IBM Supplied Sample	Page
DVHXMP	Minidisk password syntax verification exit	DIRMAINT Machine	No	176
DVHXMU	MULTIUSER authorization checking exit	DIRMAINT Machine DATAMOVE Machine DIRMSAT Machine	Yes	177
DVHXPN	Password change notification exit	DIRMAINT Machine	No	178
DVHXPROF	Post-profile exit for the DirMaint service machines. Typically a router to call DVHLOCAL for DIRMAINT, DVHDATLC for DATAMOVE, or DVHSATLC for DIRMSAT. The exit name, DVHXPROF, must **not** be renamed.	DIRMAINT Machine	Yes, see Figure 34 on page 211	211
DVHXPP	Password notice print exit	DIRMAINT Machine	Yes	179
DVHXRA	Request after processing exit	DIRMAINT Machine DATAMOVE Machine DIRMSAT Machine	No	180
DVHXRB	Request after parsing, before processing exit	DIRMAINT Machine DATAMOVE Machine DIRMSAT Machine	No	182
DVHXRC	Request before parsing exit	DIRMAINT Machine DATAMOVE Machine DIRMSAT Machine	No	184
DVHXTA	Local STAG authorization exit	DIRMAINT Machine	No	185
DVHXTP	Backup tape mount exit	DIRMAINT Machine	Yes	186
DVHXUN	User ID change notification exit	DIRMAINT Machine	No	188

DirMaint Release 5.0 Exit Routine Descriptions

This section provides specific information about each IBM-supplied exit routine. The exit routine descriptions are catalogued in alphabetical order. Each exit routine description is presented in the following format:

- *Environment:* Indicates where the exit routine is called.
- *Description:* Explains what the exit routine does.
- *Invocation:* Displays the entry to be placed in the CONFIG* DATADVH file.
- *Interface Parameter:* Identifies the parameters that are to be provided when the exit routine is called.
- *Return Codes:* The return codes that the exit routine can return (if any), and their meaning.

Command After Processing (DVHCXA)

Environment
User virtual machine

Description
New for Release 5.0

Command exit, after processing.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file:

```
COMMAND_AFTER_PROCESSING_USER_EXIT= DVHCXA EXEC
```

For more information, see “The CONFIG* DATADVH File” on page 120.

Interface Parameter
DVHCXA

Is called with the following two parameters:

- Return code from processing the command
- Command name

In addition, the following two interface variables are available:

CMD_STRING
The command string as verified and returned by the parser. Command and parameter abbreviations HAVE been resolved. All prefix variables (TOSYS, ASUSER, BYUSER, FORUSER, and ATNODE) have been stripped off and stored in their own separate global variables. They will each have a value, if not specified on the user's command then defaults will be supplied.

LOG_STRING
The command string as verified and returned by the parser, with passwords and other sensitive information *masked* for security. The prefix operands are included as entered by the user, defaults are NOT filled in for omitted parameters.

These two interface variables are obtainable using: 'PIPE VAR variable_name 2 | ...'. DVHCXB is called indirectly through the DVHCEXIT EXEC. An *invocation* or *generation* number of 2 is necessary to get the value from DVHCEXIT's caller, DVHCMD.

Return Codes
Ignored upon exit.

Command Before Processing (DVHCXB)

Environment
User virtual machine

Description
New for Release 5.0

Command exit, after parsing, before processing.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
COMMAND_BEFORE_PROCESSING_USER_EXIT= DVHCXB EXEC
```

For more information, see “The CONFIG* DATADVH File” on page 120.

Interface Parameter
DVHCXB

Is called with no parameters, but with two interface variables set:

CMD_STRING

The command string as verified and returned by the parser. Command and parameter abbreviations HAVE been resolved. All prefix variables (TOSYS, ASUSER, BYUSER, FORUSER, and ATNODE) have been stripped off and stored in their own separate global variables. They will each have a value, if not specified on the user's command then defaults will be supplied.

LOG_STRING

The command string as verified and returned by the parser, with passwords and other sensitive information *masked* for security. The prefix operands are included as entered by the user, defaults are NOT filled in for omitted parameters.

These two interface variables are obtainable using: 'PIPE VAR variable_name 2 | ...'. DVHCXB is called indirectly through the DVHCEXIT EXEC. An *invocation* or *generation* number of 2 is necessary to get the value from DVHCEXIT's caller, DVHCMD.

Return Codes

This routine must exit with one of the following:

Table 16. DVHCXB Return Codes

Return Code	Meaning
30 - Nop	Regular DirMaint processing continues.
31	A replacement command string has been set into variable CMD_STRING using 'PIPE ... | VAR CMD_STRING 2'. DVHCXB is called indirectly through the DVHCEXIT EXEC. An *invocation* or *generation* number of 2 is necessary to pass the value back to DVHCEXITs caller, DVHCMD. If the CMD_STRING is changed and contains passwords or other sensitive information, the LOG_STRING should also be changed to the equivalent string with the sensitive information changed to a string of *XXXs*.
Other	The command has been completely processed. DirMaint exits, passing back whatever return code it was given.

Command Before Parsing (DVHCXC)

Environment
User virtual machine

Description
New for Release 5.0

Command exit, before parsing.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
COMMAND_BEFORE_PARSING_USER_EXIT= DVHCXC EXEC
```

For more information, see “The CONFIG* DATADVH File” on page 120.

Interface Parameter
DVHCXC

Is called with the following parameter:

- The command string, as entered by the user; command and parameter abbreviations have NOT been resolved.

Return Codes
This routine must exit with one of the following:

Table 17. DVHCXC Return Codes

Return Code	Meaning
30 - Nop	Regular DirMaint processing continues.
31	A replacement command string has been set into variable CMD_STRING using 'PIPE ... | VAR CMD_STRING 2'. DVHCXC is called indirectly through the DVHCEXIT EXEC. An *invocation* or *generation* number of 2 is necessary to pass the value back to DVHCEXITs caller, DVHCMD.
Other	The command has been completely processed. DirMaint exits, passing back whatever return code it was given.

ESM Password Authentication (DVHDA0 MODULE)

Environment
DIRMAINT service machine

Description
New for Release 5.0

External security manager password authentication exit; with an ESM installed, the administrator has the choice of using the user's logon password owned by the ESM for command authentication instead of using the password from the user's VM directory entry. This exit calls the ESM to provide this service.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
ESM_PASSWORD_AUTHENTICATION_EXIT= DVHDA0 MODULE
```

Note: The DIRMAINT machine may abend, hang, or reject transactions if the ESM is not installed, DIRMAINT has not been granted the authority to use the authentication service, or if the ESM is temporarily inactive. For DIRMAINT to revert to verification using the directory passwords, the name of the exit routine must be removed from the `ESM PASSWORD AUTHENTICATION EXIT` statement in the CONFIG* DATADVH file(s), and issue a RLDDATA command to reset DirMaint's global variables. Then you need to restore use of the exit routine when the ESM has been reactivated.

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHDA0

Is called with the following parameters:

- The user ID whose password is to be authenticated
- The password to be authenticated.

Return Codes
This routine must exit with one of the following:

Table 18. DVHDA0 Return Codes

Return Code	Meaning
0	The password and user ID are valid.
30	Reserved for IBM use
Other nonzero	The password and user ID are not valid.

DATAMOVE non-CMS Copying (DVHDXP)

Environment
DATAMOVE service machine

Description
DATAMOVE non-CMS disk copying exit.

Invocation
The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.:

```
DATAMOVE_NONCMS_COPYING_EXIT= DVHDXP EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHDXP

Is called with the following parameters:

- The source virtual address of the DATAMOVE machine
- The destination virtual address of the DATAMOVE machine
- The target ID
- The target system affinity, usually *.

Return Codes
This routine must exit with one of the following:

Table 19. DVHDXP Return Codes

Return Code	Meaning
0 or 30 - Nop	The minidisk appears to be a standard CMS minidisk. DATAMOVE continues by making the same checks as if the exit were not present. DATAMOVE will link to both disks SW if possible (ESA feature), otherwise just W (370 feature). If present, the DFSMS MODULE will be used to perform the copy; otherwise, FORMAT and COPYFILE will be used.
0*xx*	The minidisk copy has been completed. No further processing is done to complete the copy. Cleanup processing, if active, will be done separately. A different return code may be used for each type of nonstandard CMS or non-CMS disk encountered and supported.
1*xx*	The minidisk copy has not been completed because the CP LINK command failed with RC=1*xx*. The copy will be retried later.
2*xx*	The minidisk copy has not been completed because the CP LINK command failed with RC=2*xx* The copy will be retried later.
Other nonzero	The minidisk copy has not been completed. Most likely the minidisk format is not supported by the exit routine. The exit routine has already issued the appropriate messages. The return code should be the same as the message number. The copy will not be retried.

ESM Log Recording (DVHESMLR)

Environment

DIRMAINT service machine
DATAMOVE service machine
DIRMSAT service machines

Description

New for Release 5.0

External security manager log recording exit. By default, all command and message activity is recording in the service machine's console file only. Optionally, this information (subject to filtering) may be communicated to an ESM, such as RACF for recording is a secure audit file by this exit routine.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
ESM_LOG_RECORDING_EXIT= DVHESMLR  EXEC
```

Note: The DirMaint server may abend, hang, or shutdown if the ESM is not installed, the DirMaint server has not been granted the authority to use the log recording service, or if the ESM is temporarily inactive. For the DirMaint server to operate under these conditions, the name of the exit routine must be removed from the `ESM LOG RECORDING EXIT` statement in the CONFIG* DATADVH file(s), and issue a RLDDATA command to reset the server's global variables. Then you need to restore use of the exit routine when the ESM has been reactivated.

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHESMLR

Is called with the following parameters:

- A date stamp (*yyyymmdd*) and time stamp (*hh:mm:ss*).
- The node ID and user ID for whom the message is being recorded. For a message to a distribution list, the node ID will be recorded as ..... and the user ID will be the nickname for the list.
- The message identifier is DVH*rrrnnnn*S

  Where:

  rrr
  : Specifies the routine issuing the message.

  nnnn
  : Specifies the message number.

  S
  : Specifies the message severity.

- The message text to be logged.

Return Codes

The return code is ignored upon exit.

Password After Processing (DVHPXA)

Environment
User virtual machine

Description
New for Release 5.0

User's logon password exit, after transmission to DIRMAINT.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
PASSWORD_NOTIFICATION_USER_EXIT= DVHPXA EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHPXA

Is called with the following parameters:

- The keyword USER (if invoked in the user's virtual machine) or DIRMAINT (if invoked by the DIRMAINT virtual machine).
- The command causing this notification: PW or TESTPW.
- The target ID class (always USER).
- The target ID.
- The target system affinity (always *).
- The new password.

Return Codes
The return code is ignored upon exit.

Note: The password transaction has been successfully sent to the DIRMAINT service machine, but a change has not necessarily taken effect.

Password Random Generator (DVHPXR)

Environment

User virtual machine
DIRMAINT virtual machine

Description

New for Release 5.0

Random password generation exit for logon.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

For the user's virtual machine:

```
PASSWORD_RANDOM_GENERATOR_USER_EXIT= DVHPXR EXEC
```

For more information, see “The CONFIG* DATADVH File” on page 120.

For the DIRMAINT service machine:

```
PASSWORD_RANDOM_GENERATOR_EXIT= DVHPXR EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHPXR

Is called with the following parameters:

- The keyword USER.
- The command name keyword (PW, TESTPW, or PWGEN).
- The target ID class (always USER).
- The target ID.
- The current password (or * if unknown).
- The target system affinity (always *).
- The keyword RANDOM.
- The algorithm to be used (ALPHA, NUM, ALPHANUM, and so forth), or null.
- Optional parameters determined by the specific algorithm; such as length, minimum and maximum lengths, and so forth.

Return Codes

This routine must exit with one of the following:

Table 20. DVHPXR Return Codes

Return Code	Meaning
0	The password has been generated and pushed onto the stack.
nonzero	The password could not be generated and the stack is unchanged.

Password Syntax Checking (DVHPXV)

Environment

User virtual machine
DIRMAINT virtual machine

Description

New for Release 5.0

User's logon password exit, syntax verification.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

For the user's virtual machine:

```
PASSWORD_SYNTAX_CHECKING_USER_EXIT= DVHPXV EXEC
```

For more information, see “The CONFIG* DATADVH File” on page 120.

For the DIRMAINT service machine:

```
PASSWORD_SYNTAX_CHECKING_EXIT= DVHPXV EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHPXV

Is called with the following parameters:

- The keyword USER (if invoked in the user's virtual machine) or DIRMAINT (if invoked by the DIRMAINT virtual machine).
- The keyword PW or TESTPW (if invoked in the user's virtual machine); -or- ADD, CHNGID, PW, SETPW, or TESTPW (if invoked by DIRMAINT).
- The target ID class (always USER).
- The target ID.
- The target system affinity (always *).
- The current password (or * if unknown).
- The proposed new password.
- An optional NOMSG keyword.

Note: The tracing messages should be issued regardless of the NOMSG keyword, as may other messages of a serious nature. But messages reflecting the reason for rejecting a particular proposed password should not be issued if the NOMSG keyword is specified, although it should exit with a return code equal to the message number that would have been issued if the NOMSG keyword had been omitted.

Return Codes

This routine must exit with one of the following:

Table 21. DVHPXV Return Codes

Return Code	Meaning
0	The password is accepted. No further checking is done.
30 - Nop	The password is neither accepted nor rejected. DirMaint continues by making the same checks as if the exit were not present.
31	The password is rejected. DirMaint should issue a generic error message.
Other nonzero	The password is rejected and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

Random Password Generator (DVHPXR)

Environment

DIRMAINT service machine
User virtual machine

Description

New for Release 5.0

Random password generation exit allows the installation to customize the format of passwords generated by PWGEN. The IBM-supplied default is ALPHANUMeric, with the length specified by the issuer of the PWMON command.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

For the DIRMAINT service machine:

```
PASSWORD_RANDOM_GENERATOR_EXIT= DVHPXR EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

For the user's virtual machine.

```
PASSWORD_RANDOM_GENERATOR_USER_EXIT= DVHPXR EXEC
```

For more information, see “The CONFIG* DATADVH File” on page 120.

Interface Parameter

DVHPXR

Is called with the following parameters:

- The keyword DIRMAINT.
- The command name keyword (PW, TESTPW, or PWGEN).
- The target ID class (always USER).
- The target ID.
- The target system affinity (always *).
- The current password (or * if unknown).
- The keyword RANDOM.
- The algorithm to be used (ALPHA, NUM, ALPHANUM, and so forth), or null.
- Optional parameters determined by the specific algorithm; such as length, minimum and maximum lengths, and so forth.

Return Codes

This routine must exit with one of the following:

Table 22. DVHPXR Return Codes

Return Code	Meaning
0	The password has been generated and pushed on the stack.
nonzero	The password could not be generated and the stack is unchanged.

ACCOUNT Number Notification (DVHXAN)

Environment
DIRMAINT service machine

Description
New for Release 5.0

Account number notification exit. This exit may be used to notify other service machines of changes to a user's account number. It will be called whenever an account number change is successful.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
ACCOUNT_NUMBER_NOTIFICATION_EXIT= DVHXAN EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXAN

Is called with the following parameters:

- The command causing this notification: ACCOUNT, ADD, ADD-ACCT, CHNGID-*NEW, CHNGID-*OLD, PURGE, REPLACE-NEW, REPLACE-*NEW or SETACNT.
- The target ID class (PROFILE or USER).
- The target ID.
- The target system affinity, usually *.
- The new or old account number.

  Note: If null, the user ID will be substituted. All primary and secondary account numbers from the ACCOUNT statement are included on one call, and multiple tertiary account numbers from any *AC= records are included on one call.

Return Codes
Any return code is ignored upon exit.

