z/VM

Directory Maintenance Facility Tailoring and Administration Guide

version 6 release 2

SC24-6190-02

z/VM

Directory Maintenance Facility Tailoring and Administration Guide

version 6 release 2

SC24-6190-02

Note:
Before using this information and the product it supports, read the information in “Notices” on page 275.

This edition applies to version 6, release 2, modification 0 of IBM z/VM (product number 5741-A07) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC24-6190-01.

Contents

Figures

Tables

About This Document

This is a reference document for the z/VM® Directory Maintenance Facility (DirMaint™) function level 620, for use on IBM® z/VM version 6.

DirMaint is a Conversational Monitor System (CMS) application that helps you manage your z/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 z/VM user directory. You can manage the directory with DirMaint through the use of commands. Thus, 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.

The DirMaint feature may be used with:

- RACF Security Server for z/VM
- 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.

Intended Audience

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

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 z/VM.

Conventions and Terminology

The terminology used in this document is as follows:

DirMaint An abbreviation for the program name.

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

DIRMaint The 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.

DIRM The minimum abbreviation of the DIRMaint command.

Where to Find More Information

For a list of the documents that can provide you with additional information on DirMaint and z/VM, see “Bibliography” on page 281.

Links to Other Online Documents

The online version of this document contains links to other online documents. These links are to editions that were current when this document was published. However, due to the nature of some links, if a new edition of a linked document has been published since the publication of this document, the linked document might not be the latest edition. Also, a link from this document to another document works only when both documents are in the same directory.

For further up-to-date information specific to DirMaint, see the official DirMaint website at IBM: Directory Maintenance (DirMaint)

How to Send Your Comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or give us any other feedback that you might have.

Use one of the following methods to send us your comments:

1. Send an email to mhvrcfs@us.ibm.com.
2. Go to IBM z/VM Reader's Comments (www.ibm.com/systems/z/os/zvm/zvmforms/webqs.html).
3. Mail the comments to the following address:
   IBM Corporation
   Attention: MHVRCFS Reader Comments
   Department H6MA, Building 707
   2455 South Road
   Poughkeepsie, NY 12601-5400
   U.S.A.
4. Fax the comments to us as follows:
   From the United States and Canada: 1+845+432-9405
   From all other countries: Your international access code +1+845+432-9405

Include the following information:

- Your name and address
- Your email address
- Your telephone or fax number
- The publication title and order number:
  z/VM V6R2 Directory Maintenance Facility Tailoring and Administration Guide
  SC24-6190-02
- The topic name or page number related to your comment
- The text of your comment

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

IBM or any other organizations will use the personal information that you supply only to contact you about the issues that you submit to IBM.

If You Have a Technical Problem

Do not use the feedback methods listed above. Instead, do one of the following:

- Contact your IBM service representative.
- Contact IBM technical support.
- See IBM: z/VM Service Resources (www.ibm.com/vm/service/).
- Go to IBM Support Portal (www.ibm.com/support/entry/portal/Overview/).

Summary of Changes

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the changes. Some program updates might be provided through z/VM service by program temporary fixes (PTFs) for authorized program analysis reports (APARs), which also might be available for some prior releases.

SC24-6190-02, z/VM Version 6 Release 2

This edition includes changes or additions to support the general availability of z/VM V6.2.

SC24-6190-01, z/VM Version 6 Release 1 (Updated Edition)

This edition includes changes to support product changes provided or announced after the general availability of z/VM V6.1.

SC24-6190-00, z/VM Version 6 Release 1

This edition includes changes or additions to support the general availability of z/VM V6.1.

Chapter 1. Introduction

z/VM Directory Maintenance Feature (DirMaint) is a CMS application that helps manage your z/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 with other products, each of which may have certain requirements that will impact how you configure and install DirMaint. Planning for those requirements from other products is essential to installing DirMaint.

DirMaint is installed using the Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E) component of z/VM. 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.

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 z/VM 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, z/VM 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 *z/VM: Directory Maintenance Facility Commands 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 *z/VM: Directory Maintenance Facility Commands 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 z/VM system.

DirMaint Install and Service User ID

The DirMaint install and service user ID, 6VMDIR20 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 z/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 6VMDIR20 29E disk, a DirMaint production disk. The general user also needs to access the DirMaint HELP files, usually residing on the MAINT 19D disk, which are copied from the 6VMDIR20 29D disk.

What is a Server?

A server in z/VM provides shared services to z/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 per CSE or SSI cluster. Only one DIRMAINT server can manipulate a single source directory at one time.
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 DATAMOVE servers being utilized by the one DIRMAINT server. In an SSI cluster, there must be one DATAMOVE server for each member of the SSI where available DASD to that member will be formatted, copied, or cleaned..
DIRMSAT	The DIRMSAT server is responsible for manipulating the object directory on systems other than the system the DIRMAINT server is on, or on the same system if maintaining duplicate copies of the object directory. There can be multiple DIRMSAT servers all being used by the one DIRMAINT server. If using DirMaint to synchronize

the object directory in a CSE or SSI cluster, there must be one DIRMSAT server running on each system that is *not* running the DIRMAINT 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 6VMDIR20 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.

   In an SSI cluster, it is recommended to have at least one DATAMOVE server per SSI member. Also, it is recommended that there be at least one DIRMSAT server on each SSI member that is *not* running the DIRMAINT server.

   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 61 and when you add or remove a DIRMSAT server, see Chapter 5, “Tailoring the DIRMSAT Service Machine,” on page 67.
4. The DIRMAINT, DATAMOVE and DIRMSAT service machines must have access to the DirMaint service machine code. They must access:
   - 6VMDIR20 491 disk, for production use
   - 6VMDIR20 492 disk, for testing 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 DIRECTXA command.

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 DATAMOVE 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.

DATAMOVE Service Machines

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

The DATAMOVE 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 extent 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 extent being deallocated again to prevent exposure of any residual data to the next user.

If the workload warrants it, or if DirMaint is running in a multiple system cluster, additional DATAMOVE server machines may be defined and the work divided among them.

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

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

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

- Assign each DATAMOVE server to handle a specific SSI member
- Allocate work to the various DATAMOVE servers.

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 CSE or SSI cluster, the satellite service machine 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.

In an SSI cluster, the satellite service machine will additionally route DIRMaint commands from users on its own system to the DIRMAINT machine and also route command output files from the DIRMAINT machine to command users on the satellite server's own system.

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 for performance, 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 *z/VM: Directory Maintenance Facility Commands Reference*.

Common PROFILE

A PROFILE statement defines the start of a profile entry in the source directory. 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 DIRECTXA command 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 9 (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 message is issued by CP.

3 Specifies the virtual machine type to be used.

`MACHINE ESA, MACHINE XA, or MACHINE XC`

are all acceptable for use by any of the DIRMAINT service machines: DIRMAINT, DATAMOVEs, and DIRMSATs.

The MACHINE statement is optional, if the virtual machine directory entries include a MACHINE statement, or if the system uses a GLOBALOPTS MACHINE directory statement to set the default for all virtual machines.

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. The first part contains all of the non-DASD directory statements. The second part contains all of the DASD directory statements.

IBM recommends using the default of DIRMAINT as the user ID of the directory management service machine.

Notes:

1. If you are defining DIRMAINT in an SSI cluster, DIRMAINT must be defined as a USER machine so that DIRMAINT can be logged on to only one member of the cluster at a time, and so that it may have access to the shared spool of

each DIRMSAT machine. This way, DIRMSAT can act as a spool file bridge between the DIRMAINT machine and users on the satellite server's own system.

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.
3. If you are defining DIRMAINT in a multiple-system CSE or SSI cluster, the use of shared spool files is recommended. Neither input nor output spool files of DIRMAINT should be placed on the CSE exclusion list. In an SSI cluster, the use of shared spool files is necessary so that the satellite server machine can act as a spool file bridge between users on the satellite server's system and the system on which DIRMAINT is running.
4. All systems in the cluster must be running the same level of DirMaint code.
5. Once you have a license for DirMaint, you must enable use of the DirMaint feature by following the instructions in the “Memo To Users for the z/VM Directory Maintenance Facility”. Note that if you are *not* using a single SYSTEM CONFIG file in an SSI cluster, then the SYSTEM CONFIG file for each system in the cluster must have DirMaint enabled.
6. You must add a 15D minidisk to the DIRMAINT machine.

DIRMAINT Non-DASD Directory Statements

```
USER DIRMAINT 1  AUTOONLY 2  128M 3  2047M 3  BDG 4
      5 INCLUDE COMMON
      6 ACCOUNT SYSTEM SYSPROG
      7 D8ONECMD FAIL LOCK
      8 IUCV ANY PRIORITY MSGLIMIT 100
      9 OPTION CONCEAL DIAG88 D84NOPAS IGNMAXU
```

Figure 2. DIRMAINT Non-DASD Directory Statements

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

1 Specifies the user ID of the DIRMAINT virtual machine.

2 Specifies the password for the DIRMAINT virtual machine.

3 Specifies the required virtual storage size. DIRMAINT will function on as little as 16MB of storage for small source directories. IBM recommends using the largest virtual machine size available for better performance. If you chose to use a default size less than 128MB, you should leave the maximum size at 2047MB.

4 Specifies the privilege classes based on z/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'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: 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.

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 DIAG88 option authorizes the DIRMAINT virtual machine to use DIAGNOSE X'88' for ESM password and password phrase authentication. This is needed by DirMaint when the ESM_PASSWORD_AUTHENTICATION_EXIT (DVHXPA) is configured, because DVHXPA uses the DMSPASS Callable Services Library routine (which in turn uses DIAGNOSE X'88') to authenticate ESM passwords and password phrases. If the ESM_PASSWORD_AUTHENTICATION_EXIT is not configured, this option is not needed.

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 if any directory entries contain 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

```
 1 LINK 6VMDIR20 491 591 MR  - Product code, primary.
 2 LINK 6VMDIR20 492 592 MR  - Product code, alternate.
 3 LINK 6VMDIR20 11F 11F MR  - Interface code, primary.
 4 LINK 6VMDIR20 41F 21F MR  - Interface code, alternate.
 5 LINK MAINT    123 123 MW - Object directory disk.
 6 LINK PMAINT   551 551 RR - SSI highest level part utilities
 7 MDISK 155 MR - Read/write scratch space, A-disk.
 8 MDISK 1FA MR - Spool file staging space, Z-disk.
 9 MDISK 1DF MR - Primary directory files.
10 MDISK 2DF MR - Optional secondary directory files.
11 MDISK 1AA MR - Primary transaction history files (optional).
12 MDISK 2AA MR - Optional secondary history files.
13 MDISK 1DB MR - Primary directory backup (optional).
14 MDISK 15D RR - Intersystem locking disk, for CSE cluster systems only.
15 MDISK 1DE MR - Delta object directory work edit disk.
```

Figure 3. 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 commands to all of the DIRMAINT servers for READ access (or READ and NEWREAD access if you chose to use FILECONTROL directories) to the directories that replace these 1DF and 2DF disks. You will need to update the DVHPROFA DIRMAINT and the DVHPROFA * files. For more information, see “DVHPROFA DIRMAINT” on page 28.

Ensure that access to minidisks is controlled by explicit link authorization. None of the DirMaint MDISKs should have any directory passwords (except for the 11F and 21F disks, 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 3 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” on page 11 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 3.

1 - **4** and **7** - **13**
If you chose to locate any of the MAINT or 6VMDIR20 “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. 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.

The default size of the object directory disk is 20 cylinders. For larger directories, the following formula may be used to determine the size needed based on the number of users in the directory:

$$\text{num\_cylinders} = \frac{\text{num\_users} * 4\ \text{(4K Blocks)}}{180\ \text{(4K blocks)}}$$

6 Specifies the 551 disk for the SSI highest level part utilities disk, which contains the DIRECTXA utility.

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

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

9 The 1DF **9** and 1AA **11** 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 **10** and 2AA **12** disks, if used, may reside on a single DASD volume, possibly the same volume as the 192, 21F, 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 **13** disk is on a third volume, control unit, and channel or adapter. The other disks may be on any volumes.

10 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 43.

Note: IBM recommends use of the 2DF disk.

11 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.

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 other types 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 updating the DVHPROFA file to use the 2AA disk. For more information, see "DVHPROFA DIRMAINT" on page 28.

13 Specifies the primary directory backup. This is optional.

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 and should not be present on a system maintaining only a single copy of the object directory.

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

- System maintaining duplicate copies of the object directory
- System in a multiple system CSE or SSI cluster.

15 The 1DE disk is required when the ONLINE_VIA_DELTA= ON option is used in the CONFIG DATADVH file (the default setting). In this case, the 1DE disk must be formatted and allocated in the following manner: cylinder 0 must be allocated as PERM space and cylinder 1 through END must be allocated as DRCT space for delta object directory processing. The ONLINE_VIA_DELTA value determines whether directory changes are applied by calling DIRECTXA, specifying the DELTA option and a mini-source-directory containing only the current directory changes. This creates a mini-object-directory on the 1DE disk, which is applied to the current online directory on the 123 disk.

In order to use the 1DE disk for performance enhancement, the 1DE disk must be associated with a file mode in the DVHPROFA DIRMAINT and DVHPROFA * files. The default file mode in the sample configuration file is X.

When ONLINE_VIA_DELTA processing is **not** used (ONLINE_VIA_DELTA= OFF), the 1DE disk is optional and directory changes are applied by running DIRECTXA against the full user directory file.

Directory Statements for the DATAMOVE Virtual Machine

The directory statement examples are divided into two parts. The first part contains all of the non-DASD directory statements. The second part contains all of the DASD directory statements.

Notes:

1. If you are defining DATAMOVE in a multiple-system SSI cluster, DATAMOVE must be defined as a USER machine so that it may be logged on to only one system in the cluster at a time, and so that it may have access to all publicly-defined minidisks within the SSI cluster. Also, each member of the SSI cluster should have a DATAMOVE machine defined for that member. IBM recommends naming the DATAMOVE machines in the SSI cluster as DATAMOVE for the DATAMOVE machine associated with slot 1 in the cluster, and DATAMOV2, DATAMOV3, and DATAMOV4 for the DATAMOVE machines associated with slots 2, 3, and 4, respectively.
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.
3. In a multiple-system CSE 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. However, IBM recommends that you avoid this and that you 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 log off. 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.

4. If you are defining DATAMOVE in a multiple-system CSE or SSI cluster, the use of shared spool files is recommended. Neither input nor output spool files of DATAMOVE should be placed on the CSE exclusion list.
5. All systems in the cluster must be running the same level of DirMaint code.
6. Once you have a license for DirMaint, you must enable use of the DirMaint feature by following the instructions in the "Memo To Users for the z/VM Directory Maintenance Facility". Note that if you are *not* using a single SYSTEM CONFIG file in an SSI cluster, then the SYSTEM CONFIG file for each system in the cluster must have DirMaint enabled.