ACCOUNT Number Verification (DVHXAV)

Environment

DIRMAINT service machine

Description

Replaces DVHACCT and DVHACCTM

Account number verification exit. The passed parameters are enhanced for networking support. The exit will not only be called for the ACCOUNT command; but also for ACNTADD, ACNTDEL, ADD, ADD-*AC, ADD-ACCT, CHNGID, CHNGID-*AC, SETACNT, SETACNT-DELETE, and SETACNT-*DELETE.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
ACCOUNT_NUMBER_VERIFICATION_EXIT= DVHXAV EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHXAV

Is called with the following parameters:

- The command causing this check: ACCOUNT, ADD, CHNGID, or SETACNT; or ACNTADD or ACNTDEL.
- The target ID class (PROFILE or USER).
- The target ID.
- The target system affinity, usually *.
- The proposed account number. (If null, the user ID will be substituted.)

Return Codes

This routine must exit with one of the following:

Table 23. DVHXAV Return Codes

Return Code	Meaning
0	The account number is accepted. No further checking is done.
30 - Nop	The account number is neither accepted nor rejected. DirMaint continues by making the same checks as if the exit were not present.
31	The account number is rejected. DirMaint should issue a generic error message.
Other nonzero	The account number is rejected and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

Check User Privilege (DVHXCP)

Environment
DIRMAINT service machine

Description
New for Release 5.0

Check user privilege exit. This exit may determine whether a given user ID is *privileged*. This determines which password change interval rule applies.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
CHECK_USER_PRIVILEGE_EXIT= DVHXCP EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXCP

Is called with the following parameters:

- The commands making this check: ADD, CHNGID, PW, PWMON, PW?, SETPW
- The target ID class (USER)
- The target ID
- The target system affinity, usually *.

Return Codes
This routine must exit with one of the following:

Table 24. DVHXCP Return Codes

Return Code	Meaning
0	The user is a GENERAL user. Use the PW_INTERVAL_FOR_GEN rules.
1	The user is a PRIVILEGED user. Use the PW_INTERVAL_FOR_PRIV rules.
30	Reserved for exit routine not found. Use the PW_INTERVAL_FOR_GEN rules.
Other nonzero	An error has occurred and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

DASD Authorization Checking (DVHXDA)

Environment

DIRMAINT service machine

Description

New for Release 5.0

The DASD authorization checking exit routine determines whether the originator is authorized to allocate space for the target user ID on the requested DASD volume. This supports distributed (departmental) administration and centralized (networking) administration. This exit will be called for all AMDISK, CMDISK, RMDISK, and ADD commands, because ADD generates AMDISK requests indirectly.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
DASD_AUTHORIZATION_CHECKING_EXIT= DVHXDA EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHXDA

Is called with the following parameters:

- The user ID of the user making this allocation
- The node ID of the user making this allocation
- The command causing this check: ADD (Since ADD generates AMDISK requests indirectly), AMDISK, CMDISK, or RMDISK
- The target ID class (Always USER)
- The target ID
- The target system affinity, usually *
- The affected minidisk address
- Minidisk extent information: device type
- Minidisk extent information: start of extent (Keyword DEVNO, T-DISK, V-DISK, or starting cylinder or block number)
- Minidisk extent information: size of extent (Keyword DEVNO for DEVNO minidisks)
- Minidisk extent information: volume ID (Real address for DEVNO. Null for T-DISK or V-DISK).

Return Codes

This routine must exit with one of the following:

Table 25. DVHXDA Return Codes

Return Code	Meaning
0	The minidisk allocation is accepted. No further checking is done.
30 - Nop	The allocation is neither accepted nor rejected. DirMaint continues by making the same checks as if the exit were not present.
31	The allocation is rejected. DirMaint should issue a generic error message.
Other nonzero	The allocation is rejected and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

Note: For this particular exit routine, return codes 0 and 30 are equivalent. This exit routine may need to be sensitive to an AMDISK command where the target ID is the DATAMOVE machine.

DASD Ownership Notification (DVHXDN)

Environment
DIRMAINT service machine

Description
New for Release 5.0

DASD notification exit. This exit may be used to notify other service machines of new, deleted, or transferred minidisks. It will be called whenever the ADD, REPLACE, CHNGID, PURGE, AMDISK, DMDISK, CHVADDR, or TMDISK commands have added, deleted, or changed ownership of a minidisk.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
DASD_OWNERSHIP_NOTIFICATION_EXIT= DVHXDN EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXDN

Is called with the following parameters:

- The command causing this notification: ADD, AMDISK, CHNGID-NEW, CHNGID-OLD, CHVADDR-NEW, CHVADDR-OLD, CMDISK, DMDISK, PURGE, REPLACE-NEW, REPLACE-OLD, TMDISK-NEW, or TMDISK-OLD
- The target ID class (Always USER)
- The target ID
- The target system affinity, usually *
- The affected minidisk address
- Minidisk extent information: device type
- Minidisk extent information: start of extent (Keyword DEVNO, T-DISK, V-DISK, or starting cylinder or block number)
- Minidisk extent information: size of extent (Keyword DEVNO for DEVNO minidisks)
- Minidisk extent information: volume ID (Real address for DEVNO; Null for T-DISK or V-DISK).

Return Codes
Any return code is ignored upon exit.

Note: This exit routine may need to be sensitive to the AMDISK, DMDISK, TMDISK-NEW or TMDISK-OLD commands, where the target ID is the DATAMOVE machine.

FOR Authorization Checking (DVHXFA)

Environment

DIRMAINT service machine
DATAMOVE service machine
DIRMSAT service machines

Description

New for Release 5.0

FOR authorization checking exit routine determines whether the originator is authorized to enter commands FOR the target user ID, and if so the command sets authorized. This supports distributed (departmental) administration, centralized (networking) administration, * NOTFOR privileged user IDs, and so forth.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file, Enter:

```
FOR_AUTHORIZATION_CHECKING_EXIT= DVHXFA EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHXFA

Is called with the following parameters:

- Issuer's effective user ID (Id specified with ASUSER, otherwise the origin user ID.)
- Issuer's effective node ID (Local system if ASUSER is specified, otherwise the origin node ID.)
- The target ID (user ID or profile)

  Note: The user ID or a PROFILE, can also be an user. For example, with the ADD command you will receive messages from the issuing user ID instead of the target ID.
- The target node ID (ATNODE if specified, otherwise an asterisk)
- The command level
- The command set(s) required to issue the command
- Any other command parameters and the command name.

Return Codes

This routine must exit with one of the following:

Table 26. DVHXFA Return Codes

Return Code	Meaning
0	The command is authorized. No further authorization checking is performed by DirMaint.
30 - Nop	Regular DirMaint authorization checking is performed.
31	The command is not authorized. DirMaint will issue a generic error message and exit with the appropriate return code for that message.
Other	The command has been completely processed or rejected. DirMaint exits, passing back whatever return code it was given without issuing an error message.

Link Authorization (DVHXLA)

Environment
DIRMAINT service machine

Description
New for Release 5.0

Link authorization checking exit. This exit may be used to perform alternative checking for LINK authorization. It will be called whenever a LINK command is issued, other than a LINK DELETE. It will also be called for ADD (ADD generates LINK requests indirectly), REPLACE, and CHNGID.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file, Enter:

```
LINK_AUTHORIZATION_EXIT= DVHXLA EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXLA

Is called with the following parameters:

- The command causing this check: LINK
- The target ID class (PROFILE or USER)
- The target ID
- The target system affinity, usually *
- The minidisk owner's user ID
- The minidisk owner's virtual address
- The linker's proposed virtual address
- The requested link mode.

Return Codes
This routine must exit with one of the following:

Table 27. DVHXLA Return Codes

Return Code	Meaning
0	The LINK statement is accepted. No further checking is done.
30 - Nop	The LINK is neither accepted nor rejected. DirMaint continues by making the same checks as if the exit were not present.
31	The LINK is rejected. DirMaint should issue a generic error message.
Other nonzero	The LINK is rejected and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

Message Logging Filter (DVHXLF)

Environment

DIRMAINT service machine
DATAMOVE service machine
DIRMSAT service machines

Description

New for Release 5.0

Log record filtering exit; by default, all commands received by the DIRMAINT machine are auditable, and all messages sent by the DIRMAINT machine are auditable. This exit routine may selectively reduce the quantity of data logged.

Invocation

One of the following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
MESSAGE_LOGGING_FILTER_EXIT= DVHXLF EXEC
```

or

```
ESM_LOG_FILTER_EXIT= DVHXLF EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHXLF

Is called with the following parameters:

- A destination identifier: either an * for a message being sent to the service machine's own console, a *?* for a message being sent back to the originator of the command (user ID available in global variable ORIGUSER, node ID available in global variable ORIGNODE), or a distribution list nickname: DVHCERT, DVHDIRM, DVHHELP, DVHOPER, DVHSUPT, or DVHALL.
- The message identifier is DVH*rrrnnnn*S

  Where:

  rrr
  : Specifies the routine issuing the message.

  nnnn
  : Specifies the message number.

  S
  : Specifies the message severity.

- The prospective string to be logged.

Return Codes

This routine must exit with one of the following:

Table 28. DVHXLF Return Codes

Boolean Return Code	Meaning
0	Do not log this.
1 or other nonzero	Log this.

Link Notification (DVHXLN)

Environment
DIRMAINT service machine

Description
New for Release 5.0

Link notification exit may be used to notify other service machines of changes to directory LINKs. It will be called whenever a LINK command is issued, including a LINK DELETE. It will also be called for ADD (ADD generates LINK requests indirectly), REPLACE, CHNGID, and PURGE.

Invocation
The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.

```
LINK_NOTIFICATION_EXIT= DVHXLN EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXLN

Is called with the following parameters:

- The command causing this notification: CHNGID-NEW, CHNGID-OLD, CHVADDR-NEW, CHVADDR-OLD, LINK, PURGE, REPLACE-OLD, or REPLACE-NEW
- The target ID class (PROFILE, or USER)
- The target ID
- The target system affinity, usually *
- The minidisk owner's user ID
- The minidisk owner's virtual address
- The proposed virtual address of the linker
- The link mode, or DELETE.

Return Codes
Any return code is ignored upon exit.

Minidisk Password Notification (DVHXMN)

Environment
DIRMAINT service machine

Description
New for Release 5.0

Minidisk password change notification exit may be used to notify other service machines of changes to a user's minidisk password. It will be called whenever a password change is successful.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
MINIDISK_PASSWORD_NOTIFICATION_EXIT= DVHXMN EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXMN

Is called with the following parameters:

- Command causing this notification (one of the following):
  - ADD
  - AMDISK
  - CHNGID-NEW
  - CHNGID-OLD
  - CHVADDR-NEW
  - CHVADDR-OLD
  - DMDISK
  - MDISK
  - PURGE
  - REPLACE-NEW
  - REPLACE-OLD
  - TMDISK-OLD
  - TMDISK-NEW
- Target ID class (Always USER)
- Target ID
- Target system affinity, usually *
- Affected minidisk address
- Three minidisk passwords (if provided).

Return Codes
Any return code is ignored upon exit.

Minidisk Password Checking (DVHXMP)

Environment
DIRMAINT service machine

Description
Replaces DVHMPW

Minidisk password syntax verification exit; the passed parameters are enhanced for networking support. This exit will be called for the ADD (since ADD requests generate AMDISK requests), AMDISK, CHNGID, CHVADDR, and TMDISK commands, as well as for the MDISK command.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
MINIDISK_PASSWORD_CHECKING_EXIT= DVHXMP EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXMP

Is called with the following parameters:

- Command causing this check: AMDISK, CHNGID, CHVADDR, MDISK, or TMDISK
- Target ID class (PROFILE or USER)
- Target ID
- Target system affinity, usually *
- Affected minidisk address
- Three proposed passwords (if provided).

Return Codes
This routine must exit with one of the following:

Table 29. DVHXMP Return Codes

Return Code	Meaning
0	The minidisk passwords are accepted. No further checking is done.
30 - Nop	The passwords are neither accepted nor rejected. DirMaint continues by making the same checks as if the exit were not present.
31	The passwords are rejected. DirMaint should issue a generic error message.
Other nonzero	The passwords are rejected and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

Multiple User Prefix Authorization (DVHXMU)

Environment

DIRMAINT service machine
DATAMOVE service machine
DIRMSAT service machines

Description

New for Release 5.0

MULTIUSER authorization checking exit screens all attempts to use the MULTIUSER prefix operand. This exit must approve any use of this prefix operand.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
MULTIUSER_VERIFICATION_EXIT= DVHXMU EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHXMU

Is called with the following parameters:

- The user and node making the request
- The pattern associated with the request
- The command keyword and parms being used.

Return Codes

This routine must exit with one of the following:

Table 30. DVHXMU Return Codes

Return Code	Meaning
0	The use of the MULTIUSER prefix is authorized.
30	The use of the MULTIUSER prefix is rejected with a message from DirMaint. This exit must explicitly authorize its use.
31	The use of the MULTIUSER prefix is rejected with a generic message from DirMaint. This exit must explicitly authorize its use.
Other nonzero	The use of the MULTIUSER prefix is rejected. It is assumed that the exit has issued an error message.

Password Change Notification (DVHXPN)

Environment
DIRMAINT service machine

Description
New for Release 5.0

Password change notification exit may be used to notify other service machines of changes to a user's logon password. It will be called whenever a password change is successful.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
PASSWORD_CHANGE_NOTIFICATION_EXIT= DVHXPN EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXPN

Is called with the following parameters:

- The keyword USER (if invoked in the user's virtual machine) or DIRMAINT (if invoked by the DIRMAINT virtual machine)
- The command causing this notification: ADD, CHNGID-NEW, CHNGID-OLD, PURGE, PW, REPLACE, or SETPW
- The target ID class (always USER)
- The target ID
- The target system affinity, usually *
- The new or old password.

Return Codes
Any return code is ignored upon exit.

Password Notice Printing (DVHXPP)

Environment
DIRMAINT service machine

Description
New for Release 5.0

Password notice print exit may be used to forward printed password notices to a network printer for those systems that do not have a local printer.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
PW_NOTICE_PRT_EXIT= DVHXPP EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXPP

Is called with the following parameters:

- The user ID and node ID for whom the notice should be printed
- The file name, file type, and file mode of the file to be printed
- The desired spool file class
- Days since the password was changed
- Threshold days associated with this user.

Return Codes
This routine must exit with one of the following:

Table 31. DVHXPP Return Codes

Return Code	Meaning
0	The notice has been printed.
30	Reserved for exit routine not found; the print file will be sent to the local printer, unless PW_NOTICE_PRT_CLASS= NONE is specified.
Other nonzero	An error has occurred and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

Request After Processing (DVHXRA)

Environment

DIRMAINT service machine
DATAMOVE service machine
DIRMSAT service machines

Description

Replaces DVHUA2

Request after processing exit; the passed parameters are enhanced for networking support.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
REQUEST_AFTER_PROCESSING_EXIT= DVHXRA EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHXRA

Is called with the following parameters:

- Return code from processing the command
- Command name.

In addition, the following two interface variables are available:

CMD_STRING

The command string as verified and returned by the parser. Command and parameter abbreviations HAVE been resolved. All prefix variables (TOSYS, ASUSER, BYUSER, FORUSER, and ATNODE) have been stripped off and stored in their own separate global variables. They will each have a value, if not specified on the user's command then defaults will be supplied.

LOG_STRING

The command string as verified and returned by the parser, with passwords and other sensitive information *masked* for security. The prefix operands are included as entered by the user, defaults are NOT filled in for omitted parameters.

These two interface variables are obtainable using: 'PIPE VAR variable_name 2 | ...'. DVHXRA is called indirectly through the DVHCEXIT EXEC. An *invocation* or *generation* number of 2 is necessary to get the value from DVHCEXIT's caller, DVHCMD.

Return Codes

Return codes are ignored upon exit.

Request Before Processing (DVHXRB)

Environment

DIRMAINT service machine
DATAMOVE service machine
DIRMSAT service machines

Description

Replaces DVHUA1

Request after parsing, before processing exit. The passed parameters are enhanced for networking support, and additional return codes are recognized.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
REQUEST_BEFORE_PROCESSING_EXIT= DVHXRB EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter

DVHXRB

Is called with no parameters, but with two interface variables set:

CMD_STRING
The command string as verified and returned by the parser; command and parameter abbreviations HAVE been resolved. All prefix variables (TOSYS, ASUSER, BYUSER, FORUSER, and ATNODE) have been stripped off and stored in their own separate global variables. They will each have a value, if not specified on the user's command then defaults will be supplied.

LOG_STRING
The command string as verified and returned by the parser, with passwords and other sensitive information *masked* for security; the prefix operands are included as entered by the user. Defaults are NOT filled in for omitted parameters.

These two interface variables are obtainable using: 'PIPE VAR variable_name 2 | ...'. DVHXRB is called in directly by DVHRQST. An *invocation* or *generation* number of 2 is necessary to get the value from DVHRQST.

In addition, the following interface variable is available:

SPOOLFILE
A numeric value indicates that “*spoolfile* RDRFILE Z” is associated with this command. The value may be either 4 or 6 digits. A non-numeric value indicates there is no file associated with this command.

This variable is available using:

```
GLOBALV SELECT DVH15 GET spoolfile
```

Return Codes

This routine must exit with one of the following:

Table 32. DVHXRB Return Codes

Return Code	Meaning
30 - Nop	Regular DirMaint processing continues.
31	A replacement command string has been set into variable CMD_STRING using 'PIPE ... | VAR CMD_STRING 2'. DVHXRB is called indirectly through the DVHCEXIT EXEC. An *invocation* or *generation* number of 2 is necessary to pass the value back to DVHCEXITs caller, DVHRQST. If the CMD_STRING is changed and contains passwords or other sensitive information, the LOG_STRING should also be changed to the equivalent string with the sensitive information changed to a string of *XXXs*.
Other	The command has been completely processed. DirMaint exits, passing back whatever return code it was given.

Request Before Parsing (DVHXRC)

Environment

DIRMAINT service machine
DATAMOVE service machine
DIRMSAT service machines

Description

Request before parsing exit.

Invocation

The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.