DATAMOVE Non-DASD Directory Statements

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

```
USER DATAMOVE 1  AUTOONLY 2  32M 3  256M 3  BG 4
      5 INCLUDE COMMON
      6 ACCOUNT SYSTEM SYSPROG
      7 D8ONECMD FAIL LOCK
      8 IUCV ANY PRIORITY MSGLIMIT 100
      9 OPTION CONCEAL IGNMAXU LNKEXCLV LNKSTABL
```

Figure 4. DATAMOVE Non-DASD Directory Statements

1 Specifies the user ID of the DATAMOVE virtual machine.

2 Specifies the password for the DATAMOVE virtual machine.

3 Specifies the required virtual storage size. DATAMOVE will function on as little as 32MB of storage for small source directories. IBM recommends using the largest virtual machine size available for better performance. If you chose to use a default size less than 32MB, you should leave the maximum size at 256MB.

4 Specifies the privilege classes based on z/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.

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

DATAMOVE DASD Directory Statements

```
1 LINK 6VMDIR20 491 591 RR - Product code, primary.
2 LINK 6VMDIR20 492 592 RR - Product code, alternate.
3 LINK 6VMDIR20 11F 11F RR - Interface code, primary.
4 LINK 6VMDIR20 41F 21F RR - Interface code, alternate.
5 LINK PMAINT   551 551 RR - SSI highest level part utilities
6 MDISK 155 MR - Read/write scratch space, A-disk.
7 MDISK 1FA MR - Spool file staging space, Z-disk.
8 MDISK 1AA MR - Primary transaction history files (optional).
9 MDISK 2AA MR - Optional secondary history files.
```

Figure 5. DATAMOVE DASD Directory Statements

The DirMaint product code and data files should be on conventional minidisk space rather than SFS space. However, any of these DIRMAINT MDISKS, 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 commands to all of the DATAMOVE servers for READ access (or READ and NEWREAD access if you chose to use FILECONTROL directories) to the directories that replace these 1DF and 2DF disks. You will need to update the DVHPROFM DATADVH file for each of the DATAMOVE machines accordingly. For more information, see “DVHPROFA DIRMAINT” on page 28

Ensure that access to minidisks is controlled by explicit link authorization, as determined by the minidisk owner. None of the DirMaint MDISKs should have any directory passwords (except for the 11F and 21F disks, 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 5.

1 - 9

If you have chosen to locate any of the MAINT or 6VMDIR20 “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 551 disk for the SSI highest level part utilities disk, which contains the CPFMTXA utility.

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

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

8 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.

9 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 **8** and 2AA **9** 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.

Directory Statements for the DIRMSAT Virtual Machine

The DIRMSAT directory example is divided into two parts. The first part contains all of the non-DASD directory statements. The second part contains all of the DASD directory statements.

Notes:

1. If you are defining DIRMSAT in an SSI cluster, the satellite server machine must be defined as a USER machine so that it may be logged on to only one system in the cluster at a time, and so that it may have access to the shared spool of the DIRMAINT virtual machine. This way, it can act as a spool file bridge between the DIRMAINT machine and users on the satellite server's own system. Also, each member of the SSI cluster should have a DIRMSAT machine

defined for that member. IBM recommends naming the satellite service machines in the SSI cluster as DIRMSAT for the satellite service machine associated with slot 1 in the cluster, and DIRMSAT2, DIRMSAT3, and DIRMSAT4 for the satellite service machines associated with slots 2, 3, and 4, respectively.

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.
3. In a multiple-system CSE cluster, you will probably have multiple virtual machines active and performing the DIRMSAT 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. However, IBM recommends that you avoid this and that you 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 log off. 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.
4. If you are defining DIRMSAT in a multiple-system CSE or SSI cluster, the use of shared spool files is recommended. Neither input nor output spool files of DIRMSAT should be placed on the CSE exclusion list. In an SSI cluster, the use of shared spool files is necessary so that the satellite server machine can act as a spool file bridge between users on the satellite server's system and the system DIRMAINT is running on.
5. All systems in the cluster must be running the same level of DirMaint code.
6. Once you have a license for DirMaint, you must enable use of the DirMaint feature by following the instructions in the “Memo To Users for the z/VM Directory Maintenance Facility”. Note that if you are *not* using a single SYSTEM CONFIG file in an SSI cluster, then the SYSTEM CONFIG file for each system in the cluster must have DirMaint enabled.

DIRMSAT Non-DASD Directory Statements

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

```
USER DIRMSAT 1  AUTOONLY 2  128M 3  256M 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 6. DIRMSAT Non-DASD Directory Statements

1 Specifies the user ID of the DIRMSAT virtual machine.

2 Specifies the password for the DIRMSAT virtual machine.

3 Specifies the required virtual storage size. DIRMSAT will function on as little as 16MB of storage for small source directories. IBM recommends using

the largest virtual machine size available for better performance. If you chose to use a default size less than 128MB, you should leave the maximum size at 256MB.

4 Specifies the privilege classes based on z/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.

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

```
 1 LINK 6VMDIR20 491 591 RR - Product code, primary.
 2 LINK 6VMDIR20 492 592 RR - Product code, alternate.
 3 LINK 6VMDIR20 11F 11F RR - Interface code, primary.
 4 LINK 6VMDIR20 41F 21F RR - Interface code, alternate.
 5 LINK MAINT    123 123 MW - Object directory disk.
 6 LINK PMAINT   551 551 RR - SSI highest level part utilities
 7 LINK DIRMAINT 1DF 1DF RR - Primary directory files.
 8 LINK DIRMAINT 2DF 2DF RR - Optional secondary directory files.
 9 LINK DIRMAINT 15D 15D RR - Intersystem locking disk.
10 MDISK 155 MR - Read/write scratch space, A-disk.
11 MDISK 1FA MR - Spool file staging space, Z-disk.
12 MDISK 1AA MR - Primary transaction history files (optional).
13 MDISK 2AA MR - Optional secondary history files.
14 MDISK 1DE MR - Delta object directory work edit disk.
```

Figure 7. DIRMSAT DASD Directory Statements

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 commands to all of the DIRMSATs for READ access (or READ and NEWREAD access if you chose to use FILECONTROL directories) to the directories that replace these disks. You will need to update the DVHPROFA * files for each of the DIRMSAT machines accordingly. For more information, see “DVHPROFA DIRMAINT” on page 28.

Ensure that access to minidisks is controlled by explicit link authorization, as determined by the minidisk owner. None of the DirMaint MDISKs should have any directory passwords (except for the 11F and 21F disks, 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 7.

1 - 8 and 10 - 13.
If you have chosen to locate any of the MAINT, 6VMDIR20 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. In an SSI cluster, it must match the address used on the DIRECTORY statement in the USER INPUT file. In a CSE cluster, 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 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.

The default size of the object directory disk is 20 cylinders. For larger directories, the following formula may be used to determine the size needed based on the number of users in the directory:

$$\text{num\_cylinders} = \frac{\text{num\_users} * 4\ (\text{4K Blocks})}{180\ (\text{4K blocks})}$$

6 Specifies the 551 disk for the SSI highest level part utilities disk, which contains the DIRECTXA utility.

7 Specifies the primary directory files.

8 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.

9 Specifies the intersystem locking disk.

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

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

12 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.

13 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 **12** and 2AA **13** 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.

14 The 1DE disk is required when the ONLINE_VIA_DELTA= ON option is used in the CONFIG DATADVH file (the default setting). In this case, the 1DE disk must be formatted and allocated in the following manner: cylinder 0 must be allocated as PERM space and cylinder 1 through END must be allocated as DRCT space for delta object directory processing. The ONLINE_VIA_DELTA value determines whether directory changes are applied by calling DIRECTXA, specifying the DELTA option and a mini-user-directory containing only the current directory changes. This creates a mini-object directory on the 1DE disk, which is applied to the current online directory on the 123 disk.

In order to use the 1DE disk for performance enhancement, the 1DE disk must be associated with a file mode in the DVHPROFA * files for each of the DIRMSAT machines accordingly. The default file mode in the sample configuration file is X.

When ONLINE_VIA_DELTA processing is **not** used (ONLINE_VIA_DELTA= OFF), the 1DE disk is optional and directory changes are applied by running DIRECTXA against the full user directory file.

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 *z/VM: RSCS Networking Planning and Configuration*.

Chapter 3. Tailoring the DIRMAINT Service Machine

This chapter provides guidance for tailoring the various data files used by the DIRMAINT service machine. You must tailor several files so they can be used. After you make these changes, however, the files are updated by the DirMaint commands. For more information on these commands, see the *z/VM: Directory Maintenance Facility Commands 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 DIR2PROD SAMP has placed these tailorable files on the 6VMDIR20 492 minidisk:
  - DVHNAMES DATADVH
  - DIRMAINT DATADVH
  - DIRMSAT DATADVH
  - DATAMOVE DATADVH
  - DVHPROFM DATADVH
  - DVHPROFA DIRMAINT
  - DVHPROFA DIRMSAT
  - PROFILE EXEC
- The DIR2PROD SAMP has placed these tailorable files on the 6VMDIR20 41F minidisk:
  - CONFIG DATADVH
  - DIRMMAIL DATADVH
  - 140CMDS DATADVH
  - 150CMDS DATADVH

  Note: These files should only be modified on the 6VMDIR20 492/41F test minidisks. Once DIR2PROD PROD has been run to place the DirMaint code into production on the 6VMDIR20 491/11F production minidisks, change these tailorable files as follows:

  1. Request the DIRMAINT server detach the 6VMDIR20 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*.

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. New Files for DirMaint

File Name	Description	Page
PROFILE XEDIT	This file determines the characteristics of your editing sessions. This file should be tailored first.	27
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.	28
CONFIG DATADVH	This file contains most of the tailorable parameters used throughout DirMaint.	28
DIRMAINT DATADVH	This file identifies the schedule of events to the DirMaint service machine that happen at specific set dates, times, or intervals.	43
DVHNAMES DATADVH	This file identifies the user ID's to be notified of any significant events that happen in the various DirMaint service machines.	46
DIRMMAIL SAMPDVH	The file identifies a sample for the DIRMMAIL NEWFILE file.	48

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 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.	48
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.	49
RPWLIST DATA	This file contains a list of prohibited passwords.	50

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

File Name	Description	Page
EXTENT CONTROL	This file provides information needed for DirMaint's DASD Management functions. For more information on the EXTENT CONTROL file, see “The Extent Control File” on page 74.	73
AUTHDASD CONTROL	This determines who can allocate space in what DASD groups, regions, or volumes.	73

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 Before Bringing Up DirMaint

File Name	Description	Page
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.	52
USER INPUT	This file is your existing source directory file.	55

Note: 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*.

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
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 z/VM user ID, enter:

Step 1. SENDFILE PROFILE XEDIT * 6VMDIR20

Step 2. Enter, a RECEIVE command from the 6VMDIR20 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 DIR2PROD SAMP. For information, see the *DirMaint Program Directory*.

If you have installed DirMaint 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.

Apart from renaming this file as DVHPROFA *userid*, customization also needs to be done in DVHXLVL EXEC with dirmid = '*userid*' and in ACCESS DATADVH with USE= *userid*.

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).

Multiple CONFIG* DATADVH files are allowed and recommended. There are two types of entries in these files: using single occurrence 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 Single Occurrence 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 single occurence 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 108.

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    ONLINE_VIA_DELTA= ON | OFF
6    WRK_UNIT_ONLINE= NO | YES
7  // DIRECTXA_OPTIONS= MIXED | MIXED NOMIXMSG
8    DEFAULT_DIRECT_ACTION= UNCONDITIONAL | CONDITIONAL
9    SORT_DIRECTORY= NO | YES
10   SORT_BY_DEVICE_ADDRESS= NO | YES
11   BACKUP REBUILD= CLUSTER DVHLINK <VCONTROL> | NONE
12   CLASS LIMIT ON USER STATEMENT=      8 | 0 ... 32 | 0 ... 8
13   CLASS STATEMENT IN PROFILE CHECK = NO | YES
14   WRK_UNIT_CLEANUP= ERASE | RENAME
15   LINK_MAX_INDIRECT=
```

Figure 8. 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 DIRECTXA command. 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 4/26/02 to ==/==/== or another date value suggested for the DIRECT entry in the DIRMAINT DATADVH file. For more information, see the “DIRMAINT DATADVH” on page 43.

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 ONLINE_VIA_DELTA entry determines whether directory changes are applied by calling DIRECTXA, specifying the DELTA option and a source directory subset containing only the current directory changes. This creates an object directory subset which is applied to the current online directory. ONLINE_VIA_DELTA options are ON (apply directory changes using a delta directory) or OFF (apply directory changes by running DIRECTXA against the full user directory file). The default is ON.

When ONLINE_VIA_DELTA is ON and UPDATE_IN_PLACE is YES, directory changes that can be applied using Diagnose X'84' will still be placed online using Diagnose X'84' and other directory changes will be placed online using a delta directory created by DIRECTXA with the DELTA option.

6 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 placing 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`.

7 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 Commands and Utilities 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.

8 The DEFAULT_DIRECT_ACTION value specifies the default for the optional UNCONDITIONAL or CONDITIONAL parameters on the DIRMAINT DIRECT command. If specified as CONDITIONAL, the DIRECTXA command will not be issued unless there are pending changes to be processed. If specified as UNCONDITIONAL, the DIRECTXA command will be issued, regardless of any pending changes to be processed. (If omitted, the default is UNCONDITIONAL.) For more information, see the DIRMAINT DIRECT command in *z/VM: CP Commands and Utilities Reference*.

9 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.

10 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.

11 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.
NONE	Specifies the keyword value of NONE may be used.

12 The CLASS_LIMIT_ON_USER_STATEMENT= specifies how many CP privilege classes may be included on the USER statement. The valid range is 0 to 32. The default 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 *

13 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. 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.

14 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.

15 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 limit of 50.