```
REQUEST_BEFORE_PARSING_EXIT= DVHXRC EXEC
```

For more information, see "CONFIG DATADVH" on page 44.

Interface Parameter

DVHXRC

Is called with the command string as entered by the user; command and parameter abbreviations have NOT been resolved.

In addition, the following interface variable is available:

SPOOLFILE
: A numeric value indicates that "*spoolfile* RDRFILE Z" is associated with this command. The value may be either 4 or 6 digits. A non-numeric value indicates there is no file associated with this command.

This variable is available using:

```
GLOBALV SELECT DVH15 GET spoolfile
```

Return Codes

This routine must exit with one of the following:

Table 33. DVHXRC Return Codes

Return Code	Meaning
30 - Nop	Regular DirMaint processing continues.
31	A replacement command string has been set into variable CMD_STRING using 'PIPE ... | VAR CMD_STRING 2'. DVHXRC is called indirectly through the DVHCEXIT EXEC. An *invocation* or *generation* number of 2 is necessary to pass the value back to DVHCEXITs caller, DVHRQST.
Other	The command has been completely processed. DirMaint exits, passing back whatever return code it was given.

Local Stag Authorization (DVHXTA)

Environment
DIRMAINT service machine

Description
New for Release 5.0

Local STAG authorization exit controls authorization allowing manipulation of locally defined “Star Tags,” or STAGs. A STAG is a comment in the directory in the form:

**tagname*:

The *tagname* is 1 to 10 alphanumeric characters that is preceded by an asterisk (*) and followed by a colon (:). The **tagname*: is made known to DirMaint by using the *DIRM DEFINESTAG* command. This exit may determine whether the general users are allowed to change the value of a specific *tag or *tags.

Invocation
The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.

```
LOCAL_STAG_AUTHORIZATION_EXIT= DVHXTA EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXTA

Is called with the following parameters:

- The command causing this check: SETSTAG or STAG
- The target ID class (PROFILE or USER)
- The target ID
- The target system affinity, usually *
- The tag name (including the leading asterisk and trailing colon)
- The data the user wants set.

Return Codes
This routine must exit with one of the following:

Table 34. DVHXTA Return Codes

Return Code	Meaning
0	The operation is accepted. No further checking is done.
30	Reserved for IBM use
31	The operation is rejected. DirMaint should issue a generic error message.
Other nonzero	The operation is rejected and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

Backup Tape Mount (DVHXTP)

Environment

DIRMAINT service machine

Description

New for Release 5.0

Backup tape mount exit may be used to mount backup tape using AMMR, VMTAPE, and other tape library management programs.

Invocation

The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.

```
BACKUP_TAPE_MOUNT_EXIT= DVHXTP EXEC
```

For more information, see "CONFIG DATADVH" on page 44.

Interface Parameter

DVHXTP

Is called with the following parameters:

REQUEST — A new tape mount is being requested. The *protocol* and *tdev* variables will need to be changed to suit the needs of you installation.

REMINDER — This is a periodic reminder of a previously issued request.

REJECTED — The attached tape was not accepted because the tape has:

- A standard label, but an unlabeled tape is expected
- No label, but a standard labeled tape is expected
- A label, but it does not match the label expected.

REQUEST2 — A replacement tape mount is being requested following a rejection.

CANCEL — The outstanding tape request is canceled without being satisfied.

ACCEPTED — The request has been satisfied and the tape is acceptable.

The external tape identification (1-8 file name characters).

The internal tape identification (1-6 file name characters).

Return Codes

This routine must exit with one of the following:

Table 35. DVHXTP Return Codes

Return Code	Meaning
0	The action has been completed.
30	Reserved for IBM use
Other nonzero	An error has occurred and the exit routine has already issued the appropriate messages. The return code should be the same as the message number.

Note: The exit routine must not wait for a result to be satisfied. It must send a message to the tape operator or make a request to a tape management program, such as IBM's Attachable Media Manager (AMMR) or equivalent vendor product, and then return to the calling program. The DIRMAINT service machine will periodically check to see if the tape has been attached, and call the exit again as needed.

User Change Notification (DVHXUN)

Environment
DIRMAINT service machine

Description
New for Release 5.0

User ID change notification exit may be used to notify other service machines of new, deleted, or changed user IDs. It will be called whenever the ADD, CHNGID, or PURGE commands have added, changed, or deleted a user ID, profile, or a POSIX group.

Invocation
The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
USER_CHANGE_NOTIFICATION_EXIT= DVHXUN EXEC
```

For more information, see “CONFIG DATADVH” on page 44.

Interface Parameter
DVHXUN

Is called with the following parameters:

- The command causing this notification: ADD, CHNGID-NEW, CHNGID-OLD, or PURGE
- The target ID class (PROFILE or USER)
- The target ID
- The target system affinity, usually *.

Return Codes
Any return code is ignored upon exit.

Guidelines for Creating or Modifying Exit Routines

If you choose to write or modify existing routines, follow these guidelines:

- Each exit routine should validate the INTERFACE level descriptor global variable. If unsupported, exit routines in the user machine should issue message DVH1901 and exit with return code 1901; although exit routines in the DIRMAINT, DATAMOVE, and DIRMSAT server machines should issue message 2901 and call DVHSHUT to logoff. Message DVH1901 indicates that the exec has encountered an unrecognized interface level descriptor on the users machine. Message DVH2901 indicates the same problem on the server.
- Each exit routine must support the TRACE global variable. This is a string of values of the form: DVHrtn=option, or DVH*=option. If the name of the routine is in the TRACE list, or except for DVHXLF if DVH* is in the trace list, exit routines in the user machine must issue message DVH1161 and set REXX tracing to the specified option on entry, and must issue message DVH1162 on exit, although exit routines in the server machines: DIRMAINT, DATAMOVE, and DIRMSATs must issue messages DVH2161 and DVH2162 respectively.
- In the event of an unexpected return code from any CP or CMS command, the exit routine should issue message DVH1119, user machine, or DVH2119, server machine respectively.
- Except for messages DVHx901, DVHx161, DVHx162, or DVHx119 just described above, the exit routines should generally not issue any messages. This is especially true for cases where the command originator is not authorized to enter that command or has made an error in the command. Instead, the exit routine should exit with an appropriate return code. For exceptional error conditions, indicative of incorrect installation or tailoring, the exit routine should issue appropriate error messages, and either exit with the return code passed back from the message routine, or if continued operation is inappropriate, call DVHSHUT to logoff the service machine.
- Each exit routine must be *fail safe* and *restarting*. If command processing fails for any reason after calling one or more of these exit routines, the command will usually be reprocessed and the exit routines will be called again. Be aware that human intervention may prevent the command from being reprocessed, and the exit routine documentation and site *run book* must describe what manual actions must be taken in each case if the command is not reprocessed.

Message Numbers Available for Installation-Written Exits

In addition to the messages listed in “Guidelines for Creating or Modifying Exit Routines,” the following message numbers are reserved for installation-written exit routines:

109*n* Reserved for hard coded, nontranslatable, messages not issued from the message repository through DVHMSG on the user machine side; no expected usage.

19*nn* Other messages issued through DVHMSG on the user machine side; number DVH1901 is used as stated above.

209*n* Hard coded, nontranslatable, messages not issued from the message repository through DVHMSG on the server machine side; potential callers are most likely to be either DVHXPROF or DVHXLF.

29*nn* Other messages issued through DVHMSG from: DVHXPROF, DVHXLF, DVHXRC, DVHXRB, DVHXFA, or DVHXRA; number 2901 is used as stated above.

309*n* Reserved for hard coded, nontranslatable, messages not issued from the message repository through DVHMSG on the service machine side from customer written command handlers; no expected usage

39*nn* Other messages issued through DVHMSG from: DVHXAV, DVHXAN, DVHPXV, DVHXPN, DVHXMP, DVHXMN, DVHXLA, DVHXLN, DVHXDA, DVHXDN, DVHXUN, DVHDXP, or from customer written command handlers.

Global Variables Available for the DVHCX* and DVHPX* Exits

The following persistent LASTING GLOBALV variables are available to exit routines in the user's virtual machine. If not set, the default is the first value listed.

Table 36 (Page 1 of 2). LASTING GLOBALV Variables for the DVHCX and DVHPX* Exits*

Variable	Description
CMDLEVEL	Specifies whether the user is entering the newer DirMaint Release 5.0 level command syntax or the older DirMaint Release 4,0 level command syntax. The valid values are: 150A or 140A.
CMDSET	Specifies what set of commands the user is expecting to enter. The valid values are the concatenation of the first letter for: General<Admin><Dasd><Helpdesk><Monitor><Oper><Program><Support><Zinternal>. For example, specifying the command set General, Helpdesk, and Support is entered as: `GHS` **Note:** These are the IBM-supplied default command set definitions, but may be tailored by the customer.
DASUSER	Specifies the default value for ASUSER. The valid values are an * or any valid file name that may be used as a user ID.
DATNODE	Specifies the default value for ATNODE. The valid values are an * or any valid file name.
DBYUSER	Specifies the default value for BYUSER. The valid values are an * or any valid file name that may be used as a user ID.
DFORUSER	Specifies the default value for FORUSER. The valid values are an * or any valid file name that may be used as a user ID.
DTOSYS	Specifies the default value for TOSYS. The valid values are an * or any valid file name.
LANG	Specifies the user's chosen language. Valid values are AMENG, UCENG, or 1SAPI. DirMaint Release 5.0 is enabled for additional languages. Check with your marketing representative to determine what additional languages are available. DirMaint Release 4.0 supported FRANC, GER, and KANJI.
NEEDPASS	Specifies whether the user is required to supply their password for interaction with the DIRMAINT machine. Valid values are YES or NO.
REQUEST	Keeps track of how many requests have been sent to the DIRMAINT service machine for processing. Valid values are 1 through 9999.

Table 36 (Page 2 of 2). LASTING GLOBALV Variables for the DVHCX and DVHPX* Exits*

Variable	Description
TEST	Specifies whether the user is entering commands for production or for testing and problem diagnosis. The valid values are OFF, MSG, or SAY.
TRACE	Specifies which routines, if any, should be traced and the degree of tracing desired. Valid values are one or more occurrences of DVHname=trace_opt; where *DVHname* is the file name of any executable product part, the equal sign is a required delimiter, and the *trace_opt* is any valid REXX Trace option - A/C/E/F/I/L/N/O/R/S. The *?* prefix is allowed.
LOG_STRING	Specifies the command string being processed with any passwords or other sensitive data. This string should be changed to *XXXs*.

The following temporary SESSION GLOBALV variables are available to exit routines in the user's virtual machine. Unless otherwise stated, there is no default.

Note: Of the following variables, only INTERFACE.DVHCXC is available to the COMMAND_BEFORE_PARSING_USER_EXIT (DVHCXC EXEC). The other variables are not available until after parsing has been completed.

Table 37 (Page 1 of 2). SESSION GLOBALV Variables for the DVHCX and DVHPX* Exits*

Variable	Description
ASUSER	Specifies the *userid* against which the password will be verified, and whose privileges will be used to perform the command. The default is an * for the *userid* of the user entering the command, unless overridden by DATUSER.
ATNODE	Specifies which node in a multiple system complex the command is intended to affect. The default is an * for all nodes, unless overridden by DATNODE.
BYUSER	Identifies the *userid* against which the password will be verified, but performing the command using the privileges of the command issuer. The default is an * for the user ID of the user entering the command, unless overridden by DBYUSER.
FORUSER	Identifies the *userid* for whom the command is issued. The default is an * for the *userid* of the user who entered the command, unless overridden by either DFORUSER or ASUSER or DASUSER.
INTERFACE	Specifies the transaction interface protocol being used by the user's virtual machine for exchange with the DIRMAINT machine. It is composed of the year and month of the most recent interface design change, for example 199501. There is a separate interface variable for each user exit routine: INTERFACE.DVHCXA, INTERFACE.DVHCXB, INTERFACE.DVHCXC, INTERFACE.DVHPXA, INTERFACE.DVHPXR, and INTERFACE.DVHPXV. Also, INTERFACE.DVHAPI is available for use by the caller of the DVHSAPI EXEC. For more information see the VALIDLVLS variable.
MULTIUSER	Specifies the nickname or pattern to be used in the operation on multiple directory entries.
PROMPT	Specifies that the user choose to be prompted for sensitive information (passwords) omitted from the command line.

Table 37 (Page 2 of 2). SESSION GLOBALV Variables for the DVHCX and DVHPX* Exits*

Variable	Description
TOSYS	Identifies the *nodeid* in a remote network on which the command is to be processed. The default is an * for the system or local system cluster where the command is issued, unless overridden by DTOSYS.
VALIDCMDS	Specifies the valid command levels from which the user may choose. The current value is 150A 140A.
VALIDLVLS	Specifies the valid interface design levels.

All IBM defined global variables are stored in the DVH15 variable pool. Customer defined global variables should be stored in either:

- DVH15LCL
- DVH15USR
- DVH15XIT

The INTERFACE variables are:

- ASUSER
- ATNODE
- BYUSER
- CMDSET
- FORUSER
- LANG
- TEST
- TOSYS
- TRACE

are considered to be part of the product specific program interface to the various exit routines that run in the user's virtual machine:

- DVHCXA - COMMAND_AFTER_PROCESSING
- DVHCXB - COMMAND_BEFORE_PROCESSING
- DVHCXC - COMMAND_BEFORE_PARSING
- DVHPXA - PASSWORD_AFTER_PROCESSING
- DVHPXR - PASSWORD_RANDOM_GENERATOR
- DVHPXV - PASSWORD_SYNTAX_VERIFICATION

None of these variables are intended for use outside of the product or these exit routines.

A new interface level descriptor will be assigned:

- In the event that any changes are made in the definition of the parameters passed to these exit routines, or to DVHMSG because it is called by the preceding exit routines, or if any changes are made in the definition of the expected results from DVHCXC, DVHCXB, DVHPXR, or DVHPXV.
- When any changes are made in the way information is exchanged between the user's virtual machine and the DIRMAINT service machine, including the addition of new commands or operands to the command set or changes in the user message repositories, or when any changes are made in the format of any data files that are intended for local tailoring.

Global Variables Available for the DVHX* Exits

The following persistent LASTING GLOBALV variables are available in the DIRMAINT, DATAMOVE, and DIRMSAT virtual machines.

Table 38. LASTING GLOBALV Variables for the DVHX Exits*

Variable	Description
TRACE	Specifies which routines, if any, should be traced and the degree of tracing desired. Valid values are one or more occurrences of DVHname=trace_opt; where DVH*name* is the file name of any executable product part, the equal sign is a required delimiter, and the "trace_opt" is any valid REXX Trace option - A/C/E/F/I/L/N/O/R/S. The ? prefix is allowed but not recommended.

The following temporary SESSION GLOBALV variables are available to exit routines in the DIRMAINT, DATAMOVE, and DIRMSAT virtual machines. Unless otherwise stated, there is no default.

Notes:

1. Some of the following variables are not available until after parsing has been completed, and are therefore not available to the REQUEST_BEFORE_PARSING_EXIT (DVHXRB EXEC). Those that ARE available are: CMDLEVEL, INTERFACE.XRB, ORIGNODE, ORIGUSER, and ROLE.
2. Some of the following variables are not available until after authorization checking has been completed, and therefore are not available to the REQUEST_BEFORE_PROCESSING_ EXIT (DVHXRB EXEC). Those that are NOT available are: ASUSER, ASNODE, and TARGETID.

Table 39 (Page 1 of 2). SESSION GLOBALV Variables for the DVHX Exits*

Variable	Description
ASUSER	Specifies the *userid* against which the password will be verified, and whose privileges will be used to perform the command. The default is the same as ORIGUSER.
ASNODE	Specifies the nodeid of the user whose privileges will be used to perform the command. If ASUSER is specified, ASNODE will be an asterisk (*) for a local cluster user. If ASUSER is not specified, ASNODE will be the same as ORIGNODE.
ATNODE	Specifies which node in a multiple system complex the command is intended to affect. The default is an * for all nodes within the CSE cluster.
BYUSER	Specifies the user ID against which the password will be verified, but issues the command using the privileges of the user entering the command. The default is an * for the user ID of the user entering the command.
CMDLEVEL	Specifies whether the user is entering the newer DirMaint Release 5.0 level command syntax or the older DirMaint Release 4 level command syntax. Valid values are 150A or 140A.

Table 39 (Page 2 of 2). SESSION GLOBALV Variables for the DVHX Exits*

Variable	Description
FORUSER	Specifies the user for whom the command is issued. The default is an * for the user ID of the user who entered the command, unless overridden by ASUSER.
INTERFACE	Specifies the transaction interface protocol being used by the user's virtual machine for exchange with the DIRMAINT machine. It is composed of the year and month of the most recent interface design change, for example 199501. There is a separate interface variable for each system exit routine: INTERFACE.*mmmmmm* Where: *mmmmmm* Specifies the six character identifier used by the IBM-supplied sample exit routines for use in issuing messages.
LANG	Specifies the user's chosen language. Valid values are AMENG, UCENG, or 1SAPI. DirMaint release 4.0 is enabled for additional languages. Check with your marketing representative to determine what additional languages are available. DirMaint release 4.0 supported FRANC, GER, and KANJI.
ORIGNODE	Identifies the *nodeid* where the command originated. An asterisk (*) indicates any node within the local CSE cluster.
ORIGUSER	Identifies the *userid* where the command originated.
REQUEST	Keeps track of how many requests have been sent to the DIRMAINT service machine for processing. Valid values are 1 through 9999.
RESTART	Specifies whether the command handler or exit routine is being called during restart processing to recover after an interruption of some type. Valid values are NO or YES.
ROLE	Specifies whether the service machine is the DIRMAINT machine, the DATAMOVE machine, or a DIRMSAT machine.
TARGETID	Identifies the *userid* whose directory entry is to be affected. The value will be the first one of the following that applies: • The user ID specified within the privileged command, if present. (Applicable for command level 140A only.) • The FORUSER ID, if not an asterisk. • The ASUSER ID, if not an asterisk. • The ORIGUSER ID.
VALIDCMDS	Specifies the valid command levels from which the user may choose. The current values are 150A 140A.
VALIDLVLS	Specifies the valid interface design levels.

All IBM defined global variables are stored in the DVH15 or DVH1SNDX variable pool. Customer defined global variables should be stored in either DVH15LCL or DVH15XIT.

The INTERFACE variable and:

- ASNODE
- ATNODE
- ASUSER
- BYUSER
- FORUSER

- CMDLEVEL
- CMDSET
- ORIGNODE
- ORIGUSER
- RESTART
- ROLE
- TARGETID
- TRACE

are considered to be part of the product specific program interface to the various exit routines that run in the DIRMAINT, DATAMOVE, or DIRMSAT service machines. None of these variables are intended for use outside of the product or these exit routines.

A new interface level descriptor will be assigned:

- In the event that any changes are made in the definition of the parameters passed to these exit routines, or to DVHMSG because it is called by the preceding exit routines, or if any changes are made in the definition of the expected results from DVHXRB.
- When any changes are made in the way information is exchanged between the user's virtual machine and the DIRMAINT service machine, including the addition of new commands or operands to the command set or changes in the user message repositories, or when any changes are made in the format of any data files that are intended for local tailoring.

Utility Routines

There are several house keeping utility routines that may be modified by the customer. These events are at pre-scheduled times of day or at periodic intervals. The IBM-supplied utilities are:

- DVHOURLY
- DVHDAILY
- DVHNDAY

These utilities may be either EXECs or MODULEs; the IBM supplied utilities are EXECs. These utilities run in all three service machines: DIRMAINT, DATAMOVE, and DIRMSATs.

Example: Housekeeping utility invocations found in DIRMAINT DATADVH, DATAMOVE DATADVH, and DIRMSAT DATADVH are:

```
==/==/== 00:00:05 00/00/00 CMS EXEC DVHNDAY
==/==/== 00:01:00 00/00/00 CMS EXEC DVHDAILY
==/==/== +01:00:0 00/00/00 CMS EXEC DVHOURLY
```

The format of housekeeping utility invocations are:

Table 40. Format of Housekeeping Utility Fields

Columns	Function
1-8	Specifies the day or days when the event is to be scheduled
10-17	Specifies the time or times of day when the event is to be scheduled
19-26	Must have an initial value of *00/00/00*, and are reserved for system use
28-240	Specifies the event.

Notes:

1. The invocations for all of these house keeping utilities must be preceded by the CMS keyword.
2. If the utility is an EXEC, the CMS keyword must be followed by the EXEC keyword which in turn must be followed by the utility name and any optional parameters.
3. If the utility is a MODULE, the CMS keyword must be followed by the utility name and any optional parameters; there is no MODULE keyword.
4. For more information on the format of the date and time scheduling fields, see *VM/ESA: CMS Utilities Feature*.

Chapter 10. Planning for Diagnosis

IBM Directory Maintenance VM/ESA (DirMaint*) is a Conversational Monitor System (CMS) application that helps you manage your VM directory. Directory management is simplified by DirMaint's command interface and automated facilities. DirMaint's directory statement-like commands initiate directory transactions. DirMaint's error checking ensures that only valid changes are made to the directory, and that only authorized personnel are able to make the requested changes. Any transaction requiring the allocation or deallocation of minidisk extents can be handled automatically. All user initiated transactions can be password controlled and can be recorded for auditing purposes.

To isolate and solve a problem, different people and different courses of action may be needed. DirMaint may be able to recover from, or circumvent, a problem automatically or with the help of an operator. However, if a problem recurs frequently, it should not be left unchecked.

This chapter describes a few considerations for diagnosing DirMaint problems. The following checklist describes some things to consider when planning for diagnosis.

Planning Checklist for Diagnosis and Recovery

- □ ***Ensure*** that operators are trained to respond to problem situations — either to take action themselves or to call for help.
- □ ***Ensure*** that system backups (including the source directory) are maintained and that a plan is in place to recover from their loss.
- □ ***Identify*** support personnel who will be on call when operators need help.
- □ ***Determine*** procedures that users should follow if they run into problems.
- □ ***Identify*** off-site contacts for problems that involve communication lines or other systems.

Diagnosing Problems Using DirMaint Facilities

DirMaint provides facilities to assist in diagnosing problems. These facilities include messages, commands, and tracing facilities. DirMaint also provides facilities to automatically attempt to recover from some error situations.

DirMaint produces messages to document its actions. It also produces diagnostic messages if errors occur. These messages and their explanations often suggest follow-up actions to resolve or diagnose the problem. For more information about diagnostic messages, see the *Directory Maintenance VM/ESA: Diagnosis Reference*. also contains descriptions on the general logic flow for each subsection of DirMaint code. These descriptions may prove useful when performing problem source identification.

Establishing Information-Collecting Procedures

Because operators may receive requests to collect diagnostic information, they should be instructed on what to do in various situations and when to get outside help. Some of this work can be simplified using execs, DirMaint commands or statements.

Appendix A. External Security Manager Considerations

Tailoring your DirMaint system includes implementing security measures against unauthorized access to data, as well as inadvertent destruction of data. DirMaint itself provides a level of security through its command set authorizations. These can be tailored to suit the using installation's needs. However, for critical data files, additional security measures should be implemented. This can be done using an (ESM) External Security Manager such as RACF (Resources Access Control Facility). An ESM controls who can have access, and what kind of access they can have to specific data files and disks. If an ESM is implemented at your installation, DirMaint must be given the appropriate access to the disks and files you want it to manage.

This appendix describes how to enable the proper RACF authorizations for the operation of DirMaint:

- Guidance for defining the DirMaint service machines to your ESM
- Granting the necessary authority to the various DirMaint service machines.
- Facilities available for detecting and foiling attempts to break system security
- Considerations for maintaining system integrity.

These recommendations are optional and whether you follow them depends on the level of security that your installation requires.

If you add additional DATAMOVE or DIRMSAT machines to your system at a later time, remember to review this chapter and perform the necessary steps for the new service machines.

The use of an ESM is optional. If you do not have an ESM installed on your system, you may skip this appendix.

Installing DirMaint With an External Security Manager Other Than RACF

DirMaint is intended to function on VM/ESA systems with external security manager programs other than RACF. The methods of defining the DirMaint product, data disks, other controllable resources to the ESM, and granting access to the defined resources all vary from one ESM to another.

For more information on administration, see the documentation provided with your ESM.

If you need assistance with your ESM, contact the vendor for your ESM product. If you need assistance translating the following RACF terminology into equivalent terminology for your ESM, contact the IBM marketing representative or the IBM branch office serving your area.

Installing DirMaint with RACF

RACF for VM can be used to enhance the security and integrity of your system by:

- Helping your installation implement its security policy
- Identifying and authenticating each user
- Controlling each user's access to sensitive data
- Logging and reporting events that are relevant to the system's security.

For more information on RACF for VM, see these publications:

RACF Program Directory, GC28-1034

RACF General Information, GC28-0722

RACF Command Language Reference, SC23-3731

RACF System Programming Library, SC23-3725

RACF Security Administrator's Guide, SC23-3726

RACF Auditor's Guide, SC23-3727

RACF Diagnosis Guide, LY28-1016

RACF Commands

The commands used with RACF.

All of the RACF command examples, used in this publication use the syntax for sequential RACF commands.

Example—Syntax used for RACF Commands

```
<EXEC> RAC command_string_1
<EXEC> RAC command_string_2
```

If you choose, you can establish a RACF command session, and enter the corresponding commands.

Example—Command Session for RACF

```
<EXEC> RACF
command_string_1
command_string_2
END
```

RACF Command Requirements

The set of CP commands that you can use depends on the privilege class or classes assigned to you. CP *privilege* classes RACF does not recognize implemented by the User Class Restructure (UCR) are:

- Class I through Z

  or

- Class 1 through 6

RACF expects *privileged* machines to have one or more of these CP classes:

- Class A through F

or

- Class H

Example—Preventing DirMaint Service Machine Class B Privileges: If you choose not to give the DirMaint service machine any of these privilege classes, class B in particular, then you must explicitly permit the DirMaint machine for use of subcode X'04' of DIAGNOSE code X'A0', enter:

```
RAC PERMIT DIAG0A0.VALIDATE CLASS(VMCMD) ID(xxxxxxxx) ACCESS(READ)
```

Where:

xxxxxxxx
Identifies the user ID of the DirMaint service machine.

Note: The DATAMOVE and DIRMSAT machines do not need this capability.

Enabling Auditing Using RACROUTE

If your system is intended to meet either:

- Class C2

  or

- Class B1 TCB criteria

You must enable the DirMaint service machines: DIRMAINT, DATAMOVE, and DIRMSAT to record information in the RACF audit trail. You may enable this function even if your system is not intended as a TCB.

Step 1. Recording Activity using the RACROUTE Command

To record activity in the RACF system audit trail, they must each be authorized, enter:

```
RAC SETROPTS CLASSACT(FACILITY)
RAC SETROPTS RACLIST(FACILITY)
RAC RDEFINE FACILITY ICHCONN UACC(NONE)
RAC SETROPTS RACLIST(FACILITY) REFRESH
RAC PERMIT ICHCONN CLASS(FACILITY) ID(xxxxxxxx) ACCESS(UPDATE)
```

Note: These commands may fail if they have already been issued before.

Where:

xxxxxxxx
Identifies the user ID of the DirMaint service machine.

Step 2. Linking to RACF 305

The DirMaint service machines must all be made aware of the user ID of the RACF service machine that is recording the audit log. This is contained in a file named RACF SERVMACH. For RACF 1.9.2, this file usually resides on the system 19E Y-disk, although it could also be located on the RACF machine's 305 minidisk. For RACF 1.10, this file usually resides on the 5767002P 29E disk. If you chose to have the DirMaint service machines link to the RACF 305 minidisk, you must permit this access, unless the service machine has been made exempt. To permit access, enter:

```
RAC PERMIT racfvmid.305 CLASS(VMMDISK) ID(xxxxxxxx) ACCESS(READ)
```

Where:

racfvmid
Specifies the name of the RACF service machine that you select to handle DirMaint audit requests

xxxxxxxx
Identifies the user ID of the DirMaint service machine.

Step 3. Accessing the RPIUCMS MODULE

The DirMaint service machines must all have access to the RPIUCMS MODULE. This file usually resides on the system 19E Y-disk, but may reside on any disk in the DirMaint service machine's search order. For RACF 1.10, this file usually resides on the 5767002P 29E disk. If necessary, you may add a LINK directory statement to the RACF service machine's directory entry for this disk, and update the DVHPROFA * files used by the DirMaint service machine to access the disk containing the RPIUCMS MODULE at an available filemode. Unless the disk where the RPIUCMS MODULE resides is:

- a public disk

  or

- has UACC(READ)

  or

- the DirMaint service machines have been made exempt

the service machines must be permitted for READ access to that disk, enter:

```
RAC PERMIT racfvmid.vaddr CLASS(VMMDISK) ID(xxxxxxxx) ACCESS(READ)
```

Where:

racfvmid.vaddr
Specifies the name of the virtual machine owning the disk and the virtual disk address containing the RPIUCMS MODULE file

xxxxxxxx
Identifies the user ID of the DirMaint service machine.

For more information on the RPIUCMS MODULE, see the *RACF Program Directory, GC28-1034*.

Step 4. The directory entry for the DirMaint service machines using this capability must all contain this statement:

```
IUCV ANY PRIORITY MSGLIMIT 100
```

Note: A MSGLIMIT value of 100 is initially suggested. It may be adjusted as your experience dictates.

Making the DirMaint Service Machines Exempt

If your system is intended to meet Class B1 TCB criteria you must enable Mandatory Access Control (MAC) on your system. If MAC is active on your system, you must make the DirMaint service machines (DIRMAINT, DATAMOVEs and DIRMSATs) exempt from MAC checking.

Even if MAC is not active on your system, making the DirMaint service machines exempt from as much RACF control and checking as possible will improve performance.

Example—Creating a VMXEVENT Profile

Make the DirMaint service machines exempt from MAC checking by entering the following commands:

```
RAC SETROPTS CLASSACT(VMXEVENT)
RAC SETROPTS RACLIST(VMXEVENT)
RAC RDEFINE  VMXEVENT USERSEL.xxxxxxxx
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(LINK/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(STORE.C/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(TAG/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(TRANSFER.D/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(TRANSFER.G/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(TRSOURCE/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(DIAG0D4/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(DIAG0E4/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(DIAG0A0/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(RSTDSEG/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(MDISK/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(APPCPWVL/NOCTL)
```

Note: These commands may fail if they have already been issued before.

Where:

xxxxxxxx
Identifies user ID of the DirMaint service machine to be made exempt.

If the service machine is already active before the VMXEVENT profile has been created, activate the profile by entering:

```
RAC SETEVENT REFRESH USERSEL.xxxxxxxx
```

Where:

xxxxxxxx
Identifies user ID of the DirMaint service machine to be made exempt.

A machine becomes exempt only if NOCTL is specified for ALL controllable events.

Note: The list of controllable events varies depending on which release of RACF and VM you are running. For more information, see *VM/ESA Trusted Facility Manual*, and *RACF/VM Security Administration Guide*.

Enabling Discretionary Access Control

If your system is intended to meet either:

- Class C2

  or

- Class B1 TCB criteria

you must enable your system for DAC. There are two forms of DAC that affect the DirMaint service machines. You may enable one or the other or both even if your system is not intended as a TCB.

Minidisk DAC

If minidisk DAC is enabled on your system:

Step 1. Access DirMaint primary interface files, enter:

```
RAC RALTER VMMDISK P748XE4M.11F UACC(READ)
```

Step 2. Access the secondary interface files and help files for testing, enter:

```
RAC RALTER VMMDISK P748XE4M.41F UACC(READ)
RAC RALTER VMMDISK P748XE4M.29E UACC(READ)
RAC RALTER VMMDISK P748XE4M.29D UACC(READ)
```

Step 3. Permit the DirMaint service machines to the necessary disks, unless they have been made exempt, enter:

```
RAC PERMIT P748XE4M.491 CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT P748XE4M.492 CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT P748XE4M.11F CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT P748XE4M.41F CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT P748XE4M.29E CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT MAINT.123    CLASS(VMMDISK) ID(DIRMAINT) ACCESS(ALTER)

RAC PERMIT P748XE4M.491 CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
RAC PERMIT P748XE4M.492 CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
RAC PERMIT P748XE4M.11F CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
RAC PERMIT P748XE4M.41F CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
RAC PERMIT P748XE4M.29E CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)

RAC PERMIT P748XE4M.491 CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT P748XE4M.492 CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT P748XE4M.11F CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT P748XE4M.41F CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT P748XE4M.29E CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT MAINT.123    CLASS(VMMDISK) ID(DIRMSAT) ACCESS(ALTER)
RAC PERMIT DIRMAINT.1DF CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT DIRMAINT.2DF CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT DIRMAINT.150 CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
```

Reader DAC

If the reader DAC is enabled on your system:

Step 1. Authorize all users to send files to the DIRMAINT machine's reader. If there is already a DIRMAINT VMRDR profile defined, alter it, by entering:

```
RAC RALTER VMRDR DIRMAINT UACC(UPDATE)
```

If this profile is not defined, enter:

```
RAC RDEF VMRDR DIRMAINT UACC(UPDATE)
```

Step 2. If DirMaint has not been made exempt from reader DAC, each user should also authorize the DIRMAINT machine to send files back to the user's reader by entering:

```
RAC RDEFINE VMRDR <acigroup.>vmuserid UACC(NONE)
RAC PERMIT <acigroup.>vmuserid CLASS(VMRDR) xxxxxxxx ACCESS(UPDATE)
```

Where:

xxxxxxxx
Identifies the user ID of the DirMaint service machine.

Note: This includes the DATAMOVE and DIRMSAT machines.

Step 3. If the DATAMOVE and DIRMSAT machines have not been made exempt from reader DAC, the DirMaint support staff user IDs should authorize the DATAMOVE and DIRMSAT service machines by entering:

```
RAC PERMIT <acigroup.>vmuserid CLASS(VMRDR) ID(datamove) ACCESS(UPDATE)
RAC PERMIT <acigroup.>vmuserid CLASS(VMRDR) ID(dirmsat) ACCESS (UPDATE)
```

Where:

datamove
Identifies the user ID of the DATAMOVE service machine.

dirmsat
Identifies the user ID of the DIRMSAT service machine.

Enabling Mandatory Access Control

If your system is intended to meet class B1 TCB criteria you must enable your system for MAC.

Example—Defining the DirMaint Service Machines: If MAC is enabled, you must define all DirMaint service machines (DIRMAINT, DATAMOVEs, and DIRMSATs) to RACF with a SECLABEL of SYSHIGH by entering:

```
RAC PERMIT SYSHIGH CLASS(SECLABEL) ID(xxxxxxxx) ACCESS(READ)
RAC ALTUSER xxxxxxxx SECLABEL(SYSHIGH)
```

Where:

xxxxxxxx
Identifies the user ID of the service machine.

There are two forms of MAC that affect the DirMaint service machine. You may enable one or the other, or both, even if your system is not intended as a TCB.

Minidisk MAC

If minidisk MAC is enabled on your system:

Step 1. Define all of the minidisks owned by the DirMaint machines (P748XE4M, DIRMAINT, DATAMOVEs, and DIRMSATs) with the appropriate SECLABEL. The disks needed by general users must be defined as SYSLOW. Enter:

```
RAC RALTER VMMDISK P748XE4M.11F SECLABEL(SYSLOW)
RAC RALTER VMMDISK P748XE4M.29D SECLABEL(SYSLOW)
RAC RALTER VMMDISK P748XE4M.29E SECLABEL(SYSLOW)
RAC RALTER VMMDISK P748XE4M.41F SECLABEL(SYSLOW)
```

The use of optional national language Help files also must be defined as SYSLOW disks. Enter:

```
RAC RALTER VMMDISK P748XE4M.xxx SECLABEL(SYSLOW)
RAC RALTER VMMDISK P748XE4M.xxx SECLABEL(SYSLOW)
```

The remaining product code disks may be assigned a SECLABEL of your choice. Enter:

```
RAC RALTER VMMDISK P748XE4M.491 SECLABEL(xxxxxxxx)
RAC RALTER VMMDISK P748XE4M.492 SECLABEL(xxxxxxxx)
```

Where:

xxxxxxxx
Specifies the SECLABEL you have chosen.

Step 2. The DirMaint service machine data disks should all be given a SECLABEL of SYSHIGH. Enter:

```
RAC RALTER VMMDISK DIRMAINT.155 SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.1FA SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.1DF SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.2DF SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.1AA SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.2AA SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.1DB SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.1DE SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.15D SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMAINT.2DB SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DATAMOVE.155 SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DATAMOVE.1FA SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DATAMOVE.1AA SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DATAMOVE.2AA SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMSATx.155 SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMSATx.1FA SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMSATx.1AA SECLABEL(SYSHIGH)
RAC RALTER VMMDISK DIRMSATx.2AA SECLABEL(SYSHIGH)
```

Note: If you have multiple DATAMOVE or DIRMSAT service machines, you must assign a SECLABEL to all the data disks.

Reader MAC

If reader MAC is enabled on your system you must allow the DirMaint service machines (DIRMAINT, DATAMOVEs, and DIRMSATs) to send spool files to users with a SECLABEL other than SYSHIGH. The most commonly used SECLABEL is SYSLOW. Enter:

```
RAC PERMIT SYSLOW CLASS(SECLABEL) ID(xxxxxxx) ACCESS(READ)
RAC SETROPTS RACLIST(SECLABEL) REFRESH
```

Where:

xxxxxxxx
Identifies the user ID of the service machine.

Improving Performance with RACF

On a VM/ESA Feature System: when DIRMAINT is heavily used you can improve performance by adding the P748XE4M disks needed by general users to the global minidisk table in HCPRWA. Add:

```
GLBLDSK  USERID=P748XE4M,VADDR=11F,SCOPE=GLOBAL
GLBLDSK  USERID=P748XE4M,VADDR=29D,SCOPE=GLOBAL
GLBLDSK  USERID=P748XE4M,VADDR=29E,SCOPE=GLOBAL
GLBLDSK  USERID=P748XE4M,VADDR=41F,SCOPE=GLOBAL
```

Note: This form of global disk function is not available for VM/ESA 370 feature environments. For more information, see *RACF/VM Macros and Interfaces*.

On a VM/ESA 370 Feature System: and for the VM/ESA systems where you cannot change the HCPRWA minidisk table, or simply prefer not to rebuild the CP nucleus now, you have the alternative of adding these disks to the global access checking table. Enter:

```
RAC SETROPTS GLOBAL(VMMDISK)
RAC RDEFINE GLOBAL VMMDISK
RAC RALTER GLOBAL VMMDISK ADDMEM(P748XE4M.11F/READ)
RAC RALTER GLOBAL VMMDISK ADDMEM(P748XE4M.29D/READ)
RAC RALTER GLOBAL VMMDISK ADDMEM(P748XE4M.29E/READ)
RAC RALTER GLOBAL VMMDISK ADDMEM(P748XE4M.41F/READ)
RAC SETROPTS GLOBAL(VMMDISK) REFRESH
```

Note: The exact sequence of commands and command syntax may vary from release to release of RACF.

Notice: do NOT use either method to add any disk to the global access list if any of the these conditions are true:

1. The disk has a SECLABEL, unless that SECLABEL is SYSLOW.
2. The disk has UACC other than READ.
3. Any user is explicitly permitted to the disk for ACCESS(NONE).
4. The disk profile specifies either:
   - AUDIT(SUCCESS(READ))
   - AUDIT(FAILURE(READ))
   - AUDIT(ALL(READ))

For more information, see *RACF/VM Security Administration Guide*.

Appendix B. Tuning DirMaint Performance

There are three types of DirMaint performance tuning that can be done:

- Optimizing the user machine entries in the CONFIG* DATADVH file(s).
- Optimizing the service machine entries in the CONFIG* DATADVH file(s).
- Issuing CP privileged command options

Optimizing the User Machine

DirMaint functions performed in the user's virtual machine can be improved by following the suggestions described below.

- Split the CONFIG DATADVH file into multiple files. Keep the user machine related entries in the CONFIG DATADVH file on the user interface disk, 11F and 21F. Move the service machine related entries in to a CONFIGA DATADVH file on the service machine program disks, 191 and 192. This action will benefit all DirMaint users.
- Minimize the number of entries in the REQUIRED_USER_FILE list. At least one entry is required; the DVHCMD EXEC is recommended. Having an entry for every part of the DirMaint product located on the interface disk, 11F and 21F may help with problem diagnosis by making error messages more specific when a problem occurs because of a missing file. However, this error checking process decreases DirMaint command performance slightly in a user's virtual machine. This action will benefit all DirMaint users.
- Maximize the number of entries in the LOADABLE_USER_FILE list. Reading frequently used files into storage takes time. General users, who may typically issue a DIRM PW command and a DIRM REVIEW command every month on the average, won't be affected by this. But users who have many DirMaint commands to issue will save time by making all parts resident, reading the files only once for the entire group of commands.
- Encourage general users who issue many DirMaint commands to submit them using DIRM BATCH. If the commands are not suited for batch processing, the general user should use the DIRM EXECLOAD command to make frequently used DirMaint user machine routines memory resident when they begin their DirMaint work. They should then issue a DIRM EXECDROP command at the end of the DirMaint work.
- Encourage your system administration personnel (especially those that issue many DirMaint commands daily) to put a `EXEC DIRMAINT EXECLOAD` command into their PROFILE EXEC.
- If you have a large administration staff, you may want to consider installing the user machine routines into a shared segment. For more information see, *z/VM: Planning and Administration*.
- Remove comments lines, beginning with a slash, from the CONFIG* DATADVH file(s).

Optimizing the DirMaint Service Machines

Functions performed in the DirMaint service machines can be improved by following the suggestions described below.

- Split the CONFIG DATADVH file into multiple files. Keep the user machine related entries in the CONFIG DATADVH file on the user interface disk (11F, 21F). Move the service machine related entries in to a CONFIGA DATADVH file on the service machine program disks (191, 192).
- Minimize the number of entries in the following lists:
  - REQUIRED_SERVER_FILE
  - REQUIRED_DIRMAINT_FILE
  - REQUIRED_DATAMOVE_FILE
  - REQUIRED_DIRMSAT_FILE

  While this checking is only done once, at initialization time, these additional records must be scanned and skipped every time the CONFIG* DATADVH file(s) are read.
- List all of the following common parts on LOADABLE_SERV_FILE entries:
  - DVHWAIT
  - DVHWAKE
  - DVHWAKE3
  - DVHRDR
  - DVHDF8
  - DVHRQST
  - DVHFNDCS
  - DVHCEXIT
  - DVHMSG
  - DVHXLF
  - DVHESMLR
  - DVHRACLR
  - DVHSHUT

  List the DATAMOVE specific part (DVHDMCTL) and the DIRMSAT specific part (DVHDSCTL) on the LOADABLE_DATAMOVE_FILE and LOADABLE_DIRMSAT_FILE entries. List all executable DirMaint parts; file names beginning DVH and file types of EXEC, MODULE, REXX, or XEDIT not already on a LOADABLE_xxxxxxxx_FILE entry or a LOADABLE_DIRMAINT_FILE entry.
- If your installation is a large processing center with many DATAMOVE and DIRMSAT machines running in a CSE cluster, you may want to consider installing all of the LOADABLE_xxxxxxxx_FILE entries into a shared segment. For more information, see the *z/VM: Planning and Administration*.
- Use existing system facilities for disaster recovery, rather than exploit DirMaint's redundancy capabilities. Don't use the 2AA disk if the 1AA disk is backed up nightly. Don't use the 2DF disk if the 1DF disk is backed up nightly. Don't use the MESSAGE_LOGGING_FILETYPE if you are running with an ESM and recording DirMaint activity in the ESM audit log. Use the MESSAGE_LOGGING_FILTER_EXIT and ESM_LOG_FILTER_EXIT entries to avoid recording unnecessary messages in the TRANSLOG file or in the ESM audit log.
- Exploit tailoring options that improve both performance and usability on your system. For example, use UPDATE_IN_PLACE= YES on your system.

- Avoid using options whose performance cost on your system outweighs the usability benefits for your system.
  - Avoid use of SORT_BY_DEVICE_ADDRESS= YES.
  - Use DASD_ALLOCATE= FIRST_FIT rather than EXACT_FF.
  - If privacy of residual data is not usually a concern on your system, use DISK_CLEANUP= NO to avoid the time to FORMAT deleted minidisks.

    If privacy of residual data is a concern on your system, use DISK_CLEANUP= YES in the CONFIG* DATADVH file(s). This is required for a class C1, C2, or B1 TCB system.

    When DISK_CLEANUP= YES, request the administration staff to explicitly specify the CLEAN or NOCLEAN option on DMDISK or PURGE commands to avoid the time it takes to check for overlapping minidisks.

    Note: The CMDISK command does not support the CLEAN or NOCLEAN option.
- Experiment with options that have trade-offs that could fall either way.
  - Try ONLINE= IMMED; the WRK_UNIT_ONLINE setting is irrelevant.
  - Try ONLINE= SCHED with WRK_UNIT_ONLINE= YES.
  - Try ONLINE= SCHED with WRK_UNIT_ONLINE= NO.

  Adjust queue sizes to hold all work that typically arrives between directory updates.
  - With ONLINE= IMMED or with WRK_UNIT_ONLINE= YES, try the IBM supplied default queue sizes:

    ```
    DM_MAXIMUM_RETRIES= 10
    MAXIMUM_UNASSIGNED_WORKUNITS= 10
    ```
  - With ONLINE= SCHED and WRK_UNIT_ONLINE= NO

    ```
    ==/==/== +01:00:0 DIRECT
    ```

    specified in the DIRMAINT DATADVH file, and approximately 30 DASD management commands (AMDISK, CMDISK, DMDISK) arriving per hour, try increasing the queue sizes:

    ```
    DM_MAXIMUM_RETRIES= 50
    MAXIMUM_UNASSIGNED_WORKUNITS= 50
    ```
- Customize the DVHXPROF EXEC, an IBM-supplied sample exit routine to define a VFB-512 virtual disk in storage and format it as filemode A, as shown in Figure 34.