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
4 1SAPI_REQUESTS_BEHAVIOR=    2 25   /* DAYS and PERCENTAGE */
```

Figure 9. 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 performed 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 disable directory updates and log themselves off.

4 The 1SAPI_REQUESTS_BEHAVIOR statement determines how the 1SAPI REQUESTS file (which contains a log of the synchronous application programming interface requests and their return codes) is processed. Large amounts of SAPI requests can cause the DIRMAINT 155 disk to become full. In order to prevent disk full errors (which can cause the DIRMAINT server to shutdown), the DIRMAINT server will automatically prune the 1SAPI REQUESTS file when the remaining space on the DIRMAINT 155 disk is less than or equal to the amount of disk space needed to prune the file plus 5 percent of the disk.

The 1SAPI_REQUESTS_BEHAVIOR statement can be used to configure the number of previous days for which to keep request information and the percentage of the 1SAPI REQUESTS file to prune when it is growing too large. The options on this statement are as follows:

PREVIOUS_DAYS
This is an integer between 1 and 9 that represents the number of previous days for which to keep request information. The default value is 2 days.

PRUNE_PERCENTAGE
This is an integer between 5 and 90 that represents the percentage of total requests which should be pruned from the 1SAPI REQUESTS file if the space on the DIRMAINT 155 disk becomes smaller than the space needed to prune the file plus 5 percent of the disk. The default value is 25 percent.

Refer to the *DirMaint Program Directory* for information on how to determine the size of the DIRMAINT 155 disk when DirMaint is used with the z/VM Systems Management APIs.

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

Figure 10. Selecting Security and Auditing Characteristics

1 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 statement name. The IBM-supplied routine (DVHXPA EXEC) is ready for use with RACF or other ESMs issuing a call to the DMSPASS CSL routine for the password verification interface.

2 If your system is running with Mandatory Accessed Control (MAC), and you followed the directions in Appendix A, “External Security Manager Considerations,” on page 191, 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.

3 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. 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 73.

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.

4 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*.
- 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.

5 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.

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.

6 It is possible that an error condition may arise in the message handling routines. 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 you wish to DirMaint 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.

7 The POSIX_UID_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.

8 and 9

- `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.

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.

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

10 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.

11 The ALLOW_ASUSER_NOPASS_FROM= *serverid* *|*servernode* entry identifies the *userid* and *nodeid* of trusted service machines who can make requests including the ASUSER prefix keyword (which generally forces authentication) without supplying a password and thus without authentication. For example, they are allowed to issue:

```
EXEC DIRMAINT AS DIRMAINT FOR anyid
```

and make any and all changes to the directory that could be made from the DIRMAINT server's console. This capability *must* only be given to virtual machine that are well trusted not to misuse this capability. A *nodeid* of * may be used to represent any system within the cluster where the DIRMAINT server is running.

Note: This statement is checked by the DIRM command user's virtual machine and therefore must be located on the DIRMAINT 11F disk.

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  PW_MONITOR=userid
10 PW_REUSE_HASHING_EXIT=
11 PW_REUSE_INTERVAL=
```

Figure 11. 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 149.

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 49.

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.

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 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.

10 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.

11 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.

Note: If the IBM supplied default of 365 DAYS is changed, you need to enable a PASSWORD CHANGE NOTIFICATION EXIT = DVHXPN EXEC statement in the CONFIG* DATADVH file.

Step 5. Select RACF-Specific Characteristics

If your system specifically has RACF installed as the ESM, the following entries set defaults for exit calls, RACF commands, and other characteristics for RACF functions, including user creation and deletion, password management, POSIX

segment management, ACI group management, permission requirement for facilities that require additional coordinated CP and RACF privileges, and discrete resource profile creation and deletion.

Note that if RACF administration is decentralized, then DirMaint should have the group-SPECIAL attribute. This attribute makes DirMaint an administrator at a group level, thereby enabling it to control access to its group and to issue RACF commands.

Verifying that RACF Administration is Decentralized

To verify this attribute, enter:

```
RAC LU DIRMAINT
```

On the console, look to see if it says:

```
CONNECT ATTRIBUTES=SPECIAL
```

If you do *not* see this attribute, enter:

```
RAC CONNECT DIRMAINT GROUP(grpname) SPECIAL
```

If RACF administration is centralized, then DirMaint should have the SPECIAL attribute, which makes it an administrator and enables it to issue RACF commands and to control access to all users.

Verifying that RACF Administration is Centralized

To verify this attribute, enter:

```
RAC LU DIRMAINT
```

On the console, look to see if it says:

```
ATTRIBUTES=SPECIAL
```

If you do *not* see this attribute, enter:

```
RAC ALTUSER DIRMAINT SPECIAL
```

A sample file, CONFIGRC SAMPDVH, is supplied with the product code. If no RACF communication is desired, no action is required. If RACF communication is desired, this file should be renamed to CONFIGRC DATADVH and used as an override file for RACF-specific configuration entries. The sample override file contains a USE_RACF= YES ALL configuration statement to configure the DirMaint server to use all default IBM-supplied RACF connector support. The sample file should be reviewed and changed to meet the needs of the installation, if required.

```
1  USE_RACF= YES|NO ALL|exit_name
2  PASSWORD_CHANGE_NOTIFICATION_EXIT= DVHXPN EXEC
3  POSIX_CHANGE_NOTIFICATION_EXIT= DVHXPESM EXEC
4  LOGONBY_CHANGE_NOTIFICATION_EXIT= DVHXLB EXEC
5  USER_CHANGE_NOTIFICATION_EXIT= DVHXUN EXEC
6  DASD_OWNERSHIP_NOTIFICATION_EXIT= DVHXDN EXEC
7  RACF_ADDUSER_DEFAULTS= UACC(NONE)
8  RACF_RDEFINE_VMMDISK_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
9  RACF_DISK_OWNER_ACCESS= ACC(ALTER)
10 RACF_RDEFINE_VMPOSIX_POSIXOPT.QUERYDB= UACC(READ)
11 RACF_RDEFINE_VMPOSIX_POSIXOPT.SETIDS= UACC(NONE)
12 RACF_RDEFINE_SURROGAT_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
13 RACF_RDEFINE_VMBATCH_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
14 RACF_RDEFINE_VMRDR_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
15 RACF_VMBATCH_DEFAULT_MACHINES= BATCH1 BATCH2
16 TREAT_RAC_RC.4= 0 | 4 | 30
17 ESM_PASSWORD_AUTHENTICATION_EXIT= DVHXPA EXEC
```

Figure 12. Selecting RACF-Specific Characteristics

1 The USE_RACF= entry controls the use of the DoRACF global variable within the DIRMAINT service machine. This variable controls the execution of the IBM-supplied RACF connector function. Whenever a configured exit is called by DIRMAINT, the DoRACF global variable is set based on all configured USE_RACF statements. DoRACF is set to true when the exit is configured to be enabled for automatic RACF communication; otherwise DoRACF is set to false.

USE_RACF= YES ALL indicates that all DirMaint user exits are enabled for automatic RACF communication (except for those configured on a USE_RACF= NO statement). With USE_RACF= YES ALL, all exits (except for those overridden using a USE_RACF= NO statement) will be called with a DoRACF value of true. Also, all of the following IBM-supplied user exits containing automatic RACF communication are defined with their default values in the CONFIGRC sample configuration file:

- ESM_PASSWORD_AUTHENTICATION_EXIT= DVHXPA
- DASD_OWNERSHIP_NOTIFICATION_EXIT= DVHXDN
- LOGONBY_CHANGE_NOTIFICATION_EXIT= DVHXLB
- PASSWORD_CHANGE_NOTIFICATION_EXIT= DVHXPN
- POSIX_CHANGE_NOTIFICATION_EXIT= DVHXPESM
- USER_CHANGE_NOTIFICATION_EXIT= DVHXUN

Note: If any of these exits are defined explicitly in a DirMaint override configuration file, then the name specified on the definition statement will be used.

USE_RACF= NO ALL indicates that all DirMaint user exits are disabled for automatic RACF communication (except for exits overridden using USE_RACF= YES statements). This is the default if no USE_RACF= YES ALL statement is configured. With USE_RACF= NO ALL, all exits will be called with a DoRACF value of false (except for those configured on a USE_RACF= YES statement). When USE_RACF= NO ALL is used, all USE_RACF= YES ALL statements will be ignored. If there is not a USE_RACF= YES ALL statement configured, USE_RACF= NO ALL will be in effect.

USE_RACF= YES|NO *exit_name* indicates that the specified exit will be enabled (YES) or disabled (NO) for automatic RACF communication.

Multiple USE_RACF statements may be used to enable or disable multiple exits. Whenever the *exit_name* specified on a USE_RACF= YES or USE_RACF= NO statement is called by DirMaint, the DoRACF global variable will be set to true or false, respectively. The *exit_name* specified on the USE_RACF statement is the file name and file type of the exit. For example, to enable the automatic RACF communication in the default USER_CHANGE_NOTIFICATION_EXIT, use the following statement in a DirMaint configuration override file:

```
USE_RACF= YES DVHXUN EXEC
```

Note: When using the *exit_name* option, the exit must also be defined using the exit's definition statement.

2 The PASSWORD_CHANGE_NOTIFICATION_EXIT= entry identifies the exit to be called to issue the necessary RACF commands for DIRMaint PW and SETPW command processing.

3 The POSIX_CHANGE_NOTIFICATION_EXIT= entry identifies the exit to be called to issue the necessary RACF commands for DIRMAINT POSIXGLIST, POSIXGROUP, POSIXINFO, POSIXFSROOT, POSIXIUPGM, POSIXIWDIR, POSIXUID, and POSIXOPT command processing.

4 The LOGONBY_CHANGE_NOTIFICATION_EXIT= entry identifies the exit to be called to issue the necessary RACF commands for DIRMAINT LOGONBY command processing.

5 The USER_CHANGE_NOTIFICATION_EXIT= entry identifies the exit to be called to issue the necessary RACF commands for processing the user profile-related DirMaint commands (such as DIRMAINT ADD, PURGE, etc.).

6 The DASD_OWNERSHIP_NOTIFICATION_EXIT= entry identifies the exit to be called to issue the necessary RACF commands for processing the DASD-related DirMaint commands (such as DIRMAINT AMDISK, DMDISK, etc.).

7 The RACF_ADDUSER_DEFAULTS= entry specifies the defaults that will be used by DVHXUN when it issues a RACF ADDUSER command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE).

8 The RACF_RDEFINE_VMMDISK_DEFAULTS= entry specifies the defaults that will be used by DVHXDN when it issues a RACF RDEFINE VMMDISK command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied defaults are UACC(NONE) AUDIT(FAILURES(READ)).

9 The RACF_DISK_OWNER_ACCESS= entry specifies the access authority that will be used by DVHXDN when it issues a RACF PERMIT command for the owner of the disk. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default in the CONFIGRC sample configuration file is RACF_DISK_OWNER_ACCESS= ACC(ALTER). To use the default access configured in RACF, use a RACF_DISK_OWNER_ACCESS= statement without an access authority specified (i.e, a blank RACF_DISK_OWNER_ACCESS= statement).

10 The RACF_RDEFINE_VMPOSIX_POSIXOPT.QUERYDB= entry specifies the defaults that will be used by DVHXUN or DVHXPESM when it issues a RACF RDEFINE VMPOSIX POSIXOPT.QUERYDB command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(READ).

11 The RACF_RDEFINE_VMPOSIX_POSIXOPT.SETIDS= entry specifies the defaults that will be used by DVHXUN or DVHXPESM when it issues a RACF RDEFINE VMPOSIX POSIXOPT.SETIDS command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE).

12 The RACF_RDEFINE_SURROGAT_DEFAULTS= entry specifies the defaults that will be used by DVHXUN or DVHXLB when it issues a RACF RDEFINE SURROGAT command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE) AUDIT(FAILURES(READ)).

13 The RACF_RDEFINE_VMBATCH_DEFAULTS= entry specifies the defaults that will be used by DVHXUN when it issues a RACF RDEFINE VMBATCH command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE) AUDIT(FAILURES(READ)).

14 The RACF_RDEFINE_VMRDR_DEFAULTS= entry specifies the defaults that will be used by DVHXUN when it issues a RACF RDEFINE VMRDR command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE) AUDIT(FAILURES(READ)).

15 The RACF_VMBATCH_DEFAULT_MACHINES= entry identifies the batch machines available on the system.

16 The TREAT_RAC_RC.4= entry identifies how DVHXUN, DVHXDN, DVHXPESM, and DVHXLB will interpret the RACF return code 4 (authorization decision deferred by RACF to z/VM) from the RACF commands – as if the return code was 0 (successful) or 30 (RACF not installed). The default is 4, which denotes no change in interpretation.

17 The ESM_PASSWORD_AUTHENTICATION_EXIT= entry identifies the exit to be called to issue the necessary commands to authenticate a user using a CP logon password or External Security Manager password phrase.

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 DATADVH File Example

```
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 DATADVH 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 delete 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 4/26/02 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 4/26/02, 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 4/26/02 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 4/26/02 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 236.

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 4/26/02 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 173.

The IBM-supplied file uses a date of 4/26/02 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 236.

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 235.

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 these nicknames:

Table 7. 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.

Table 7. DVHNAMES DATADVH Nickname Entries (continued)

Nickname	Function
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.
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 DIRMMAIL SAMPDVH file is a sample for a DIRMAINT NEWMAIL file. The IBM supplied sample provides a description of DirMaint's key features 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.

The LINKS EXCLUDE file has this 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 155.

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.

The PWMON CONTROL file has this 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 z/VM 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	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 DIRECTXA MODULE. For more information on the DIRECTXA MODULE, see the *z/VM: CP Commands and Utilities Reference*. The file is not altered by DirMaint in any way, and you must maintain it in exactly the format documented in the z/VM publications.

Note: If you rename or erase the RPWLIST DATA file, a warning message will be issued by the 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.

SUBSCRIB DATADVH

The SUBSCRIB DATADVH file contains a list of subscribers to be notified when any userid, or particular userids, are changed. This file is built and maintained by the SUBSCRIBE command.

The SUBSCRIB DATADVH file has this format:

Columns 1-7	The ENTRYTYPE field – INCLUDE or EXCLUDE.
Column 8	Blank.
Columns 9-15	The TARGETID field – the userid subscribed to, or ALL.
Column 16	Blank.
Columns 17-22	The ENCODING field – ASCII (default for TCP or UDP) or EBCDIC (default for RDR or SMSG).
Column 23	Blank.
Columns 24-27	The PROTOCOL field – RDR, SMSG, TCP, or UDP.
Column 28	Blank.
Columns 29-44	The DESTPARM1 field – the z/VM userid for RDR or SMSG, or the IP address for TCP or UDP.
Column 45	Blank.
Columns 46-53	The DESTPARM2 field – the node ID for RDR or SMSG, or the port number for TCP or UDP. An asterisk is allowed and will be treated as the local node ID, but such a subscription cannot be separately deleted or queried.
Column 54	Blank.
Columns 55-119	The SUBSCRIBER_DATA_STRING field – an optional character or hexadecimal string supplied by the subscriber, with no embedded spaces. An asterisk is allowed, but such a subscription cannot be separately deleted or queried.

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.

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.

This file must be in this 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. Specify an asterisk (*) for requests originating on the local system (the same system where the DIRMAINT server is running) or other systems within the same local CSE system cluster. For requests originating beyond the local system or system cluster, specify the actual system node name.

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.

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 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 E, by default). The first time you type **DVHBEGIN** when there is no USER DIRECT file, 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 name given to a PROFILE, USER, IDENTITY or SUBCONFIG in the directory must be unique. You may not have two profiles with the same name, two virtual machines with the same user ID, a profile and a virtual machine with the same name/user ID, and so on.
- 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'

  + = X'4E'

  – = X'60'

  : = X'7A'

  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'
+ = X'4E'
– = X'60'
: = X'7A'

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, USER, IDENTITY or SUBCONFIG statement to which they apply. DirMaint considers a directory entry to begin with the PROFILE, USER, IDENTITY, or SUBCONFIG statement (or an external format of the SYSAFFIN statement), and includes all records up to the next PROFILE, USER, IDENTITY or SUBCONFIG 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 .....
```

- 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

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, and so forth 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 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.