```
1  PURPOSE= RWS FM= A ACC= 255   V-disk copy of 155.
2  PURPOSE= SRV FM= C ACC= 291   V-disk copy of 191.
3  PURPOSE= USR FM= D ACC= 31F   V-disk copy of 11F.
4  PURPOSE= PDF FM= E ACC= 3DF   V-disk copy of 1DF.
          PURPOSE= PDB FM= G ACC= 1DB
          PURPOSE= PTH FM= H ACC= 1AA
3  PURPOSE= SDF FM= J ACC= 1DF
       / PURPOSE= SDB FM= - ACC= --- not used.  PDB covered by nightly backup
       / PURPOSE= STH FM= - ACC= --- not used.  PTH covered by nightly backup
4  PURPOSE= abc FM= K ACC= 155
5  PURPOSE= def FM= L ACC= 191
6  PURPOSE= ghi FM= M ACC= 11F
          PURPOSE= SFA FM= Z ACC= 2FA   V-disk.
```

Figure 34. Copying all files from the 155 disk to the new V-disk

These notes will help you customize the DVHXPROF EXEC.

Step 1. Create the V-disks, as shown in **1** - **4**

Step 2. Copy files to the V-disks from the 155, 191, 11F, and 1DF disks by using the DVHXPROF EXEC, with nothing being copied to the 2FA disk, as shown in **3** - **6**

Step 3. Customize the DVHPROFA DIRMAINT, DVHPROFA DIRMSAT and DVHPROFM DATADVH files to access these disks.

For more information on the DVHPROFA DIRMAINT, see the "DVHPROFA DIRMAINT" on page 43.

Note: IBM recommends that the conventional minidisk files be backed up nightly, there is no protection of redundant conventional minidisks. However, there is a potential for loss of a few DirMaint transactions in the event of a system failure. Generally, this is accepted as a worthwhile risk to take in order to obtain the performance benefit of using V-disks.

- Remove comments lines, beginning with a slash, from the CONFIG* DATADVH file(s).
- Exploit the exit routines for performance. If you have a central administration staff and do not delegate use of the privileged commands, for example:

```
/* */ Exit 0
```

or

```
/* */
If WordPos(Userid(),'adminid1 adminid2 adminid3') ¬= 0
   Then Exit 0
   Else Exit 30
```

for some of the exit routines will bypass further authorization checking. Candidates for this include:

  - ACCOUNT_NUMBER_VERIFICATION_EXIT
  - DASD_AUTHORIZATION_CHECKING_EXIT
  - LINK_AUTHORIZATION_EXIT
  - LOCAL_STAG_AUTHORIZATION_EXIT

Setting CP Performance Options

SET QDROP OFF Option

This option eliminates the overhead in CP caused by scanning DirMaint page and segment tables. It also prevents resident pages from being put on the flush list whenever DirMaint enters an idle wait state.

The NOQ2 and NOQ3 options of the QDROP command improve the performance of a service machine like DirMaint. The NOQ2 option significantly reduces the erratic delays in DirMaint because of delays in Q2 and Q3.

This privileged CP command eliminates some paging overhead on virtual machines that frequently communicate with other virtual machines.

The format of the command is *SET QDROP userid OFF*.

The SET QDROP OFF option is not supported on VM/ESA or VM/XA systems.

SET FAVORED Option

This option prevents DirMaint from being dropped from the CP scheduler queue although in a wait state. This privileged CP command specifies the percentage of processor time that is used for the DirMaint server.

The format of the command is *SET FAVORed userid percentage*. The *percentage* value to supply depends on your particular system.

SET RESERVE Command

On systems with high paging load, DirMaint will very likely be paged out when an interrupt comes in. The SET RESERVE command, however, lets most DirMaint active pages remain in real storage.

Dispatching Priority

A priority can be specified on the USER control statement for DirMaint in the VM directory. CP uses the specified value, or the default value (64), to determine the DirMaint dispatching priority. The system operator can alter this value, for a given IPL, by using the CP SET command.

The format of the command is *SET PRIORITY userid level*. The priority *level* to supply depends on your particular system, but a value of 50 is probably a good start. (A priority of 50 is higher than what an *average* system user gets, 64.)

Note: This is an option only for the VM/ESA Feature 370 system.

SET QUICKDSP Command

The QUICKDSP designation is intended for selective use on virtual machines with critical response time requirements. The scheduler always moves a QUICKDSP user immediately into the dispatch list whenever it is ready to run, regardless of resource requirements and current system load. Indiscriminate use, therefore, increases response time overall and may severely affect maintenance of system storage.

QUICKDSP is generally provided for use by selected service virtual machines interacting with several other users, thus having stringent response time requirements. RSCS and IUCV applications are common examples.

For more information on these commands, see the *z/VM: CP Command and Utility Reference*.

Appendix C. DirMaint Configuration Data Files

This appendix provides a summary of each of the CONFIG* DATADVH entries provided by DirMaint Release 5.0. These data files provide information about the configuration of the DirMaint product. If there are multiple files, they are searched in reverse alphabetical order so that entries in CONFIG99 will override entries in the IBM supplied CONFIG default. Table 41 describes many of the CONFIG* DATADVH entries provided by DirMaint Release 5.0.

Table 41 (Page 1 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
ADD_COMMAND_PROCESSING= FULL or SHORT	NONE	This statement specifies whether LINK and MDISK directory statements in a directory entry being added are processed using full authorization checking, or if they are allowed to short cut any of the LINK and AMDISK authorization checks.	53
BACKUP REBUILD= CLUSTER DVHLINK <VCONTROL> | NONE	NONE	This statement controls the balance between the time taken to complete a BACKUP operation and the amount of cleanup needed.	47
CLASS LIMIT ON USER STATEMENT=	8 | 0 ... 32 | 0 ... 8	Specifies how many CP privilege classes may be included on the USER statement.	48
CLASS STATEMENT IN PROFILE CHECK =	NO or YES	Specifies whether DirMaint will do the additional checking to see if the included PROFILE contains a CLASS statement.	48
CYL0_BLK0_CLEANUP=	NO or YES	This entry supports the OBJECT REUSE policy. The value must be YES for TCB levels C1, C2, or B1.	50
DASD_ALLOCATE=	FIRST_FIT or EXACT_FF	Specifies which allocation algorithm to use for AUTOR, RBLK*, AUTOV, VBLK*, AUTOG, and GBLK* requests. FIRST_FIT is the faster choice, while EXACT_FF reduces fragmentation.	100
DATAMOVE_MACHINE=	*MachName MachNode SysAffin*	DirMaint DASD Management functions that require a CMS FORMAT, COPYFILE command, or both, may take a while to perform. So they are assigned to another service virtual machine for processing. Each machine must be identified on a DATAMOVE_MACHINE statement, along with the node ID within the complex where the DATAMOVE machine is running, and the system affinity that the DATAMOVE machine is authorized to process.	88

Table 41 (Page 2 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
DEFAULT_CMDLEVEL=	140A or 150A	This value determines which messages and command parsing files should be used when the user has not entered a DIRM GLOBALV CMDLEVEL command to select their own default CMDLEVEL. The preferred default for general users is 150A; for compatibility with DFSMS, DSO, IPF, NVAS, RACF, and other disconnected service machines running programs that entered DIRMAINT commands using R4 syntax, you may with to change it to 140A.	125
DEFAULT_CMDSET.*xxxx*=	The IBM-supplied default is *G*.	This value determines which privileges a user has if the user has not been explicitly authorized for specific privileges. A different default may be specified for each defined CMDLEVEL.	121
DEFAULT_SERVER_LANG=	Language identifier	This value determines the language used for messages sent to the DIRMAINT/DATAMOVE/DIRMSAT machine's console and to the broadcast list for service messages. If not specified, the default is *AMENG*. Messages to an individual user are sent in the user's requested language.	N/A
DIRECTXA_OPTIONS=	MIXED or MIXED NOMIXMSG	This value specifies the options used when the DIRECTXA command places the directory online. If you have a *clean* source directory for VM/ESA, leave this blank. If your source directory has been migrated from the VM/ESA 370 feature (or its VM/SP or VM/SP HPO predecessors) and contains a few 370 flavor directory statements, you may chose to use MIXED to assist you in completing your migration. If your directory contains too many 370 flavor directory statements, you may use MIXED NOMIXMSG to suppress the messages.	47
DISK_CLEANUP=	NO or YES	This entry supports the OBJECT REUSE policy. The value must be YES for TCB levels C1, C2, or B1.	50

Table 41 (Page 3 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
DISK_SPACE_THRESHHOLD	IBM supplied defaults are 75 and 90. However, you can use any number from 1 - 99, the first value must be less than the second value, the second value must be less than or equal to 1 - 99.	This value identifies the warning and shutdown limitation on DASD space usage.	49
DM_MAXIMUM_RETRIES=	Numeric value	If a DATAMOVE machine is unable to link to a minidisk (most likely because a user is linked to a disk for which a CMDISK command has been entered, or the directory change to transfer the minidisk to DATAMOVE has not been put online), the FORMAT/COPY/CLEAN request will be put onto the DATAMOVE machine's retry queue. The DM_MAXIMUM_RETRIES value determines the maximum size of this retry queue. (It has no effect on the number of times one request will be retried.) After DIRMAINT has been notified that this limit has been reached, DIRMAINT will not assign any more work to that particular DATAMOVE machine.	78
DVHWAIT BATCH INTERVAL=	*mm:ss* or *hh:mm:s*	Specifies how long DVHWAIT should delay to wait for other input when a BATCH job file is active. If hours are specified, granularity is 10 second intervals. The default is 1 second (00:01).	N/A
DVHWAIT CLUSTER INTERVAL=	*mm:ss* or *hh:mm:s*	Specifies how long DVHWAIT should delay while waiting for a DIRECTXA request to complete on the satellite systems within a CSE multiple system cluster. If hours are specified, granularity is 10 second intervals. The default is 15 seconds (00:15).	N/A
DVHWAIT IDLE INTERVAL=	*mm:ss* or *hh:mm:s*	Specifies how often DVHWAIT must *wakeup* to prevent the DIRMAINT servers from being forced off the system due to lack of activity. If hours are specified, granularity is 10 second intervals. The default is 5 minutes (05:00).	N/A

Table 41 (Page 4 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
ESM_LOG_FILTER_EXIT=	DVHXLF EXEC	This entry supports the AUDITING policy. As you review the entries in the ESM log, you may find many DirMaint messages that are of no interest to you. This entry suppresses future collection of these messages. If your system is intended to meet either class C2 or class B1 TCB criteria, you must obtain the approval of your DAA for any messages you choose to filter out.	52
ESM_LOG_RECORDING_EXIT=	DVHESMLR EXEC	This entry supports the AUDITING policy. Use of an ESM with application audit logging capability is required if your system is intended to meet either class C2 or class B1 TCB criteria.	51
ESM_PASSWORD_AUTHENTICATION_EXIT=	DVHDA0 MODULE	This entry supports the AUTHENTICATION policy. A non-blank value is required for TCB levels C2 or B1.	50
INTENDED_TCB_LEVEL=	D, C1, C2, or B1	This value identifies to an auditor or to your DAA the TCB criteria your system is intended to meet. This value is not checked by DirMaint release 5. The value you choose may impose restrictions upon the valid choices for the following security related parameters. Deviations from the requirements stated below require the approval of your auditor or DAA.	50
MAXIMUM_UNASSIGNED_WORKUNITS=	Numeric value	DirMaint DASD Management functions are queued for asynchronous processing by the DATAMOVE machine(s). The value specified for MAXIMUM_UNASSIGNED_WORKUNITS determines the maximum size of this queue. Too low a value results in DASD management commands being rejected because the queue is full while all DATAMOVE machines are busy. A value of 0 will completely disable all DASD management processing. Too high a value could result in problems not being noticed and reported to the support team for resolution in a timely manner.	101

Table 41 (Page 5 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
MDPW_INTERVAL=	warn expire	This determines how old a minidisk password may become before entering a WARNING period, and before entering the EXPIRED period. The first value must be less than the second value. The second value must be less than or equal to 373 (one year plus one week grace). Use of 0 0 disables checking. **Note:** Minidisk passwords of ALL never expire. DirMaint takes no action based on minidisk password expiration, but does flag them appropriately on the MDAUDIT report.	55
POSIX_UID_AUTO_RANGE=	low high	This entry specifies a UID range for use during automatic assignment of POSIX UIDs to users during DIRM ADD and DIRM POSIXINFO operations. The input parms consist of two integer values, the first represents the lower bound, the second represents the upper bound. Note there is a space between “low” and “high.”	52
MESSAGE_LOG_RETENTION_PERIOD=	*months*	This entry supports the AUDITING policy. A value of 3 months is suggested. This value may need to be adjusted up or down, depending on the amount of DirMaint activity on your system and the size of the minidisk you have allocated for the transaction history files.	51
MESSAGE_LOGGING_FILETYPE= MESSAGE_LOGGING_FILTER_EXIT=	TRANSLOG DVHXLF EXEC	These entries support the AUDITING policy. They are required for TCB level C1. Your logging filter must be approved by your auditor or DAA.	51
ONLINE=	OFFLINE or SCHED or IMMED	This value determines the initial value of the ONLINE CONTROL file. After DIRMAINT has been initialized, the value be changed using the DIRM OFFLINE and DIRM ONLINE commands.	46

Table 41 (Page 6 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
PURGE_COMMAND_PROCESSING= FULL or SHORT	NONE	This statement specifies whether LINK and MDISK directory statements in a directory entry being added are processed using full authorization checking, or if they are allowed to short cut any of the LINK and AMDISK authorization checks. **Note:** This entry could also affect the OBJECT REUSE policy. Use of the `PURGE_COMMAND_PROCESSING= SHORT` will bypass the disk cleanup. Both options must be set or defaulted to FULL, for the TCB levels C1, C2, or B1.	53
PW_INTERVAL_FOR_GEN=	0 0	This identifies how old a *general* user's logon password may become before entering a WARNING period, and before entering the EXPIRED period. The first value must be less than the second value. The second value must be less than or equal to 373 (one year plus one week grace). Use of 0 0 disables checking.	54
PW_INTERVAL_FOR_PRIV=	0 0	This identifies how old a *privileged* user's logon password may become before entering a WARNING period and before entering the EXPIRED period. The first value must be less than the second value. The second value must be less than or equal to 373 (one year plus one week grace). Use of 0 0 disables checking. Blanks cause privileged users' passwords to be treated the same as general users.	54

Table 41 (Page 7 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
PW_INTERVAL_FOR_SET=		Unless otherwise specified, a password that is set using the ADD, CHNGID or SETPW commands will be valid for the full duration specified on the respective PW_INTERVAL_FOR_GEN or PW_INTERVAL_FOR_PRIV statements. The PW_INTERVAL_FOR_SET values specify a shorter expiration period when the password has been changed by one of these commands. The first value applies to *general* users. The second applies to *privileged* users. The values must be less than the respective expiration periods. The recommended minimum value is 1. The maximum suggested value is equal to the difference between the expiration period and the warning period. **Notes:** 1. All users are *general* users unless the system CHECK_USER_PRIVILEGE_EXIT is in use and identifies the users in question as *privileged*. 2. Logon passwords of AUTOONLY, LBYONLY, NOLOG, and NOPASS never expire.	55

Table 41 (Page 8 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
PW_LOCK_MODE=	MANUAL or AUTOMATIC	This determines whether DIRMAINT automatically generates and sends password expiration notices and changes expired passwords to NOLOG (if AUTOMATIC), or if this must be done by the administrator (if MANUAL). **Notes:** 1. PW_WARN_MODE is also AUTOMATIC. 2. The PW_INTERVAL_FOR_GEN and PW_INTERVAL_FOR_PRIV entries specify reasonable periods for your installation. 3. Disconnected service machines have a surrogate identified in the PWMON CONTROL file to receive their password notices. 4. Critical system user IDs, OPERATOR, DIRMAINT, MAINT, are listed in the PWMON CONTROL file as being exempt from lockout.	55
PW_MIN_LENGTH=	3	This value is used by the IBM-supplied exits for PASSWORD_SYNTAX_CHECKING_USER_EXIT and PASSWORD_SYNTAX_CHECKING_EXIT. If your installation has modified these exits, or is not using them, then you may delete this value. For more information, see the *Directory Maintenance VM/ESA: Command Reference*.	N/A
PW_NOTICE_PRT_CLASS=	One letter from A to Z or NONE	This identifies the spool file print class to be used for printed password warning and expiration notices. A value of NONE indicates that password notices will not be printed.	55

Table 41 (Page 9 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
PW_NOTICE_RDR_CLASS=	One letter from A to Z or NONE	This identifies the spool file reader class to be used for password warning and expiration notices sent to a user's reader. A value of NONE indicates that password notices will not be sent. **Note:** A value other than NONE will be treated the same as class A if DirMaint is running on a system prior to CMS level 11 on VM/ESA 1.2.2 (CMS level 7 on VM/ESA 1.1.5 370 feature or CMS level 10 on VM/ESA 1.2.1).	55
PW_REUSE_HASHING_EXIT	None	The routine hashes the user's password for storage in the password history file. The file type may be either EXEC or MODULE. The IBM supplied default is DVHHASH MODULE. If not specified, the passwords will be stored in the history file as hexadecimal digits.	122
PW_REUSE_INTERVAL	None	This identifies how long an entry is kept in the password history file. It may be either a time period with a DAYS suffix, or a count with no suffix. The IBM supplied default is 365 DAYS.	122
PW_WARN_MODE=	MANUAL or AUTOMATIC	This determines whether DIRMAINT automatically generates and sends password warning notices (if AUTOMATIC), or if this must be done by the administrator (if MANUAL).	55
RUNMODE=	TESTING or OPERATIONAL	This value determines whether directory source changes are actually made (if OPERATIONAL) or discarded (if TESTING). For safety, the IBM-supplied default is TESTING.	46

Table 41 (Page 10 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
SATELLITE_SERVER=	*userid nodeid cpuid*	The DIRMAINT service machine maintains a single object directory, usually on the system residence volume. For redundancy in case of hardware errors with that volume, a satellite machine can be used to maintain a second object directory on a different volume. In a multiple system CSE cluster, one or two separate object directories must be maintained on each system - also using satellite servers. Each satellite server must be defined on a SATELLITE_SERVER statement, along with the node ID within the complex where the satellite server is running, and the CPU ID to be used to associate that satellite server with the correct DIRECTORY statement and system affinity in the source directory file.	81
SHUTDOWN_LOGOFF_THRESHHOLD=	2, 3, or 4	This value specifies the number of error induced shutdown conditions that may be encountered before the service machine logs itself off, if running disconnected.	49
SHUTDOWN_MESSAGE_FAILURE=	LOGOFF or REIPL	This value identifies the action to be taken if the failure causing the shutdown occurred in the message handler. If your system is intended to meet TCB criteria, the correct value is LOGOFF. Otherwise you may choose either LOGOFF or REIPL for this value. (Because of the TCB relevance, the SHUTDOWN_MESSAGE_FAILURE entry is located with the other security related configuration parameters.)	52
SHUTDOWN_REIPL_COMMAND=	CP IPL CMS PARM AUTOCR	This value specifies the CP command to be entered in order to accomplish the re-IPL. The AUTOCR keyword is required. Any other keywords that are valid on the IPL command may also be used if appropriate for your system environment. For example: `CP IPL 190 PARM AUTOCR NOSPROF FILEPOOL SERVERX`	49

Table 41 (Page 11 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
SHUTDOWN_RESET_THRESHHOLD=	s_r_t	This value specifies the number of commands that must be successfully processed after an error induced shutdown before the logoff counter is reset. The s_r_t must be >= 1. A *successfully processed command* is one that does not result in a shutdown condition, but does not necessarily result in a zero return code. The minimum recommended value is 2; the maximum recommended value is 5. **Notes:** 1. Shutdown events are handled in pairs. The first shutdown, or any odd numbered shutdown, causes a re-IPL, and the failing command is retried. The second shutdown, or any even numbered shutdown is probably the retry of the failing command. (The lower the value for the RESET threshold, the more likely this is true; a RESET value of 1 ensures this.) Even numbered shutdowns cause either a re-IPL or a LOGOFF after purging the command from the retry queue. 2. After the specified number of shutdown events have occurred, a CP LOGOFF command is entered if running disconnected. If running connected, the system will continue to re-IPL.	49
SORT_BY_DEVICE_ADDRESS=	NO or YES	The SORT_BY_DEVICE_ADDRESS value specifies whether or not the device statements in each user directory are maintained in sorted order by device address. Specifying YES increases the time and storage requirements for all updates to directory entries (either PROFILE or USER) containing device statements.	46

Table 41 (Page 12 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
SORT_COMMENTS_WITH_STATEMENTS=	NO or YES	The SORT_COMMENTS_WITH_STATEMENTS value specifies whether comments in a directory with SYSAFFIN statements are kept with the statement they follow (YES) or whether they drop to the bottom of the directory entry (NO). For optimum performance and minimum directory size, specify NO. The default is YES, for minimum impact to customer supplied arrangement of data. This option has no affect on directory entries that do not contain a SYSAFFIN statement, where the action is always YES.	46
SORT_DIRECTORY=	NO or YES	This value specifies whether the USER DIRECT file is to be maintained in sorted order. Specifying YES increases the time and storage requirements	46
SPOOL_CONSOLE=	Spool console command text	This value identifies the USER id to receive the console spool files from the various DirMaint service machines. The data following the = is usually the command syntax after the CP SPOOL CONSOLE command. **Note:** When the DIRM GETCONSOLE command is issued, a copy of the spool file is sent to the command issuer and a copy is sent to the user ID identified. If the user ID's are the same, only one copy is sent. The same action will occur if the DIRM GETCONSOLE command is to retrieve a spool file residing in the virtual printer.	54
SPOOL_FILE_SECLABEL=	SYSLOW	This entry supports the ACCESS CONTROL policy. A non-blank value is highly recommended for TCB level B1.	50
SRCUPDATE=	DISABLED or NOP	This value determines whether DIRMAINT disables itself from accepting directory update commands from users each time DIRMAINT is restarted. If DISABLED, a DISABLE CONTROL file is created; otherwise this statement is ignored (NOP). The DISABLE CONTROL file will be erased by a DIRM ENABLE command.	46

Table 41 (Page 13 of 13). CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
UPDATE_IN_PLACE=	YES or NO	This value controls whether DIRMAINT will attempt to use DIAGNOSE code X'84' to put directory changes online. be changed using the DIRM OFFLINE and DIRM ONLINE commands.	47
WRK_UNIT_CLEANUP=	ERASE or RENAME	This value controls whether the WORKUNIT files will be erased or renamed to WORKSAVE after the completion of the DASD management commands. In the event of a failure, they will be renamed to WUCFFAIL in either case.	48
WRK_UNIT_ONLINE=	NO or YES	This value controls whether DIRMAINT will enter DIRECT or DIRECTXA commands in the middle of a work unit. A work unit is a group of DIRMAINT commands created by DIRMAINT itself in response to a DASD management request that requires use of a DATAMOVE service machine to complete the request.	47

Language Dependent Configuration Entries

The occurrence of *lang* must be replaced in the files listed in this section with one of the following language identifiers:

AMENG	American English
UCENG	Uppercase English
KANJI	Japanese
SAPI	Synchronous Application Programming Interface

The following identifies the language dependent files used for the user's active language:

```
lang_BATCH_HEADER_140A= DVHBHEAD DATAADVH             Batch header file
lang_BATCH_HEADER_150A= DVHBHEAD DATAADVH
lang_COPYRIGHT_NOTICE=  DVHCOPYR DATAADVH             Copyright notice
lang_HELP_140A=         DIRM     HELPDIRM             Help files
lang_HELP_150A=         DVHlang HELPADVH
lang_MENU_DEFS_150A=    DVHMENUS DATAADVH             Menu data file
lang_USER_MSGS_140A=    LCLAUSER MSGADVH              Message repositories
lang_USER_MSGS_140A=    140AUSER MSGADVH
lang_USER_MSGS_140A=    150AUSER MSGADVH
lang_USER_MSGS_150A=    LCLAUSER MSGADVH
lang_USER_MSGS_150A=    150AUSER MSGADVH
```

The following entries define the files for those user languages not otherwise listed above. The **.....** indicates that these are the default. The defaults are set to mixed case American English.

```
.....BATCH_HEADER_140A= DVHBHEAD DATAADVH
.....BATCH_HEADER_150A= DVHBHEAD DATAADVH
.....HELP_140A=         DIRM     HELPDIRM
.....HELP_150A=         DVHAMENG HELPADVH
.....MENU_DEFS_150A=    DVHMENUS DATAADVH
.....USER_MSGS_140A=    LCLAUSER MSGADVH
.....USER_MSGS_140A=    140AUSER MSGADVH
.....USER_MSGS_140A=    150AUSER MSGADVH
.....USER_MSGS_150A=    LCLAUSER MSGADVH
.....USER_MSGS_150A=    150AUSER MSGADVH
```

The following identifies the language dependent files that are common to both the user's and the server's machines.

```
COMMANDS_140A=    LCLCMDS  DATADVH
COMMANDS_140A=    140CMDS  DATADVH
COMMANDS_150A=    LCLCMDS  DATADVH
COMMANDS_150A=    150CMDS  DATADVH
  PARSER_140A=    DVHADZ   EXEC
  PARSER_150A=    DVHAEZ   EXEC
```

The following identifies the language dependent files used for the server's active language:

```
lang_MDISK_AUDIT_NOTICES=   AUTOMAIL DATAADVH     Mdisk audit notice file
lang_PW_NOTICE_LOCK_OTHERW= PWLOTHER DATAADVH     Password notice files
lang_PW_NOTICE_LOCK_OTHERL= PWLOTHER DATAADVH
lang_PW_NOTICE_LOCK_NOLOCK= PWLNOLCK DATAADVH
lang_PW_NOTICE_LOCK_LOCKED= PWLOCKED DATAADVH
lang_PW_NOTICE_WARN_OTHER=  PWWOTHER DATAADVH
lang_PW_NOTICE_WARN_NOLOCK= PWWNOLCK DATAADVH
lang_PW_NOTICE_WARN_B4LOCK= PWWB4LCK DATAADVH
lang_SERV_MSGS_140A=    LCLASERV MSGADVH          Message repositories
lang_SERV_MSGS_140A=    140ASERV MSGADVH
lang_SERV_MSGS_140A=    150ASERV MSGADVH
lang_SERV_MSGS_150A=    LCLASERV MSGADVH
lang_SERV_MSGS_150A=    150ASERV MSGADVH
```

The following entries define the files for those user languages not otherwise listed above. The **.....** indicates that these are the default. The defaults are set to mixed case American English.

```
....._SERV_MSGS_140A=    LCLASERV MSGADVH         Message repositories
....._SERV_MSGS_140A=    140ASERV MSGADVH
....._SERV_MSGS_140A=    150ASERV MSGADVH
....._SERV_MSGS_150A=    LCLASERV MSGADVH
....._SERV_MSGS_150A=    150ASERV MSGADVH
....._PW_NOTICE_WARN_OTHER=  PWWOTHER DATAADVH    Password notice files
....._PW_NOTICE_WARN_NOLOCK= PWWNOLCK DATAADVH
....._PW_NOTICE_WARN_B4LOCK= PWWB4LCK DATAADVH
....._PW_NOTICE_LOCK_OTHERW= PWLOTHER DATAADVH
....._PW_NOTICE_LOCK_OTHERL= PWLOTHER DATAADVH
....._PW_NOTICE_LOCK_NOLOCK= PWLNOLCK DATAADVH
....._PW_NOTICE_LOCK_LOCKED= PWLOCKED DATAADVH
....._MDISK_AUDIT_NOTICES=   AUTOMAIL DATAADVH    Mdisk audit notice file
```

The following identifies how to route files to other systems within a local multiple-system cluster.

Format:

```
FROM= *  DEST= *  S= *  T= *
```

Note: Cluster support is enabled by default.

FROM, DEST, and T are node IDs as determined from an IDENTIFY command; the DEST and T values are usually the same. S is the user ID of the local network service machine. FROM * = anywhere, DEST * = wherever the DirMaint server is running as recorded in the WHERTO DATADVH file, S * = the network server obtained using IDENTIFY, T * = same as DEST.

The defaults are appropriate in most cases without shared spool files. With shared spool files, the S value is usually specified as the user ID of the DirMaint server. For example:

```
FROM= CORPVM1  DEST= CORPVM2  S= DIRMAINT  T= *
FROM= CORPVM2  DEST= CORPVM1  S= DIRMAINT  T= *
```

For performance reasons, you may chose to establish a dedicated network for the local cluster. In this case, the SPOOL value must identify the special server for the cluster, and TAG1 must provide the correct tag data for that particular server.

Here is an example of using a Spool File Bridge:

```
FROM= VM370     DEST= VMESA121 S= SFBRIDGE  T= VMESA121
FROM= VM370     DEST= VMESA122 S= SFBRIDGE  T= VMESA122
FROM= VM370     DEST= VMESA210 S= SFBRIDGE  T= VMESA210
FROM= VMESA121  DEST= VM370    S= SFBRIDGE  T= VM370
FROM= VMESA121  DEST= VMESA122 S= DIRMAINT  T= VMESA122
FROM= VMESA121  DEST= VMESA210 S= DIRMAINT  T= VMESA210
FROM= VMESA122  DEST= VM370    S= SFBRIDGE  T= VM370
FROM= VMESA122  DEST= VMESA121 S= DIRMAINT  T= VMESA121
FROM= VMESA122  DEST= VMESA210 S= DIRMAINT  T= VMESA210
FROM= VMESA210  DEST= VM370    S= SFBRIDGE  T= VM370
FROM= VMESA210  DEST= VMESA121 S= DIRMAINT  T= VMESA121
FROM= VMESA210  DEST= VMESA122 S= DIRMAINT  T= VMESA122
```

Each system in a multiple system cluster must have a satellite server machine defined to update the object directory on that system. Optionally, a second server may be defined on each system to maintain a second object directory for backup - even if the system is not part of a CSE multiple system cluster.

```
SATELLITE_SERVER= DIRMSAT1 VMESA121  ESA121A
SATELLITE_SERVER= DIRMSAT2 VMESA121  ESA121B
SATELLITE_SERVER= DIRMSAT3 VMESA122  ESA122A
SATELLITE_SERVER= DIRMSAT4 VMESA122  ESA122B
SATELLITE_SERVER= DIRMSAT5 VMESA210  ESA210A
SATELLITE_SERVER= DIRMSAT6 VMESA210  ESA210B
SATELLITE_SERVER= DIRMSAT7 VM370A    VM370A
SATELLITE_SERVER= DIRMSAT8 VM370B    VM370B
```

The following identifies how to route files to other remote systems beyond the local cluster.

Format:

```
FROM= *  DEST= *  S= *  T= *  U= DIRMAINT
```

Note: Network support is disabled by default.

FROM and T are node IDs as determined from an IDENTIFY command; DEST is your nickname for the T node. S is the user ID of the local network service machine, or an * if this is to be determined from the IDENTIFY command. FROM * = anywhere, DEST * = wherever specified, S * = the network server obtained using IDENTIFY, T * = same as DEST, U * = DIRMAINT. For example:

```
FROM= ABCVM  DEST= XYZ1  S= *  T= XYZVM1  U= DIRMXYZ1
FROM= ABCVM  DEST= XYZ2  S= *  T= XYZVM2  U= DIRMR5
```

The following identifies the service machine files that are made resident at initialization time, and reloaded with RLDCODE; for more information, the location in this guide has been provided for your reference:

File	Page
LOADABLE_SERV_FILE= *filename filetype*	44
LOADABLE_DATAMOVE_FILE= *filename filetype*	79
LOADABLE_DIRMSAT_FILE= *filename filetype*	84
LOADABLE_DIRMAINT_FILE= *filename filetype*	56

The following identifies the files that must be present for the service machine to run correctly.