Message Destination

The first parameter on a call to the DVHMSG EXEC when issuing a message is the message destination. The message will be sent to one of these destination:

: Specifies self, as An XEDIT message for use in the menu processor. This parameter should only be used for messages issued within the command originators (users) virtual machine

***** Specifies self, in the service machines the message may be logged. This parameter can be used for messages issued by a DirMaint Server or from within the command originators virtual machine.

? Originator. The message may be logged.

- Originator. The message will NOT be logged.

+ Self. The message may be logged, AS IF sent to originator.

Note: The ?, -, and the + parameters along with all nickname codes should only be used for messages issued by a DirMaint Server.

nick Nickname for one of DirMaint's special distribution lists:

Nick	Distribution List Name	Who's in the List
A	DVHALL	All of the groups that follow in this table.
C	DVHCERT	Computer Emergency Response Team, also known as the Security Team.
M	DVHPWMON	The password Monitoring team.
O	DVHOPER	The system Operator.
S	DVHSUPT	The Support programmer (usually MAINT) for command from self, but the same as ? for a user entered command.
0	DVHDRCT	DVHDRCT progress notices.
1	DVHOURLY	DVHOURLY notices.
2	DVHDAILY	DVHDAILY notices.

Other single characters are DVHSUPT notices. The message may be logged.

Restart or Shutdown Processing After Encountering an Error

This DVHSHUT EXEC checks to see if a DVHSHUTX CONTROL file exists. This file contains two counters and an action indicator. The LOGOFF counter determines when DVHSHUT issues a CP LOGOFF command, and the RESET counter determines when DVHWAIT erases the DVHSHUTX CONTROL file. The action indicator determines whether DVHSHUT will PURGE the command retry information before doing a re-IPL.

1. If the DVHSHUTX CONTROL file did not exist previously, the file is created with both counters initialized to the threshhold values. Both counters are countdown counters.
2. Whether or not the file previously existed, the LOGOFF counter is decremented by one, the RESET counter is set to the threshhold value, the action indicator is toggled, and the DVHSHUTX CONTROL file is re-written.
3. The command retry information is purged, if the action indicator is set accordingly.
4. If the LOGOFF counter is not yet zero, a restart re-IPL is done.
5. If the LOGOFF counter is less than or equal to zero, and the service machine is running disconnected, a CP LOGOFF is done. If the LOGOFF counter is less than or equal to zero, and the service machine is running connected, another re-IPL is done.

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 DirMaint 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 191.

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 28. The format is:

```
DATAMOVE_MACHINE= userid nodeid sysaffin
```

Where:

userid
: Identifies the DATAMOVE service machine you are logging on.

nodeid
: Identifies the system on which the DATAMOVE machine is running.

sysaffin
: Identifies the system affinity code that can be processed. This is a required parameter. A single DATAMOVE machine can either process requests for a single system affinity or for all system affinities. Specify an asterisk (*) if the DATAMOVE machine will process all system affinities. At least one entry with a system affinity of asterisk (*) is required.

  In an SSI cluster, this field is not used and should be specified as an asterisk (*). The DATAMOVE machine is instead selected through the use of the ATnode prefix to the DIRMaint command, by the member system on which a subconfig is defined, or by the next INACTIVE DATAMOVE machine. See the *z/VM:*

Directory Maintenance Facility Commands Reference for more information pertaining to the DIRMaint command ATnode prefix.

Example—DATAMOVE Server on a Stand-Alone System:

```
DATAMOVE_MACHINE= DATAMOVE * *
```

Example—DirMaint SSI Cluster:

```
DATAMOVE_MACHINE= DATAMOVE DVHTEST1 *
DATAMOVE_MACHINE= DATAMOV2 DVHTEST2 *
DATAMOVE_MACHINE= DATAMOV3 DVHTEST3 *
DATAMOVE_MACHINE= DATAMOV4 DVHTEST4 *
```

Example—DirMaint CSE Cluster:

```
DATAMOVE_MACHINE= DATAMOVE 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
```

Step 2. Identify the Communication Path

In some configurations, it may be necessary to identify the communications path between DIRMAINT and each of the DATAMOVE machines, as well as the return path between the DATAMOVE machines and DIRMAINT.

If RSCS is in use, the default communications path should work for DATAMOVE machines within the same multiple-system CSE or SSI cluster. Likewise, if RSCS is not in use and shared spool files are in use, the default communications path should work for DATAMOVE machines within the same multiple-system CSE or SSI cluster. To run with the default communications path, do not configure any routing statements.

If the default communications path is not sufficient to route commands between the DATAMOVE machines and the DIRMAINT machine, you can identify the appropriate communications path with the following configuration statements in a DirMaint override configuration file:

Note: This is not required for a DATAMOVE machine running on the same *nodeid* as the DIRMAINT machine.

```
FROM= fromspec DEST= destspec S= spoolid T= tagspec1 U= tagspec2
```

Where:

fromspec
: Identifies the network *nodeid* or service machine *userid* where the transaction originates.

destspec
: Identifies the network *nodeid* or service machine *userid* 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 or spool file bridge.

tagspec1
Identifies the network *nodeid* or service machine *userid* of the spool file tag.

tagspec2
Identifies the *userid* of the spool file tag.

Notes:

1. Without CSE shared spooling, the routing definitions for the DIRMSAT machines should be the same as those for the DATAMOVE machines.
2. For making changes to a DirMaint override configuration file, refer to "CONFIG DATADVH" on page 28.

Example—DirMaint SSI or CSE Cluster With RSCS:

```
FROM= DVHTEST1 DEST= DVHTEST2 S= DIRMNET1 T= DVHTEST2
FROM= DVHTEST2 DEST= DVHTEST1 S= DIRMNET2 T= DVHTEST1
FROM= DVHTEST1 DEST= DVHTEST3 S= DIRMNET1 T= DVHTEST3
FROM= DVHTEST3 DEST= DVHTEST1 S= DIRMNET3 T= DVHTEST1
FROM= DVHTEST2 DEST= DVHTEST3 S= DIRMNET2 T= DVHTEST3
FROM= DVHTEST3 DEST= DVHTEST2 S= DIRMNET3 T= DVHTEST2
```

Where:

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

Example—DirMaint SSI or CSE Cluster With CSE Shared Spooling:

```
* Routing records for DIRMAINT to DATAMOVE
FROM= DVHTEST1 DEST= DATAMOV2 S= DATAMOV2 T= DATAMOV2
FROM= DVHTEST1 DEST= DATAMOV3 S= DATAMOV3 T= DATAMOV3
FROM= DVHTEST1 DEST= DATAMOV4 S= DATAMOV4 T= DATAMOV4
FROM= DVHTEST2 DEST= DATAMOVE S= DATAMOVE T= DATAMOVE
FROM= DVHTEST2 DEST= DATAMOV3 S= DATAMOV3 T= DATAMOV3
FROM= DVHTEST2 DEST= DATAMOV4 S= DATAMOV4 T= DATAMOV4
FROM= DVHTEST3 DEST= DATAMOVE S= DATAMOVE T= DATAMOVE
FROM= DVHTEST3 DEST= DATAMOV2 S= DATAMOV2 T= DATAMOV2
FROM= DVHTEST3 DEST= DATAMOV4 S= DATAMOV4 T= DATAMOV4
FROM= DVHTEST4 DEST= DATAMOVE S= DATAMOVE T= DATAMOVE
FROM= DVHTEST4 DEST= DATAMOV2 S= DATAMOV2 T= DATAMOV2
FROM= DVHTEST4 DEST= DATAMOV3 S= DATAMOV3 T= DATAMOV3
**
* Routing records for DATAMOVE to DIRMAINT
FROM= DATAMOVE DEST= DIRMAINT S= DIRMAINT T= DVHTEST1 U= DATAMOVE
FROM= DATAMOV2 DEST= DIRMAINT S= DIRMAINT T= DVHTEST2 U= DATAMOV2
FROM= DATAMOV3 DEST= DIRMAINT S= DIRMAINT T= DVHTEST3 U= DATAMOV3
FROM= DATAMOV4 DEST= DIRMAINT S= DIRMAINT T= DVHTEST4 U= DATAMOV4
```

Notes:

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

Step 3. Define the DATAMOVE Retry and Autolog Limits

If a DATAMOVE machine is unable to link to a minidisk because a user is linked to a disk for which a CMDISK command has been issued, or because the directory change to transfer the minidisk to DATAMOVE has not been placed online, then 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. After DirMaint has been notified that this limit has been reached, DirMaint will not assign any more work to that particular DATAMOVE machine. A value in the range of 0 through 9999 may be specified. The default is 10. If a value outside the valid range is specified, then the value will be set to 1.

The MAXIMUM_WORKUNIT_RETRIES value determines the number of times DirMaint will retry a specific request after the first attempt to process request. Once the number of retries are attempted without success, the request will be cancelled and rolled back. A value in the range of 0 through 999 may be specified. The default is 10. If a value outside of the valid range is specified, then the value will be set to the default 10.

The format for these entries is:

```
DM_MAXIMUM_RETRIES= integer
MAXIMUM_WORKUNIT_RETRIES= integer
```

Before assigning a workunit to a DATAMOVE machine, DirMaint will determine if the DATAMOVE machine is logged on. If the DATAMOVE machine is *not* logged on, DirMaint will attempt to autolog it. The MAXIMUM_DATAMOVE_AUTOLOGS value specifies the number of times DirMaint will attempt to autolog a DATAMOVE machine which is not logged on to the system before quiescing the machine for manual intervention. A value in the range of 0 through 99 may be specified. The default is 10. If a value outside of the valid range is specified, then the value will be set to the default 10.

Note that the autolog count is reset to zero during DAILY processing so that if an autolog is successful and no further failures occur before DAILY processing, then MAXIMUM_DATAMOVE_AUTOLOGS will be attempted again. The format of the entry is:

```
MAXIMUM_DATAMOVE_AUTOLOGS= integer
```

Notes:

1. DATAMOVE machines defined on remote nodes, with respect to the DIRMAINT server, will *not* be autologged in a CSE cluster.
2. DATAMOVE machines defined on remote nodes, with respect to the DIRMAINT server, *will* be autologged in an SSI cluster.
3. If DirMaint is unable to determine if the DATAMOVE machine is logged on, the workunit will be assigned to the DATAMOVE server for processing. If the DATAMOVE machine is not logged on in this case, the workunit will not be processed until the system administrator autologs the associated DATAMOVE machine.

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 61.

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 sparse 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 DATADVH File Example

```
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 DATADVH 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 235.

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 SSI or 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 can be used to synchronize the object directory of a node in the cluster with the object directory of the node running DIRMAINT. A satellite service machine can also 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. Due to the use of the satellite servers as a spool file bridge for DIRM commands in an SSI cluster, IBM requires the satellite service machines to be defined as single-configuration (USER) virtual machines and *not* multi-configuration (IDENTITY) virtual machines.

Except for the name of the virtual machine, each one should be defined to DirMaint as described in "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 in Appendix A, "External Security Manager Considerations," on page 191.

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 28. The format is:

```
SATELLITE_SERVER= userid nodeid
```

Where:

userid
: Identifies the satellite service machine you are logging on.

nodeid
: Identifies the system on which the satellite machine is running.

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

```
SATELLITE_SERVER= DIRMSAT  DIRMNODE
```

Example—DirMaint SSI Cluster:

```
SATELLITE_SERVER= DIRMSAT DVHTEST1
SATELLITE_SERVER= DIRMSAT2 DVHTEST2
SATELLITE_SERVER= DIRMSAT3 DVHTEST3
SATELLITE_SERVER= DIRMSAT4 DVHTEST4
```

Example—DirMaint CSE Cluster:

```
SATELLITE_SERVER= DIRMSAT DVHTEST1
SATELLITE_SERVER= DIRMSAT2 DVHTEST2
SATELLITE_SERVER= DIRMSAT3 DVHTEST3
SATELLITE_SERVER= DIRMSAT4 DVHTEST1
SATELLITE_SERVER= DIRMSAT5 DVHTEST2
SATELLITE_SERVER= DIRMSAT6 DVHTEST3
```

Step 2. Identify the Communication Path

In some configurations, it may be necessary to identify the communications path between DIRMAINT and each of the DIRMSAT machines, as well as the return path between the DIRMSAT machines and DIRMAINT.

If RSCS is in use, the default communications path should work for DIRMSAT machines within the same multiple-system CSE or SSI cluster. Likewise, if RSCS is not in use and shared spool files are in use, the default communications path should work for DIRMSAT machines within the same multiple-system CSE or SSI cluster. To run with the default communications path, do not configure any routing statements.

If the default communications path is not sufficient to route commands between the DIRMSAT machines and the DIRMAINT machine, you can identify the appropriate communications path with the following configuration statements in a DirMaint override configuration file:

Note: This is not required for a DIRMSAT machine running on the same *nodeid* as the DIRMAINT machine.

```
FROM= fromspec DEST= destspec S= spoolid T= tagspec1 U= tagspec2
```

Where:

fromspec
: Identifies the network *nodeid* or service machine *userid* where the transaction originates.

destspec
: Identifies the network *nodeid* or service machine *userid* 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 DIRMSAT, at that node. Otherwise, it is the *userid* of an RSCS network machine or spool file bridge.

tagspec1
: Identifies the network *nodeid* or service machine *userid* of the spool file tag.

tagspec2
: Identifies the *userid* of the spool file tag.

Notes:

1. Without CSE shared spooling, the routing definitions for the DIRMSAT machines should be the same as those for the DATAMOVE machines.
2. For making changes to a DirMaint override configuration file, refer to "CONFIG DATADVH" on page 28.

Example—DirMaint SSI or CSE Cluster With RSCS:

```
FROM= DVHTEST1 DEST= DVHTEST2 S= DIRMNET1 T= DVHTEST2
FROM= DVHTEST2 DEST= DVHTEST1 S= DIRMNET2 T= DVHTEST1
FROM= DVHTEST1 DEST= DVHTEST3 S= DIRMNET1 T= DVHTEST3
FROM= DVHTEST3 DEST= DVHTEST1 S= DIRMNET3 T= DVHTEST1
FROM= DVHTEST2 DEST= DVHTEST3 S= DIRMNET2 T= DVHTEST3
FROM= DVHTEST3 DEST= DVHTEST2 S= DIRMNET3 T= DVHTEST2
```

Where:

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

Example—DirMaint SSI Cluster With CSE Shared Spooling:

```
* Routing records for DIRMAINT to DIRMSAT
FROM= DVHTEST1 DEST= DIRMSAT2 S= DIRMSAT2 T= DIRMSAT2
FROM= DVHTEST1 DEST= DIRMSAT3 S= DIRMSAT3 T= DIRMSAT3
FROM= DVHTEST1 DEST= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST2 DEST= DIRMSAT S= DIRMSAT T= DIRMSAT
FROM= DVHTEST2 DEST= DIRMSAT3 S= DIRMSAT3 T= DIRMSAT3
FROM= DVHTEST2 DEST= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST3 DEST= DIRMSAT S= DIRMSAT T= DIRMSAT
FROM= DVHTEST3 DEST= DIRMSAT2 S= DIRMSAT2 T= DIRMSAT2
FROM= DVHTEST3 DEST= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST4 DEST= DIRMSAT S= DIRMSAT T= DIRMSAT
FROM= DVHTEST4 DEST= DIRMSAT2 S= DIRMSAT2 T= DIRMSAT2
FROM= DVHTEST4 DEST= DIRMSAT3 S= DIRMSAT3 T= DIRMSAT3
**
* Routing records for DIRMSAT to DIRMAINT
FROM= DIRMSAT DEST= DIRMAINT S= DIRMAINT T= DVHTEST1 U= DIRMSAT
FROM= DIRMSAT2 DEST= DIRMAINT S= DIRMAINT T= DVHTEST2 U= DIRMSAT2
FROM= DIRMSAT3 DEST= DIRMAINT S= DIRMAINT T= DVHTEST3 U= DIRMSAT3
FROM= DIRMSAT4 DEST= DIRMAINT S= DIRMAINT T= DVHTEST4 U= DIRMSAT4
**
* Route DIRM users on DIRMAINT machine node through DIRMAINT machine.
* In this example, DIRMAINT is running on DVHTEST1.
FROM= DVHTEST1 DEST= DVHTEST2 S= DIRMAINT T= *
FROM= DVHTEST1 DEST= DVHTEST3 S= DIRMAINT T= *
FROM= DVHTEST1 DEST= DVHTEST4 S= DIRMAINT T= *
**
* Route DIRM users on satellite node through DIRMSAT machine.
FROM= DVHTEST2 DEST= DVHTEST1 S= DIRMSAT2 T= *
FROM= DVHTEST2 DEST= DVHTEST3 S= DIRMSAT2 T= *
FROM= DVHTEST2 DEST= DVHTEST4 S= DIRMSAT2 T= *
FROM= DVHTEST3 DEST= DVHTEST1 S= DIRMSAT3 T= *
FROM= DVHTEST3 DEST= DVHTEST2 S= DIRMSAT3 T= *
FROM= DVHTEST3 DEST= DVHTEST4 S= DIRMSAT3 T= *
FROM= DVHTEST4 DEST= DVHTEST1 S= DIRMSAT4 T= *
FROM= DVHTEST4 DEST= DVHTEST2 S= DIRMSAT4 T= *
FROM= DVHTEST4 DEST= DVHTEST3 S= DIRMSAT4 T= *
```

Notes:

1. Routes for DIRM users must be redefined if the DIRMAINT machine is brought up on a different member system. The network node associated with the

DIRMAINT machine would need to change, and the DIRMSAT machine associated with the network node where DIRMAINT formerly ran would need to act as a spool file bridge for users on that node.

2. Routes to service machine user IDs must come before routes to network node IDs. In other words, more specific routes must be configured in front of more general routes.

Example—DirMaint CSE Cluster With CSE Shared Spooling:

```
* ROUTING RECORDS FOR DIRMAINT TO DIRMSAT
FROM= DVHTEST1 DEST= DIRMSAT2 S= DIRMSAT2 T= DIRMSAT2
FROM= DVHTEST1 DEST= DIRMSAT3 S= DIRMSAT3 T= DIRMSAT3
FROM= DVHTEST1 DEST= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST1 DEST= DIRMSAT5 S= DIRMSAT5 T= DIRMSAT5
FROM= DVHTEST1 DEST= DIRMSAT6 S= DIRMSAT6 T= DIRMSAT6
FROM= DVHTEST2 DEST= DIRMSAT S= DIRMSAT T= DIRMSAT
FROM= DVHTEST2 DEST= DIRMSAT3 S= DIRMSAT3 T= DIRMSAT3
FROM= DVHTEST2 DEST= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST2 DEST= DIRMSAT5 S= DIRMSAT5 T= DIRMSAT5
FROM= DVHTEST2 DEST= DIRMSAT6 S= DIRMSAT6 T= DIRMSAT6
FROM= DVHTEST3 DEST= DIRMSAT S= DIRMSAT T= DIRMSAT
FROM= DVHTEST3 DEST= DIRMSAT2 S= DIRMSAT2 T= DIRMSAT2
FROM= DVHTEST3 DEST= DIRMSAT4 S= DIRMSAT4 T= DIRMSAT4
FROM= DVHTEST3 DEST= DIRMSAT5 S= DIRMSAT5 T= DIRMSAT5
FROM= DVHTEST3 DEST= DIRMSAT6 S= DIRMSAT6 T= DIRMSAT6
**
* ROUTING RECORDS FOR DIRMSAT TO DIRMAINT
FROM= DVHTEST1 DEST= DVHTEST2 S= DIRMAINT T= DVHTEST2
FROM= DVHTEST1 DEST= DVHTEST3 S= DIRMAINT T= DVHTEST3
FROM= DVHTEST2 DEST= DVHTEST1 S= DIRMAINT T= DVHTEST1
FROM= DVHTEST2 DEST= DVHTEST3 S= DIRMAINT T= DVHTEST3
FROM= DVHTEST3 DEST= DVHTEST1 S= DIRMAINT T= DVHTEST1
FROM= DVHTEST3 DEST= DVHTEST2 S= DIRMAINT T= DVHTEST2
```

Notes:

1. Routes to service machine user IDs must come before routes to network node IDs. In other words, more specific routes must be configured in front of more general routes.
2. For a full 16-system CSE 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. Note, however, that the DIRMAINT machine may only run on one system in the SSI or CSE cluster at a time.

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

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 DATADVH File Example

```
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 DATADVH 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 235.

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 5.

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 28.

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 13 on page 75. 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 DIR2PROD 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  :SSI_VOLUMES.
   * ------------------------------------------------------ *
   * SSI volumes used for subconfig clone are mapped here.  *
   * VolumeFamily  Member  Volser                           *
   * ------------------------------------------------------ *
   RESIDENCE DVHTEST1 RES1
   RESIDENCE DVHTEST2 RES2
   RESIDENCE DVHTEST3 RES3
   RESIDENCE DVHTEST4 RES4
   :END.
4  :EXCLUDE.
   * ------------------------------------------------------ *
   * All excluded users and user devices are                *
   * placed into this section                               *
   * ------------------------------------------------------ *
   MAINT 0190
   MAINT 02*
   :END.
5  :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.
6  :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

.
.
.
7  :END.
```

Figure 13. 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 :SSI_VOLUMES.	Defines which DASD volumes to use for new minidisk allocation when cloning a new SUBCONFIG entry from an existing SUBCONFIG entry.
4 :EXCLUDE.	Defines directory entry or entry/device combinations that should be considered as excluded by the DirMaint DASD subsystem.
5 :AUTOBLOCK.	Defines blocking factors and device architectures for various device types. These supplement or override the IBM supplied definitions in the AUTOBLK DATADVH file
6 :DEFAULTS.	Defines the default maximum size for various DASD devices. These supplement or override the IBM supplied definitions in the DEFAULT DATADVH file
7 :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 regstart regend ttttmmmm
```

You can also use a region entry to define a full-volume minidisk. In this case, the *volser* value is defined twice, and the format is:

```
volser volser volstart volend ttttmmmm
```

where *regend* and *volend* are calculated using the *size* and *start* values when the EXTENT CONTROL file is updated by the DASD command. The values are calculated as follows:

```
regend = (start + size) - 1
volend = size - 1
```

The parameters are:

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 $.

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

regend/volend
Specifies the ending block or cylinder (inclusive) of the region/volume. A keyword of END can be used to define the end of a region/volume. 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 as 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.

:SSI_VOLUMES. Section

DirMaint provides the capability to clone a new SUBCONFIG entry using an existing SUBCONFIG entry via the LIKE option on the DIRM ADD command. The :SSI_VOLUMES. (**3**) section allows you to define which DASD volumes DirMaint will use when allocating new minidisks associated with the new cloned SUBCONFIG entry. Entries within the :SSI_VOLUMES. section define which DASD volume corresponds to which user-defined set of volumes across each member of the SSI cluster. When cloning an existing minidisk, DirMaint will determine to which set of volumes the existing minidisk belong, and then determine which volume to use for the same set on the new member of the SSI cluster. When allocating DASD space using an SSI_VOLUME. section entry, DirMaint will use the same cylinder offsets on the new member's per-system volume as are used on the existing member's per-system volume. Similarly, the device type of the old member disk will be used as the device type for the new member disk. Note that DirMaint does not

copy data from the existing subconfig disk to the new subconfig disk. The data copy must be done using DDR, as specified in the use case scenario documented in *z/VM: CP Planning and Administration*.

The format for an :SSI_VOLUMES. section entry is:

```
volume_set_name ssi_member volser
```

Where:

volume_set_name
: Specifies the name of the set of volumes DirMaint will use when allocating new minidisks on a per-system basis – for example, SYSRES for the set of system-residence volumes across the SSI cluster. Volume set names supported by DirMaint must be less than or equal to 16 characters in length and cannot contain any spaces.

ssi_member
: Specifies the system name of the SSI cluster member for which new minidisks will be allocated.

volser
: Specifies the volume ID of the DASD to be used when allocating new minidisks on the specified *ssi_member* system.

Notes:

1. This value represents the value placed into the *volser* field on any minidisks generated from this entry.
2. Volume IDs supported by DirMaint can contain only the characters A-Z, 0-9, #, @ and $.
3. Each entry in the :SSI_VOLUMES. section must have a unique volume ID. The same volume ID cannot be used for multiple nodes or multiple volume sets.

Usage Notes

1. :SSI_VOLUMES. section entries can be specified in mixed case, but note that 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 :SSI_VOLUMES. entry is ignored.
3. DirMaint does not impose a limit on the number of :SSI_VOLUMES. entries.

Example—Defining a Volume Set/SSI Cluster Member Relationship:

The following entries define *volser*'s RES1, RES2, RES3, and RES4 associated with a volume set called RESIDENCE for SSI members DVHTEST1, DVHTEST2, DVHTEST3 and DVHTEST4, respectively.

```
:SSI_VOLUMES.
RESIDENCE DVHTEST1 RES1
RESIDENCE DVHTEST2 RES2
RESIDENCE DVHTEST3 RES3
RESIDENCE DVHTEST4 RES4
:END.
```

: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:

entry_name address

Where:

entry_name
Identifies the directory entry name in which the minidisk with the extent or extents to be excluded is defined. This is the USER, IDENTITY or SUBCONFIG name associated with the entry. An entry name can be followed by an asterisk (*) 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 within the directory entry 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 in any directory entry starting with HOWLAND, is shown as:

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

If you specify an * without an *entry_name*, 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.

:AUTOBLOCK. Section

AutoBlock (**5**) 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 empty. Modifications should only be made to this section if your installation is using a device type not defined in the AUTOBLK DATADVH file.
2. When initializing the volume control files, the device architecture is taken from the AUTOBLOCK section, or from the AUTOBLK DATADVH file.
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. 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.

:DEFAULTS. Section

As discussed in ":REGIONS. Section" on page 76, 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. (**6**) 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:

ttttmmmm 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 as 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.

cyl|blk
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., :SSI_VOLUMES., :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 150. 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 121.
- 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 89.

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 *z/VM: Directory Maintenance Facility Commands 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 Commands and Utilities 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 87. 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 14. 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 76. 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 *z/VM: Directory Maintenance Facility Commands 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 *z/VM: Directory Maintenance Facility Commands 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 WURETRIES: &RETRY
16 SSINODE: *
17 BEGINCMDS:
18 NTRIED  19 WORKUNIT  20 ENABLE
18 NTRIED  19 AMDISK  20 FOR DSSERV 0306 33        90 AUTOG 500 GDLVM7 MR
18 NTRIED  19 UNLOCK  20 0306 DSSERV NOMSG
18 NTRIED  19 WORKUNIT  20 RESET
18 NTRIED  19 DIRECT
```

Figure 15. 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 the number of times the work unit has been retried or &RETRY in an unassigned work unit.

16 Is the SSI member name associated with the command invocation. An '*' indicates that the disk being processed may be handled by a DATAMOVE server on any member of the SSI. Any other value indicates the member on which the associated DATAMOVE machine should be running. This is the value of the ATnode prefix in the associated DIRM command. If the ATnode prefix was '*' and the FORuser is a SUBCONFIG entry, then this is the SSI member associated with the SUBCONFIG on the BUILD statement in the associated IDENTITY entry.

17 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:

18 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 *z/VM: Directory Maintenance Facility 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.

19 Part 2

Command Name
Denotes the subcommand name.

20 Part 3

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

Subsystem Control

The DASD subsystem control structure consist of these files that exist on the DirMaint servers 1DF disk:

DATAMOVE Control File

The DATAMOVE control file is the primary control structure for managing interaction with the DATAMOVE machine. There is only a single occurrence of this file on the 1DF disk, its name is DATAMOVE CONTROL. A single entry for each defined DATAMOVE machine is contained within. This file provides the serialization required when dealing with DATAMOVE machines. Its format is as follows:

```
userid nodeid sysaffin activity fdevtab pending autologs unitid
```

- `Userid` is the userid of the DATAMOVE machine
- `Nodeid` is the node ID of the DATAMOVE machine
- `Sysaffin` is the system affinity associated with this DM
- `Activity` is the current activity of this DATAMOVE machine
  - ACTIVE
    - Indicates that a transaction was sent to the DATAMOVE machine and no response has yet been received.
  - INACTIVE
    - Indicates that the DATAMOVE machine is available for additional work.
  - QUIESCE
    - Indicates that the DATAMOVE machine is not available for additional work and should not be dispatched.
- `Fdevtab` is the filename of the free device table
- `Pending` is the number of pending requests

  - Due to DATAMOVE retry situations
    - Copy delayed while waiting for a link to device to drop
    - Copy delayed while waiting for directory to be placed online.
  - It is also used to load level the assignment of work
- `Autologs` is the number of times the DATAMOVE machine has been autologged. DirMaint will autolog a DATAMOVE machine if the machine is logged off when DirMaint needs to assign a work unit to it.
- `Unitid` is the current Work Unit Control File (WUCF)
  - Only valid if ACTIVE

This file is only rebuilt during initialization if no active Work Unit Control Files exist. As this file contains status information regarding the activity of work units, it's information must not be lost by rebuilding this file. If no active work units exist the file is rebuilt by DVHINITI during system initialization.

xxxxFDEV DVHTABLE File

Essentially is a bit map describing what devices are currently in use in the associated DATAMOVE machine. The 'xxxx' is an ordinal between 0000 and 9999 that is assigned to a specific DATAMOVE machine. One xxxxFDEV DVHTABLE exists for each DATAMOVE machine. The DATAMOVE CONTROL file associates a specific DATAMOVE machine with this file. The table is referenced during a device assignment being made to a DATAMOVE machine. The mapped range is '0100'x to '05FF'x on the DATAMOVE virtual machine.

Unassigned Queue

The unassigned queue is used as a repository for unassigned WUCFs. If a DATAMOVE machine can not be assigned once the work unit is created it is placed into this file. The maximum size of this file is governed by the MAXIMUM_UNASSIGNED_WORKUNITS= configuration file entry. The date and time the WUCF was created is also placed into this file. This file only exists if WUCFs are unassigned. It exists on the DirMaint servers 1DF disk and is named

- UNASSIGN DVHQUEUE

Processing Retry or Stalled Work Units

Work units processed by the DATAMOVE machine may sometimes not be able to complete. There are a number of possible reasons for this, but in most cases, a disk which needs to be formatted or copied to is linked by another user. The DirMaint server will attempt to retry work units based on the MAXIMUM_WORKUNIT_RETRIES configuration statement value. The default is to retry the work unit indefinitely. Eventually, the disk may become detached and the DATAMOVE machine will be able to successfully complete the work unit.

If the DirMaint server is waiting indefinitely for the disk to be detached and you decide that the work unit processing is no longer desired, you may choose to use manual intervention. The disk may have been detached and you may want to retry the work unit immediately, or you may wish to cancel the work unit and roll back any processing which has already occurred.

In other cases, because the DirMaint DASD management subsystem is so complex, it's possible that DirMaint will sometimes lose track of everything it's supposed to be tracking. If a work unit is stalled or if the DATAMOVE machine is hung, manual

intervention may be necessary to get the DATAMOVE machine working again. The commands in the following two sections can be used to diagnose and process retry or stalled work units.

See *z/VM: Directory Maintenance Facility Commands Reference* for more information on any of these commands.

Commands For Diagnosing Work Units

Use the following command to diagnose work units:

- **DIRMaint Query**

  The DIRMaint Query command may be used to request current system information from the DIRMAINT server. The current status of DATAMOVE machines may be requested, as well as the current number of pending work elements for a DATAMOVE machine, or the status of the current unassigned work unit queue. Detailed information for specific work units may also be queried.

- **DIRMaint STATus**

  The DIRMaint STATus command may be used to request the current status of DATAMOVE virtual machines, as well as list all work units being processed by DirMaint. Detailed information on specific work units may also be requested.

All of this information may be used to determine if a work unit is in a retry or stalled state, or if a DATAMOVE machine has become quiesced or hung. If the status of the DATAMOVE is QUIESCED, the DATAMOVE machine is disabled and will not process any more work units without manual intervention. If the number of pending work units for a DATAMOVE machine or the number of work units on the unassigned queue is growing and the DATAMOVE machine status is ACTIVE or INACTIVE, then the DATAMOVE machine may be in a hung condition. Individual work units may be queried to determine what step or process is stalled for the work unit. Once it is determined that a work unit or the entire DATAMOVE machine is stalled, the commands in the following section may be helpful in addressing your specific situation.

Commands For Processing Work Units

Use the following command to process work units:

- **DIRMaint WORKUNIT**

  The DIRMaint WORKUNIT command may be used to cancel or retry specific work units. If the DATAMOVE machine is not quiesced or hung, this command may be used to initiate retry or cancel operations.

- **DIRMaint CLEANUP**

  The DIRMaint CLEANUP command may be used to unhang a DATAMOVE machine by cleaning up internal files and retrying or canceling all work units associated with that DATAMOVE machine. If the DATAMOVE machine is in a QUIESCED state or in a hung condition, this command may be used to perform an automatic cleanup.

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
- Failure of old minidisk CLEANUP. The WUCF created for an AMDISK, CMDISK, or DMDISK that requires clean-up of an old minidisk being released. If the clean-up should fail, DIRMAINT is able to:
  - Transfer a new minidisk to user
  - Release the old 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.

Important

Note: The 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 89, 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.”

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 *z/VM: Directory Maintenance Facility 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
WURETRIES: &RETRY
SSINODE: *
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 16. AMDISK With No DATAMOVE Interaction

The following steps are provided to help you with your “AMDISK With No DATAMOVE Interaction” on page 96 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: DATAMOVE GDLVMK1
       DEV.ONE: 100
       DEV.TWO:
       ORIGNODE: GDLVMK1
       ORIGUSER: DIRMAINT
       ORIGSEQ#: 92
       ORIGCMD: AMDISK 6543 3380 2001 1 K1CP04 LABEL MJH191
       SYSAFFIN: *
       TARGETID: HOWLAND9
       LANG: AMENG
       CMDLEVEL: 150A
       ASUSER: DIRMAINT
       REQUEST: 92
       ORIGEXTENT: N/A
       WURETRIES: &RETRY
|      SSINODE: *
       BEGINCMDS:
     1 NTRIED WORKUNIT ENABLE
|    2 NTRIED AMDISK FOR DATAMOVE 100 3380 2002 1 K1CP04
     3 NTRIED WORKUNIT RESET
     4 NTRIED DIRECT
|    5 NTRIED DMVCTL DATAMOVE GDLVM7 DMVCTL FORMAT DIRMAINT GDLVMK1 32027026 1 HOWLAND9 6543 * 100 = MJH191
     6 NTRIED WORKUNIT ENABLE
|    7 NTRIED TMDISK FOR DATAMOVE 100 HOWLAND9 6543
     8 NTRIED WORKUNIT RESET
     9 NTRIED DIRECT
    10 NTRIED UNLOCK 6543 HOWLAND9 NOMSG
```

Figure 17. 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: DATAMOVE 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
       WURETRIES: &RETRY
|      SSINODE: *
       BEGINCMDS:
     1 NTRIED WORKUNIT ENABLE
|    2 NTRIED AMDISK FOR DATAMOVE 100 XXXX AUTOR 5 MIKEPLAY MR SAM DOC HARRY
|    3 NTRIED TMDISK FOR HOWLAND9 0191 DATAMOVE 101
     4 NTRIED WORKUNIT RESET
     5 NTRIED DIRECT
|    6 NTRIED DMVCTL DATAMOVE GDLVMK1 DMVCTL COPY DIRMAINT GDLVMK1 04003811 1 HOWLAND9 0191 * 101 100
     7 NTRIED WORKUNIT ENABLE
|    8 NTRIED TMDISK FOR DATAMOVE 100  HOWLAND9 0191
     9 NTRIED UNLOCK 0191 HOWLAND9 NOMSG
|   10 NTRIED DMDISK FOR DATAMOVE 101 NOCLEAN KEEPLINKS
    11 NTRIED WORKUNIT RESET
    12 NTRIED DIRECT
```

Figure 18. 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
WURETRIES: &RETRY
SSINODE: *
BEGINCMDS:
1 NTRIED DMDISK FOR HOWLAND9 0194 NOCLEAN
2 NTRIED UNLOCK 0194 HOWLAND9 NOMSG
3 NTRIED DIRECT
```

Figure 19. 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
WURETRIES: &RETRY
SSINODE: *
BEGINCMDS:
1 NTRIED DMDISK FOR HOWLAND9 0194 CLEAN
```

Figure 20. 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
WURETRIES: &RETRY
SSINODE: *
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 21. 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
WURETRIES: &RETRY
SSINODE: *
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 22. TMDISK

The following steps are provided help you with your “ZAPMDISK (Auxiliary DMDISK)” on page 104 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 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 6VMDIR20 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 *z/VM: Directory Maintenance Facility Commands 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= 6VMDIR20 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 *z/VM: RACF Security Server Macros and Interfaces*.

The CONFIG* DATADVH File

The CONFIG* DATADVH file is created with the entry keywords as shown in Table 11. DirMaint allows entries in multiple CONFIG* DATADVH files, therefore, it is not necessary to duplicate the entire file to supplement or override a single line.

CONFIG* DATADVH File Example

An example of a CONFIG* DATADVH file is shown in Figure 23. 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= DVHTEST1  DEST= DVHTEST2 S= DIRMAINT  T= DVHTEST2
13 PW_MIN_LENGTH= 3
14 PW_REUSE_HASHING_EXIT=
15 PW_REUSE_INTERVAL=
16 SAMPL_LINESIZE_140A= 222
17 SAMPL_LINESIZE_150A= 222
18 DVHSAPI_END_MSG.DVHSCU3541I= DVHREQ2289I DVHSHN3430I
19 DVHSAPI_ENTER_KEY_ACTION= END
```

Figure 23. CONFIG DATADVH File*

Table 11. 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.

Table 11. Summary of CONFIG DATADVH File Entries (continued)*

Entry Keyword	Function
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.
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. **Note:** If the IBM supplied default of 365 DAYS is changed, you need to enable a PASSWORD CHANGE NOTIFICATION EXIT = DVHXPN EXEC statement in the CONFIG* DATADVH file.
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.
18 DVHSAPI_END_MSG.*message*=	Identifies user tailorable choices for when the DVHSAPI routine exits and returns control back to the calling application. The default, if no entries are specified, is to end when message DVHREQ2289I is received.
19 DVHSAPI_ENTER_KEY_ACTION= END | IGNORE	Specifies whether pressing the ENTER key either terminates DVHSAPI or is ignored. The default (for compatibility) is END.

Table 11. Summary of CONFIG DATADVH File Entries (continued)*

Entry Keyword	Function

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 24. CONFIG DATADVH File

```
.
.
.
1 LOADABLE_USER_FILE= DVHFILE EXEC
1 LOADABLE_USER_FILE= DVHADZ EXEC
2 USER_EXIT= MYPXR EXEC
.
.
.
```

Figure 25. CONFIGAA DATADVH File

The following notes are to help you with your Figure 24 and Figure 25.

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 26. REQUIRED_USER_FILE= Entries

The following notes are to help you with “The REQUIRED_USER_FILE= Entries” on page 110.

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 27. 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 command parsing files should be used when the user has not entered a DIRM DEFAULTS CMDLEVEL command to select their own default CMDLEVEL.

DirMaint supports two command levels:

- The 150A level provides all of the function supported in DirMaint using the 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 DirMaint Release 4 that remains in the DirMaint feature, 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, NVAS, RACF, and so forth to run without changes, even if the administrator and the general user population exploit the full capabilities of DirMaint.

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 DirMaint 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 28. National Language Support

The following notes are to help you with “National Language Support” on page 113.

Notes:

1. Each command level has its own set of online directions, Help files, menus, and messages. DirMaint supports translation of each of these varieties of information into any left-to-right language whose character set is supported by z/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
....._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_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.

   If your site needs to create overrides, enter:

   ```
   ....._USER_MSGS_140A= LCLAUSER MSGADVH
   ....._USER_MSGS_150A= LCLAUSER MSGADVH
   ```

   and include them in a separate CONFIG* DATADVH file with a name (CONFIGZZ perhaps) that will be searched before the IBM-supplied file.

   For more information, see “Overriding and Supplementing DirMaint Messages” on page 59.

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 125.