```
REQUIRED_SERV_FILE= filename filetype
REQUIRED_DATAMOVE_FILE= filename filetype
REQUIRED_DIRMSAT_FILE= filename filetype
REQUIRED_DIRMAINT_FILE= filename filetype
```

The following identifies the service machine exit routines. Refer to Chapter 9, “Exit Routines” on page 141 for more information.

```
REQUEST_BEFORE_PARSING_EXIT=          DVHXRC   EXEC
REQUEST_BEFORE_PROCESSING_EXIT=       DVHXRB   EXEC
REQUEST_AFTER_PROCESSING_EXIT=        DVHXRA   EXEC
FOR_AUTHORIZATION_CHECKING_EXIT=      DVHXFA   EXEC
ACCOUNT_NUMBER_VERIFICATION_EXIT=     DVHXAV   EXEC
ACCOUNT_NUMBER_NOTIFICATION_EXIT=     DVHXAN   EXEC
PASSWORD_RANDOM_GENERATOR_EXIT=       DVHPXR   EXEC
PASSWORD_SYNTAX_CHECKING_EXIT=        DVHPXV   EXEC
PASSWORD_CHANGE_NOTIFICATION_EXIT=    DVHXPN   EXEC
MINIDISK_PASSWORD_CHECKING_EXIT=      DVHXMP   EXEC
MINIDISK_PASSWORD_NOTIFICATION_EXIT=  DVHXMN   EXEC
DASD_AUTHORIZATION_CHECKING_EXIT=     DVHXDA   EXEC
DASD_OWNERSHIP_NOTIFICATION_EXIT=     DVHXDN   EXEC
USER_CHANGE_NOTIFICATION_EXIT=        DVHXUN   EXEC
CHECK_USER_PRIVILEGE_EXIT=            DVHXCP   EXEC
LINK_AUTHORIZATION_EXIT=              DVHXLA   EXEC
LINK_NOTIFICATION_EXIT=               DVHXLN   EXEC
PW_NOTICE_PRT_EXIT_EXIT=              DVHXPP   EXEC
DATAMOVE_NONCMS_COPYING_EXIT=         DVHDXP   EXEC
LOCAL_STAG_AUTHORIZATION_EXIT=        DVHXTA   EXEC
BACKUP_TAPE_MOUNT_EXIT=               DVHXTAPE EXEC
MULTIUSER_VERIFICATION_EXIT=          DVHXMU   EXEC
```

Note: A *required* exit routine must be defined, and must exist; although it may be given any valid unique file name and may be tailored.

The following identifies the user files that are made resident or non-resident by the EXECLOAD and EXECDROP commands.

```
LOADABLE_USER_FILE= filename filetype
```

The following identifies the files that must be present for the user's virtual machine to correctly enter any DIRMAINT command.

```
REQUIRED_USER_FILE= filename filetype
```

The following identifies the user exit routines. For more information, see Chapter 9, “Exit Routines” on page 141.

```
COMMAND_BEFORE_PARSING_USER_EXIT=    DVHCXC   EXEC
COMMAND_BEFORE_PROCESSING_USER_EXIT= DVHCXB   EXEC
COMMAND_AFTER_PROCESSING_USER_EXIT=  DVHCXA   EXEC
PASSWORD_RANDOM_GENERATOR_USER_EXIT= DVHPXR   EXEC          (Required)
PASSWORD_SYNTAX_CHECKING_USER_EXIT=  DVHPXV   EXEC
PASSWORD_NOTIFICATION_USER_EXIT=     DVHPXA   EXEC
```

Note: A *required* exit routine must be defined, and must exist; although it may be given any valid unique file name and may be tailored. The COMMAND_BEFORE_PROCESSING exit is *required* for compatibility with 1.4

(SYS processing on ADD or AMDISK commands, and for compatibility with 1.4 BATCH processing without a file identification being supplied. The PASSWORD_RANDOM_GENERATOR exit is required if random passwords are to be generated automatically.

Appendix D. WAKEUP Command

The WAKEUP command controls the startup of an event-driven machine (typically a disconnected service virtual machine) by ending its wait state condition whenever a specified event occurs. WAKEUP events can be specified using optional WAKEUP parameters. WAKEUP events that may end a wait state include:

- The passing of a time of day (including a date or day of the week)
- The presence or arrival of reader files
- The arrival of a Virtual Machine Communication Facility message
- The arrival of a Special Message Facility message (from CP SMSG)

The WAKEUP Times File

Entries in a WAKEUP Times file are coded as follows:

Table 42. Format of Records in a WAKEUP Times File

Columns				
1–8	**10–17**	**19–26**	**28–255**	**Stacked**
ALL	HH:MM:SS	datestamp	user-text	once a day
MM/DD/YY	HH:MM:SS	datestamp	user-text	once
==/DD/YY	HH:MM:SS	datestamp	user-text	once a month
==/==/==	HH:MM:SS	datestamp	user-text	once a day
==/01/==	HH:MM:SS	datestamp	user-text	on the 1st
dayofweek	HH:MM:SS	datestamp	user-text	once a week
WEEKEND	HH:MM:SS	datestamp	user-text	on weekends
S-S	HH:MM:SS	datestamp	user-text	same as above
WEEKDAY	HH:MM:SS	datestamp	user-text	on weekdays
M-F	HH:MM:SS	datestamp	user-text	same as above
==/==/==	+05	timestamp	user-text	every 5 minutes
WEEKEND	+10:30	timestamp	user-text	every 10 minutes 30 seconds on weekends
WEEKDAY	+20	timestamp	user-text	every 20 minutes on weekdays
dayofweek	+5	timestamp	user-text	every 5 minutes on the specified day of the week
M-F	+02:30:0	timestamp	user-text	every 150 minutes on weekdays

WAKEUP Times File Format

The WAKEUP Times file format is:

```
date time stamp rest-of-record
```

The WAKEUP Times file only looks at the *date*, *time stamp* and *stamp fields*. These fields determine:

- If it should run the record today
- The time it should run the record
- When it last ran the record.

Note: If the date and time fields indicate a time before the current time, the virtual machine will wake up immediately.

The Date Field (Columns 1–8)

The date field is eight characters long and begins in column 1. It tells WAKEUP the date or day of the week when the record should be considered.

The date field format is:

mm/dd/yy
Specifies the exact date. You can also use an equal sign (=) for any of the numbers to specify general dates.

Example: If you Enter:

```
==/10/==
```

This tells the WAKEUP file to process something on the tenth of every month.

Example—If the date and time fields are exact: If you Enter:

```
03/15/87 03:45:30
```

The WAKEUP file changes the first slash to a period.

Example: If you Enter:

```
03.15/87
```

When the WAKEUP file processes the record, the first slash changes to a period. This makes it easy to write an EXEC to delete records that will never be ran again.

ALL
Specifies every day.

Example: If you Enter:

```
==/==/==
```

The WAKEUP file shows every day.

Day Name
Specifies a day of the week.

Example: If you Enter:

```
MONDAY
TUESDAY
WEDNESDA (You have to leave off the Y.)
THURSDAY
FRIDAY
SATURDAY

or

SUNDAY
```

Note: You can abbreviate any name to three letters.

WEEKEND or S-S
Specifies Saturday and Sunday.

WEEKDAY or M-F
Specifies every weekday.

MONTHLY
Specifies once a month.

Example: If your system is always up on the first of the month, you can Enter:

```
==/01/==
```

instead of Entering:

```
MONTHLY
```

MONTHLY is designed to ensure that the record will be processed once a month, even if your system happens to be down on the first of the month.

Example: You must Enter:

```
HH:MM:SS
```

in the time field with *MONTHLY*.

Note: You cannot use relative time intervals.

YEARLY
Specifies once a year.

Example: If your system is always up on New Year's Day you can, Enter:

```
01/01/==
```

instead of Entering:

```
YEARLY
```

Example: You must Enter:

```
HH:MM:SS
```

in the time field with *YEARLY*.

Note: You cannot use relative time intervals.

***** Specifies that this record is a comment. Comment records, those beginning with an asterisk can be anywhere in a WAKEUP Times file.

The Time Field (Columns 10–17)

The time field is also eight characters long and begins in column 10. It tells the WAKEUP file the time you want the record stacked.

The date field format is:

HH:MM:SS
Specifies the exact time.

+*MM*
Specifies every *MM* minutes.

For example, if you Enter:

```
+05
```

This tells the WAKEUP file to do this every 5 minutes.

+HH:MM:S
Specifies every *HH* hours, *MM* minutes, and *S0* seconds.

Note: The seconds can only be specified in multiples of 10.

Date/Time Stamp Field (Columns 19–26)

The date or time stamp fields are eight characters long and begin in column 19. The WAKEUP File records the last WAKEUP date or time here.

Example: If the time field contains an exact time:

```
23:55:00
```

The WAKEUP file records a *date stamp*.

Example: If the time field contains a relative time:

```
+15
```

The WAKEUP file records a *time stamp*

Note: This field should not be altered by the user to set the WAKEUP events. This should be done by coding the appropriate date field and time field entries.

The Rest of the Record (Columns 28–255)

The rest of the records field information or commands describing the event can start in column 28 and can extend to column 255.

Your application can put its own data here.

Note

The WAKEUP Times file must always have:

```
ALL 23:59:00 datestamp CP SLEEP 2 MIN
```

as its last entry; or some other event that will begin just prior to Midnight and will not end until after Midnight. Otherwise, WAKEUP will not run the WAKEUP Times file events on the next scheduled *WAKEUP* day.

Items Stacked by WAKEUP

The seven WAKEUP options that cause data to be stacked are: EXT, FILE, IO, IUCVMSG, SMSG, TIME, and VMCF. WAKEUP stacks the data in the following order regardless of the order the options are specified when you invoke WAKEUP:

1. Current date and time
2. Line from the WAKEUP Times file, or an asterisk (*) if no line is found
3. An IUCV, SMSG or VMCF message, or EXT or IO interrupt data.

Note: In general, the last line stacked by WAKEUP is the one you really want to use, not the first line.

For more information on the WAKEUP Times file, see *VM/ESA: CMS Utilities Feature*.

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes to the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300,
522 South Road
Poughkeepsie, NY 12601-5400
U.S.A.
Attention: Information Request

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 on non-IBM products should be addressed to the suppliers of those products.

All 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 the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information may contain sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to IBM's application programming interfaces. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Programming Interface Information

This guide is intended to help DirMaint administrators tailor DirMaint to meet the needs of their particular VM system. This book documents Diagnosis, Modification, or Tuning Information provided by DirMaint.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

- BookManager
- DFSMS/VM
- IBM
- IBMLink
- Library Reader
- RACF
- VM/XA
- VM/ESA
- z/VM

Other company, product, and service names may be trademarks or service marks of others.

Glossary

A list of VM terms and their definitions is available through the online HELP Facility. For example, to display the definition of “cms,” enter:

```
help glossary cms
```

You will enter the HELP Facility's online glossary file and the definition of “cms” will be displayed as the current line. When you are in the glossary file, you can also search for other terms.

If you are unfamiliar with the HELP Facility, you can enter:

```
help
```

to display the main HELP Menu, or enter:

```
help cms help
```

for information about the HELP command.

For more information about the HELP Facility, see the *z/VM: CMS User's Guide*. For more about the HELP command, see the *z/VM: CMS Command Reference*.

Bibliography

This bibliography lists the DirMaint books, and other books that you may find useful.

DirMaint Library

The following table lists the books in the DirMaint Release 5.0 library and their order numbers.

Title	Order Number
DirMaint: License Program Specifications	GC20-1837
DirMaint: General Information	GC20-1836
DirMaint: Tailoring and Administration	SC23-0533
DirMaint: Command Reference	SC20-1839
DirMaint: Messages	SC23-0437
DirMaint: Diagnosis Reference	SC24-5883

z/VM Version 3 Release 1.0 Library

The following table lists books in the z/VM Version 3 Release 1.0 library that may be helpful.

Title	Order Number
z/VM: VMSES/E Introduction and Reference	GC24-5947
z/VM: System Messages and Codes	GC24-5974
z/VM: Planning and Administration	SC24-5948
z/VM: CP Programming Services	SC24-5956
z/VM: REXX/VM Reference	SC24-5963
z/VM: CP Command and Utility Reference	SC24-5967
z/VM: CMS Command Reference	SC24-5969
z/VM: CMS Pipelines Reference	SC24-5971
z/VM: XEDIT Command and Macro Reference	SC24-5973

VM/ESA Version 2 Release 4.0 Library

The following table lists books in the VM/ESA Version 2 Release 4.0 library that may be helpful.

Title	Order Number
VM/ESA: VMSES/E Introduction and Reference	GC24-5837
VM/ESA: Planning and Administration	SC24-5750
VM/ESA: CP Command and Utility Reference	SC24-5773
VM/ESA: CMS Command Reference	SC24-5776
VM/ESA: REXX/VM Reference	SC24-5770
VM/ESA: CMS Pipeline Reference	SC24-5778
VM/ESA: XEDIT Command and Macro Reference	SC24-5780
VM/ESA: CP Programming Services	SC24-5760
VM/ESA: System Messages and Codes	GC24-5841

Other Related Books

The following table lists other books, outside the DirMaint and z/VM libraries, that may be helpful when using this book.

Title	Order Number
VM/ESA: CMS Utilities Reference	SC24-5535
RACF Security Administrator's Guide	SC28-1340
RACF Macros and Interfaces	SC28-1345
RACF Command Syntax Reference Booklet	SX22-0014
C2/B1 Security Feature User's Guide for VM/ESA with RACF	SC24-5564
C2/B1 Trusted Facility Manual for VM/ESA with RACF	SC24-5563
DFSMS/VM Function Level 221 Storage Administration Guide and Reference	SH35-0111
External Security Interface (RACROUTE) Macro Reference for MVS and VM	GC28-1366

CD-ROM

The following CD-ROM contains all the IBM libraries that are available in IBM BookManager format for current VM system products and current IBM licensed programs that run on VM. It also contains PDF versions of many z/VM publications and publications for some related IBM licensed programs.

- *Online Omnibus Edition: VM Collection*, SK2T-2067

Note: Only unlicensed publications are included.

Index

Special Characters

A

C

D

E

G

H

I

M

O

P

R

S

T

U

V

W

Readers' Comments

Directory Maintenance VM/ESA
Tailoring and Administration Guide
Release 5.0

Publication No. SC23-0533-05

You may use this form to report errors, to suggest improvements, or to express your opinion on the appearance, organization, or completeness of this book.

Date: _____

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note

Report system problems to your IBM representative or the IBM branch office serving you.
U.S. customers can order publications by calling the IBM Software Manufacturing Solutions at **1-800-879-2755**.

In addition to using this postage-paid form, you may send your comments by:

FAX	1-607-752-2327	Internet	pubrcf@vnet.ibm.com
IBMLink	GDLVME(PUBRCF)		

Would you like a reply? ___YES ___ NO If yes, please tell us the type of response you prefer.

___ Electronic address: _____

___ FAX number: _____

___ Mail: (Please fill in your name and address below.)

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments
SC23-0533-05

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department G60
International Business Machines Corporation
Information Development
1701 North Street
ENDICOTT NY 13760-5553

Fold and Tape **Please do not staple** Fold and Tape

SC23-0533-05

Cut or Fold
Along Line

File Number: S370/S390-34
Program Number: 5748-XE4

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC23-0533-05

Spine information:

IBM Directory Maintenance VM/ESA Tailoring and Administration Guide *Release 5.0*