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,

```
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 z/VM 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 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 sets on your system. This is necessary if you want to allow users to enter commands other than privilege class G or General user commands.

Command Classes

DirMaint employs a command set structure similar to the CP command privilege structure. A specific command can be placed in one or more command sets and a specific user can be authorized to enter one or more command sets.

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 sets are represented by alphanumeric characters.

With DirMaint 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 provides the following nine command sets at installation:

Table 12. 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
P	DASD management automated Programs, such as DFSMS/VM®
S	Support programmer
Z	Internal communication

Note: Additional command classes can be defined if specific command subsets are required.

Command Sets on the DIRMAINT Server

When the transaction is received on the DirMaint server, the command set 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 set against their user directory entry.

Defining a New Command Set

The shipped command sets may be inappropriate for your local installation. DirMaint allows administrators to define a custom set of commands and assign them to locally defined command sets. This command set is then treated as a shipped command set by the DirMaint installation.

Custom command sets are particularly useful for allowing users to use a few commands from a potentially dangerous set 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 set A, but several other administrative commands also share the command set A.

One alternative is to allow your POSIX administrator to use command set A and to trust that they will only use the POSIXINFO command. A better solution is to define a new command set and place the POSIXINFO command in this new set. Then authorize the POSIX administrator to use the new set.

Command sets are established in the 150CMDS DATADVH and 140CMDS DATADVH files. Each file corresponds to the command level being used.

Note: 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 sets associated with the command. As you can see, POSIXINFO is considered a command set A command. To define a new set simply place an alphanumeric character adjacent to the existing command sets. You should choose a set that is not being used as an IBM default. You should also note that the command set field must be contiguous, with no imbedded blanks. The position you place the new set is unimportant but placing it on the end is recommended.

As all IBM-supplied defaults are alphabetic, the following example will use command set 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.......5
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 command set 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 150A. The following step should be done:

Table 13. Granting a User Class A Authority

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`

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 14. 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.

Chapter 9. Exit Routines

This chapter describes each of the exits available with DirMaint. 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, 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.

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
- DVHCXA

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
- One of the following, according to the command:
  - DVHDXF for AMDISK FORMAT processing
  - DVHDXD for CLONEDISK processing
  - DVHDXP for CMDISK processing
  - DVHDXC for CMDISK processing of a CMS formatted disk
  - DVHDXN for CMDISK processing of a non-CMS formatted disk
  - DVHDXE for DMDISK CLEAN processing.
- 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:

All DVHXRC, DVHXRB, those listed below for each command, DVHXRA

ACCOUNT DVHXAV, DVHXAN

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

LOGONBY DVHXLB

MDISK DVHXMP, DVHXMN

POSIXFSROOT DVHXPESM

POSIXGLIST DVHXPESM

POSIXGROUP DVHXPESM

POSIXINFO DVHXPESM

POSIXIUPGM DVHXPESM

POSIXIWDIR DVHXPESM

PURGE DVHXMN - for each disk, DVHXDN - for each disk, DVHXLN - for each old link, DVHXAN, DVHXPN, DVHXUN.

PW DVHPXV, DVHXPN

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
STAG	DVHXTA
SUBSCRIBE	DVHXNE
TESTPW	DVHPXV
TMDISK	DVHXFA - new, DVHXMP - new, DVHXDN - new, DVHXMN - new, DVHXMN - old, DVHXDN - old

Exit Routines Summary

Table 15 summarizes the DirMaint 150A exit routines. You can find more information about each exit routine by referring to the referenced page.

Table 15. Exit Routines Summary

Exit Routine	Function	Machine Environment	IBM Supplied Sample	Page
DVHCXA	Command exit, after processing	User	Yes	130
DVHCXB	Command exit, after parsing, before processing	User	No	131
DVHCXC	Command exit, before parsing	User	Yes	132
DVHDXC	DATAMOVE COPY CMS exit	DATAMOVE	No	133
DVHDXD	DATAMOVE DDR exit	DATAMOVE	Yes	134
DVHDXE	DATAMOVE ERASE exit	DATAMOVE	Yes	136
DVHDXF	DATAMOVE FORMAT exit	DATAMOVE	No	137
DVHDXN	DATAMOVE COPY NONCMS exit	DATAMOVE	No	138
DVHDXP	DATAMOVE non-CMS disk copying exit	DATAMOVE	No	139
DVHESMLR	External Security Manager log recording exit	User DIRMAINT DATAMOVE DIRMSAT	Yes	141
DVHPXA	User's logon password exit, after transmission to DIRMAINT	User	Yes	142
DVHPXR	Random password generation exit for logon	User DIRMAINT	Yes	143 146
DVHPXV	User's logon password exit, syntax verification	User DIRMAINT	Yes	144
DVHXAN	Account number notification exit	DIRMAINT	No	147

Table 15. Exit Routines Summary (continued)

Exit Routine	Function	Machine Environment	IBM Supplied Sample	Page
DVHXAV	Account number verification exit	DIRMAINT	Yes	148
DVHXCP	Check user privilege exit	DIRMAINT	No	149
DVHXDA	DASD authorization checking exit	DIRMAINT	No	150
DVHXDN	DASD notification exit	DIRMAINT	No	152
DVHXFA	FOR authorization checking exit	DIRMAINT DATAMOVE DIRMSAT	No	154
DVHXLA	Link authorization checking exit	DIRMAINT	No	155
DVHXLB	LOGONBY change notification exit	DIRMAINT	No	156
DVHXLF	Log record filtering exit	DIRMAINT DATAMOVE DIRMSAT	Yes	157
DVHXLN	Link notification exit	DIRMAINT	No	158
DVHXLVL	Pre-startup exit for switching service levels	DIRMAINT DATAMOVE DIRMSAT	Yes	159
DVHXMN	Minidisk password change notification exit	DIRMAINT	Yes	160
DVHXMP	Minidisk password syntax verification exit	DIRMAINT	No	161
DVHXMU	MULTIUSER authorization checking exit	DIRMAINT DATAMOVE DIRMSAT	Yes	162
DVHXNE	Asynchronous update notification exit	DIRMAINT	Yes	163
DVHXPA	External security manager password authentication exit	DIRMAINT	Yes	164
DVHXPESM	POSIX change notification exit	DIRMAINT	No	165
DVHXPN	Password change notification exit	DIRMAINT	No	166
DVHXPROF	Post-profile exit for the DirMaint service machines. The exit name, DVHXPROF, must **not** be renamed.	DIRMAINT	Yes, see Figure 29 on page 213	213
DVHXPP	Password notice print exit	DIRMAINT	Yes	167
DVHXRA	Request after processing exit	DIRMAINT DATAMOVE DIRMSAT	No	168
DVHXRB	Request after parsing, before processing exit	DIRMAINT DATAMOVE DIRMSAT	No	169
DVHXRC	Request before parsing exit	DIRMAINT DATAMOVE DIRMSAT	No	171
DVHXTA	Local STAG authorization exit	DIRMAINT	No	172
DVHXTP	Backup tape mount exit	DIRMAINT	Yes	173
DVHXUN	User ID change notification exit	DIRMAINT	No	175

DirMaint 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

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 108.

Interface Parameter

This exit is called with these 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 | ...'. This exit 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

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 108.

Interface Parameter

This exit 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 | ...'. This exit 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. COMMAND_BEFORE_PROCESSING_USER_EXIT

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'. This exit 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

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 108.

Interface Parameter

This exit 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. COMMAND_BEFORE_PARSING_USER_EXIT

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'. This exit 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.

DATAMOVE CMS Copying (DVHDXC)

Environment

DATAMOVE service machine

Description

DATAMOVE 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_COPY_CMS_EXIT= DVHDXC EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Interface Parameter

This exit 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 address
- The target system affinity, usually *
- The block size with which the destination disk should be formatted: 512, 1024, 2048, 4096, or an '=' sign to use the same block size as the source disk. If both the block size and the label are omitted, an '=' sign is presumed.
- The label with which the destination disk should be formatted; or an '=' sign to use the same label as the source disk. If omitted, an '=' sign is presumed.

Return Codes

This routine must exit with one of these:

Table 18. DATAMOVE_COPY_CMS_EXIT Return Codes

Return Code	Meaning
0	Processing complete. DATAMOVE skips the remainder of its normal processing for this function and reports the successful completion back to DIRMAINT.
30 - Nop	DATAMOVE processing continues as if the exit routine were not even present. If DFSMS is installed and the destination disk is at least as large as the source disk, DATAMOVE will use DFSMS COPY to format the disk and copy the data. Otherwise, DATAMOVE will use a FORMAT command to format the disk and then a COPYFILE command to copy the data.
Other nonzero	An error has occurred. The exit routine has already issued the appropriate error messages. DATAMOVE will report the failure back to DIRMAINT.

DATAMOVE DDR Processing (DVHDXD)

Environment

DATAMOVE service machine

Description

DATAMOVE DDR processing exit.

Invocation

The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.:

```
DATAMOVE_DDR_EXIT= DVHDXD EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Interface Parameter

This exit is called with the following parameters:

- The source virtual address of the DATAMOVE machine
- A left parenthesis
- Device characteristics for the source disk, consisting of the volume label, device type, starting cylinder or block number (or the keyword 'START'), and the size of the disk in cylinders or blocks (or the keyword 'END')
- A right parenthesis
- The destination virtual address of the DATAMOVE machine
- A left parenthesis
- Device characteristics for the destination disk, consisting of the volume label, device type, starting cylinder or block number (or the keyword 'START'), and the size of the disk in cylinders or blocks (or the keyword 'END')
- A right parenthesis

Return Codes

This routine must exit with one of these:

Table 19. DATAMOVE_DDR_EXIT Return Codes

Return Code	Meaning
0	Processing complete. DATAMOVE skips the remainder of its normal processing for this function and reports the successful completion back to DIRMAINT.
30 - Nop	DATAMOVE processing continues as if the exit routine were not even present. DATAMOVE will use DDR to perform the copy.
95	A COMMAND RESULTS LOST response was received for the issued CP FLASHCOPY command. DATAMOVE skips the remainder of its normal processing for this function and reports the successful completion back to DirMaint.
96	The target disk specified in the DIRMaint CLONEDisk command is a space-efficient volume. CLONEDISK to a space-efficient target disk is not available because DirMaint does not implement IBM FlashCopy/SE support, and DDR copy to all extents of the space-efficient target would defeat the purpose of space-efficient DASD (which is to copy as little data as possible).
296-297	The source or target disk specified in the DIRMaint CLONEDisk command is being used in an existing FLASHCOPY operation. DATAMOVE adds the associated workunit to the DATAMOVE retry queue. The workunit is retried based on the DMVCTL WAKEUP schedule configured in the DATAMOVE DATADVH configuration file.
Other nonzero	An error has occurred. The exit routine has already issued the appropriate error messages. DATAMOVE will report the failure back to DIRMAINT.

DATAMOVE ERASE Processing (DVHDXE)

Environment

DATAMOVE service machine

Description

DATAMOVE ERASE processing exit.

Invocation

The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.:

```
DATAMOVE_ERASE_EXIT= DVHDXE EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Interface Parameter

This exit is called with the following parameters:

- The virtual address of the DATAMOVE machine
- The former owner's userid
- The former owner's virtual address
- The target system affinity, usually *

Return Codes

This routine must exit with one of these:

Table 20. DATAMOVE_ERASE_EXIT Return Codes

Return Code	Meaning
0	Processing complete. DATAMOVE skips the remainder of its normal processing for this function and reports the successful completion back to DIRMAINT.
30 - Nop	DATAMOVE processing continues as if the exit routine were not even present. DATAMOVE will use a CPFMTXA command to overwrite the disk space with binary zeros.
Other nonzero	An error has occurred. The exit routine has already issued the appropriate error messages. DATAMOVE will report the failure back to DIRMAINT.

DATAMOVE FORMAT Processing (DVHDXF)

Environment

DATAMOVE service machine

Description

DATAMOVE FORMAT processing exit.

Invocation

The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.:

```
DATAMOVE_FORMAT_EXIT= DVHDXF EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Interface Parameter

This exit is called with the following parameters:

- The virtual address of the DATAMOVE machine
- The target ID
- The target address
- The target system affinity, usually *
- The block size with which the destination disk should be formatted: 512, 1024, 2048, 4096, or an '=' sign to use the system default. If both the block size and the label are omitted, an '=' sign is presumed.
- The label with which the destination disk should be formatted – or omitted to use a default consisting of the leftmost characters of the userid followed by the target address.

Return Codes

This routine must exit with one of these:

Table 21. DATAMOVE_FORMAT_EXIT Return Codes

Return Code	Meaning
0	Processing complete. DATAMOVE skips the remainder of its normal processing for this function and reports the successful completion back to DIRMAINT.
30 - Nop	DATAMOVE processing continues as if the exit routine were not even present. DATAMOVE will use a FORMAT command to format the disk use with the CMS file system.
Other nonzero	An error has occurred. The exit routine has already issued the appropriate error messages. DATAMOVE will report the failure back to DIRMAINT.

DATAMOVE non-CMS Copying (DVHDXN)

Environment

DATAMOVE service machine

Description

DATAMOVE CMS non-CMS disk copying exit.

Note: DVHDXP (the DATAMOVE_NONCMS_COPYING_EXIT) is retained for compatibility with earlier function levels of DirMaint, and may be removed in a future release. It is recommended that you use this exit (DVHDXN, the DATAMOVE_COPY_NONCMS_EXIT) instead.

Invocation

The following entry should be entered in the text file in upper case as required in the CONFIG* DATADVH file.:

```
DATAMOVE_COPY_NONCMS_EXIT= DVHDXN EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Interface Parameter

This exit 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 address
- The target system affinity, usually *
- The block size with which the destination disk should be formatted: 512, 1024, 2048, 4096, or an '=' sign to use the same block size as the source disk. If both the block size and the label are omitted, an '=' sign is presumed.
- The label with which the destination disk should be formatted; or an '=' sign to use the same label as the source disk. If omitted, an '=' sign is presumed.

Return Codes

This routine must exit with one of these:

Table 22. DATAMOVE_COPY_NONCMS_EXIT Return Codes

Return Code	Meaning
0	Processing complete. DATAMOVE skips the remainder of its normal processing for this function and reports the successful completion back to DIRMAINT.
30 - Nop	DATAMOVE processing continues as if the exit routine were not even present. DATAMOVE will treat this as a failure.
Other nonzero	An error has occurred. The exit routine has already issued the appropriate error messages. DATAMOVE will report the failure back to DIRMAINT.

DATAMOVE non-CMS Copying (DVHDXP)

Environment

DATAMOVE service machine

Description

DATAMOVE non-CMS disk copying exit.

Note: This exit is retained for compatibility with earlier function levels of DirMaint FL 4.1.0, and may be removed in a future release. It is recommended that you use DVHDXN (the DATAMOVE_COPY_NONCMS_EXIT) instead.

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 28.

Interface Parameter

This exit 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 these:

Table 23. DATAMOVE_NONCMS_COPYING_EXIT Return Codes

Return Code	Meaning
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, otherwise just W. 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

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 28.

Interface Parameter

This exit 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

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
```

Interface Parameter

This exit 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

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 108.

For the DIRMAINT service machine:

```
PASSWORD_RANDOM_GENERATOR_EXIT= DVHPXR EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Interface Parameter

This exit 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 these:

Table 24. PASSWORD_RANDOM_GENERATOR_EXIT 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

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 108.

For the DIRMAINT service machine:

```
PASSWORD_SYNTAX_CHECKING_EXIT= DVHPXV EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Interface Parameter

This exit 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 (USER or IDENTITY).
- 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 these:

Table 25. PASSWORD_SYNTAX_CHECKING_USER_EXIT Return Codes

Return Code	Meaning
0	The password is accepted. No further checking is done.

Table 25. PASSWORD_SYNTAX_CHECKING_USER_EXIT Return Codes (continued)

Return Code	Meaning
30 - Nop	The DVH3276E message will be displayed indicating that the exit routine was not located.
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

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 28.

For the user's virtual machine.

```
PASSWORD_RANDOM_GENERATOR_USER_EXIT= DVHPXR EXEC
```

For more information, see “The CONFIG* DATADVH File” on page 108.

Interface Parameter

This exit 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 these:

Table 26. PASSWORD_RANDOM_GENERATOR_EXIT 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

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 28.

Interface Parameter

This exit 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 (USER, IDENTITY, SUBCONFIG or PROFILE).
- 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 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 28.

Interface Parameter

This exit is called with the following parameters:

- The command causing this check: ACCOUNT, ADD, CHNGID, or SETACNT; or ACNTADD or ACNTDEL.
- The target ID class (USER, IDENTITY, SUBCONFIG or PROFILE).
- 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 27. ACCOUNT_NUMBER_VERIFICATION_EXIT 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

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 28.

Interface Parameter

This exit is called with the following parameters:

- The commands making this check: ADD, CHNGID, PW, PWMON, PW?, SETPW
- The target ID class (USER, IDENTITY or SUBCONFIG)
- The target ID
- The target system affinity, usually *.

Return Codes

This routine must exit with one of these:

Table 28. CHECK_USER_PRIVILEGE_EXIT Return Codes

Return Code	Meaning
0	The user is a GENERAL user.___ Use the PWINTERVALFORGEN rules.
1	The user is a PRIVILEGED user.___ Use the PWINTERVALFORPRIV rules.
30	Reserved for exit routine not found.___ Use the PWINTERVALFORGEN 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

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 28.

Interface Parameter

This exit 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 (USER, IDENTITY or SUBCONFIG)
- 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 these:

Table 29. DASD_AUTHORIZATION_CHECKING_EXIT 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

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 28.

Interface Parameter

This exit 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 (USER, IDENTITY or SUBCONFIG)
- 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

Values other than 0 or 30 will terminate processing of the request.

Usage Notes

1. 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.
2. While DirMaint provides RACF support for such DASD-related commands as AMDISK, CLONEDISK, and DMDISK, please note that DirMaint does *not* currently provide RACF support for the LINK command.
3. The IBM supplied sample routine will issue RAC RDEFINE and RAC RDELETE commands as needed when called by ADD, AMDISK, CLONEDISK, DMDISK, or PURGE processing. (If the entry contains an ACIGROUP statement, the

ACIGROUP name will prefaced to the resource profile name.) For RAC RDEFINE command, defaults will be taken from the following entry in the CONFIG* DATADVH file(s):

```
DVHXDN_RDEFINE_VMMDISK_DEFAULTS= UACC(NONE) AUDIT(ALL(READ))
```

Note that the following entry in the CONFIG* DATADVH file(s) can be used to change the way DVHXDN interprets the RACF return code 4:

```
TREAT_RAC_RC.4=n
```

where *n* can be 0 (successful) or 30 (RACF not installed). The default is 4, which denotes no change in interpretation.

FOR Authorization Checking (DVHXFA)

Environment

DIRMAINT service machine

DATAMOVE service machine

DIRMSAT service machines

Description

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 28.

Interface Parameter

This exit 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)
- The target node ID (ATNODE if specified, otherwise an asterisk)
- The command level
- The command set(s) required to issue the command
- The command name and any other command parameters

Return Codes

This routine must exit with one of these:

Table 30. FOR_AUTHORIZATION_CHECKING_EXIT 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

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 28.

Interface Parameter

This exit is called with the following parameters:

- The command causing this check: LINK
- The target ID class (USER, IDENTITY or SUBCONFIG)
- 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 31. LINK_AUTHORIZATION_EXIT 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.

LOGONBY Change Notification (DVHXLB)

Environment

DIRMAINT service machine

Description

The LOGONBY change notification exit may be used to notify other service machines of RACF updates associated with the DIRM LOGONBY command.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
LOGONBY_CHANGE_NOTIFICATION_EXIT= DVHXLB EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Interface Parameter

This exit is called with the following parameters:

- The command causing this notification: LOGONBY
- The target ID class (PROFILE or USER)
- The target ID
- The target system affinity, usually *
- The ADD or DELETE parameter associated with the LOGONBY command
- The userid or list of userids as parameters for the LOGONBY command.

Return Codes

Values other than 0 or 30 will terminate processing of the request.

Usage Notes

1. The IBM supplied sample routine will issue RACF commands as needed when called by LOGONBY processing. For RAC RDEFINE, defaults will be taken from the following entry in the CONFIG* DATADVH file(s):

   ```
   RACF_RDEFINE_SURROGAT_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
   ```

   Note that the following entry in the CONFIG* DATADVH file(s) can be used to change the way DVHXLB interprets the RACF return code 4:

   ```
   TREAT_RAC_RC.4=n
   ```

   where *n* can be 0 (successful) or 30 (RACF not installed). The default is 4, which denotes no change in interpretation.

Message Logging Filter (DVHXLF)

Environment

DIRMAINT service machine

DATAMOVE service machine

DIRMSAT service machines

Description

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 28.

Interface Parameter

This exit 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 these:

Table 32. MESSAGE_LOGGING_FILTER_EXIT 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

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 28.

Interface Parameter

This exit 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 (USER, IDENTITY or SUBCONFIG)
- 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.

Pre-startup Exit for Switching Service Levels (DVHXLVL)

Environment

DIRMAINT service machine

DATAMOVE service machine

DIRMSAT service machines

Description

This is a pre-startup exit routine to enable switching between a production service level and a test service level. As supplied by IBM, it provides options to set up DirMaint to use default, test, or production disks. This is useful for testing a specific service level before moving the level to production.

Invocation

The following entry should be entered in the PROFILE EXEC file before calling DVHPROF EXEC:

```
EXEC DVHXLVL
```

The routine can also be called from the command line.

Interface Parameter

This exit is called with the following parameter:

- The choice of disk set to link:

  DFLT To initialize using the directory defaults. This is the default if no parameter is specified.

  PROD To initialize using the PRODUCTION disks.

  TEST To initialize using the TEST disks.

Return Codes

This routine must exit with one of these:

Table 33. DVHXLVL Return Codes

Return Code	Meaning
0	Disk setup succeeded. Startup may continue.
Any nonzero	Error encountered. Initialization stops. If userid is disconnected, the routine logs it off.

Minidisk Password Notification (DVHXMN)

Environment

DIRMAINT service machine

Description

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 28.

Interface Parameter

This exit 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 (USER, IDENTITY or SUBCONFIG)
- 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

Minidisk password syntax verification exit. 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 28.

Interface Parameter

This exit is called with the following parameters:

- Command causing this check: AMDISK, CHNGID, CHVADDR, MDISK, or TMDISK
- Target ID class (PROFILE, USER, IDENTITY or SUBCONFIG)
- 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 34. MINIDISK_PASSWORD_CHECKING_EXIT 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

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 28.

Interface Parameter

This exit 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 these:

Table 35. MULTIUSER_VERIFICATION_EXIT 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.

Asynchronous Update Notification (DVHXNE)

Environment

DIRMAINT service machine

Description

Exit to notify another server when directory updates have been made.

Invocation

The following entry should be made in your local CONFIG* DATADVH supplement/override file(s):

```
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.xxxx= DVHXNE EXEC
```

where *xxxx* matches one of the supported protocols: RDR, SMSG, TCP, or UDP. If the protocol is TCP or UDP, the TCPIP DATA file (typically found on the TCPMAINT 592 disk) should be copied to the DIRMAINT 155 A disk.

Interface Parameter

This exit is called with the following parameters:

- The encoding (ASCII or EBCDIC)
- The protocol (RDR, SMSG, TCP, or UDP)
- The destination parameter 1 (userid for RDR or SMSG, IP address for TCP or UDP)
- The destination parameter 2 (nodeid or * for RDR or SMSG, port number for TCP or UDP)
- The updated userid
- The subscriber data string (an optional character or hexadecimal string supplied by the subscriber).

Return Codes

This routine must exit with one of these:

Table 36. ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT Return Codes

Return Code	Meaning
0	The notification has been sent.
8	Error – recipient not reachable.
30	NOP – notification not sent.

ESM Password Authentication (DVHXPA EXEC)

Environment

DIRMAINT service machine

Description

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 z/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= DVHXPA EXEC
```

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 the IBM supplied default (DVHXPA EXEC), the DIRMAINT machine must have OPTION DIAG88 in its directory entry to enable the DIRMAINT service machine to use DIAGNOSE X'88' for ESM password authentication. 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 28.

Interface Parameter

This exit 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 these:

Table 37. ESM_PASSWORD_AUTHENTICATION_EXIT Return Codes

Return Code	Meaning
0	The password was successfully authenticated for the provided userid.
28	The External Security Manager is not available.
30	Regular DirMaint password authentication should be done using the DMSPASS CSL routine.
Other nonzero	The password failed authentication for the provided userid.

POSIX Change Notification (DVHXPESM)

Environment

DIRMAINT service machine

Description

The POSIX change notification exit may be used to notify other service machines of RACF updates. It will be called whenever a POSIX related command is issued, such as POSIXGROUP, POSIXGLIST, POSIXINFO, POSIXOPT, POSIXIWDIR, POSIXIUPGM, or POSIXFSROOT.

Invocation

The following entry should be entered in the text file in upper case, as required in the CONFIG* DATADVH file.

```
POSIX_CHANGE_NOTIFICATION_EXIT= DVHXPESM EXEC
```

For more information, see “CONFIG DATADVH” on page 28.

Note that this exit uses the POSIX_UID_AUTO_RANGE entry (see page 35), which specifies a UID range for use during automatic assignment of POSIX UIDs to users during DIRM ADD and DIRM POSIXINFO operations.

Interface Parameter

This exit is called with the following parameters:

- The command causing this notification: POSIXGLIST, POSIXINFO, POSIXGROUP, POSIXIWDIR, POSIXIUPGM, or POSIXFSROOT
- The target ID class (USER, PROFILE, IDENTITY or SUBCONFIG)
- The target ID
- The target system affinity, usually *
- Parameters for the POSIX related command (POSIXGLIST, POSIXINFO, POSIXGROUP, POSIXIWDIR, POSIXIUPGM, or POSIXFSROOT).

Return Codes

Values other than 0 or 30 will terminate processing of the request.

Usage Notes

1. The IBM supplied sample routine will issue RACF commands as needed when called by POSIXGLIST, POSIXINFO, POSIXGROUP, POSIXIWDIR, POSIXIUPGM, or POSIXFSROOT processing. For RAC RDEFINE, defaults will be taken from the following entries in the CONFIG* DATADVH file(s):

   ```
   RDEFINE_VMPOSIX_POSIXOPT.QUERYDB= UACC(READ)
   RDEFINE_VMPOSIX_POSIXOPT.SETIDS= UACC(NONE)
   ```

   Note that the following entry in the CONFIG* DATADVH file(s) can be used to change the way DVHXPESM interprets the RACF return code 4:

   ```
   TREAT_RAC_RC.4=n
   ```

   where *n* can be 0 (successful) or 30 (RACF not installed). The default is 4, which denotes no change in interpretation.

Password Change Notification (DVHXPN)

Environment

DIRMAINT service machine

Description

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

Note: You need to enable this statement if the IBM supplied default of 365 DAYS is changed in the PW_REUSE_INTERVAL statement.

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 28.

Interface Parameter

This exit 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 (USER or IDENTITY)
- The target ID
- The target system affinity, usually *.
- The new or old password.

Return Codes

Values other than 0 or 30 will terminate processing of the request.

Usage Notes

1. The IBM-supplied sample will issue RACF ALTUSER commands using the PASSWORD and PHRASE options as needed when called by PW or SETPW command processing.

Password Notice Printing (DVHXPP)

Environment

DIRMAINT service machine

Description

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 28.

Interface Parameter

This exit 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 these:

Table 38. PW_NOTICE_PRT_EXIT 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 PWNOTICEPRTCLASS= 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

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 28.

Interface Parameter

This exit 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 | ...'`. This exit 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

Request after parsing, before processing exit.

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 28.

Interface Parameter

This exit is called with no parameters, but with two interface variables set:

_CMDSTRING
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.

_LOGSTRING
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_' variablename 2 | .... This exit is called indirectly 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 39. REQUEST_BEFORE_PROCESSING_EXIT 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'. This exit 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 28.

Interface Parameter

This exit 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 40. REQUEST_BEFORE_PARSING_EXIT 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'. This exit 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

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 28.

Interface Parameter

This exit is called with the following parameters:

- The command causing this check: SETSTAG or STAG
- The target ID class (USER, PROFILE, IDENTITY or SUBCONFIG)
- 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 these:

Table 41. LOCAL_STAG_AUTHORIZATION_EXIT 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

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 28.
```

Interface Parameter

This exit is called with the following parameters:

- Request type one of these:

  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 these:

Table 42. BACKUP_TAPE_MOUNT_EXIT 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: To avoid the DirMaint machine waiting for a tape mount, 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, 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

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.

Note: The IBM-supplied DVHXUN exit changes the user's password in the External Security Manager when a user or identity is being added.

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 28.

Interface Parameter

This exit is called with the following parameters:

- The command causing this notification: ADD, CHNGID-NEW, CHNGID-OLD, or PURGE
- The target ID class (USER, PROFILE, IDENTITY or SUBCONFIG)
- The target ID
- The target system affinity, usually *.
- `Oldid`, old ID information passed for CHNGID processing
- `OldPw`, old password information for CHNGID processing.
- `Otherinfo`, old information related to the old ID

Return Codes

Values other than 0 or 30 will cause the associated ADD, CHNGID, or PURGE request to fail with error message DVH3288E. See *z/VM: Directory Maintenance Facility Messages* for further information on message DVH3288E.

Usage Notes

1. The IBM supplied sample will issue RACF commands as needed when called by ADD or PURGE processing. For RAC ADDUSER, the new password will be set to be the same as the directory source password, and additional defaults will be taken from one of the following two entries in the CONFIG* DATADVH file(s):

   ```
   DVHXUN_ADDUSER_DEFAULTS= UACC(NONE)
   RACF_ADDUSER_DEFAULTS= UACC(NONE)
   ```

   For RAC RDEFINE, defaults will be taken from the following entries in the CONFIG* DATADVH file(s):

```
RDEFINE_VMPOSIX_POSIXOPT.QUERYDB= UACC(READ)
RDEFINE_VMPOSIX_POSIXOPT.SETIDS= UACC(NONE)
RACF_RDEFINE_SURROGAT_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
RACF_RDEFINE_VMBATCH_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
RACF_RDEFINE_VMRDR_DEFAULTS= UACC(NONE) AUDIT(FAILURES(READ))
```

Note that the following entry in the CONFIG* DATADVH file(s) can be used to change the way DVHXUN interprets the RACF return code 4:

```
TREAT_RAC_RC.4=n
```

where *n* can be 0 (successful) or 30 (RACF not installed). The default is 4, which denotes no change in interpretation.

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; while 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, while 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 *restartable*. 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.

Product Specific Program Interface

The INTERFACE variable and (for internal interface level 199501) the ASUSER, ATNODE, BYUSER, FORUSER, IMMED, LANG, TEST, TOSYS, TRACE, and CMDSET variables are considered to be part of the product specific program interface to the various exit routines (DVHCXB, DVHCXA, DVHPXR, DVHPXV, and DVHPXA) that run in the user's virtual machine. None of these variables are intended for use outside of the product or these exit routines.

A new internal 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 DVHCXB or DVHPXR.

A new internal interface level descriptor will also be assigned 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 when any changes are made in the format of any data files that are intended for local tailoring.

General Program Interface

Neither DIRMAINT 1.4.0 nor DirMaint 1.5.0 provide a true General Programming Interface.Significant effort has been made to keep DirMaint 1.5.0 upwards compatible with DIRMAINT 1.4.0 in critical areas; and similar efforts will be made to keep follow-on releases upwards compatible with DirMaint 1.5.0. These critical areas include:

1. Command Syntax:

   Most valid DIRMAINT 1.4 commands which were expected for use within application programs will be correctly processed by DirMaint 1.5.0 when running in 140A compatibility mode. The primary incompatibility is that menu support is not provided. Other differences are upwards compatible enhancements.

2. Messages:

   Since this is a complete re-implementation of the product, it is strictly a coincidence if any messages are the same. The specific conditions which resulted in a particular 1.4 message may or may not arise in 1.5, and the required corrections may or may not be the same. However, DirMaint 1.5 has been specifically designed to enable the message handling routines to substitute: a specific return code, calling module identification, alternate message number and severity, alternate variable substitutions, and alternate message text; for any message which is issued. It not necessary to either copy entire primary message repositories or to modify the existing primary repositories to enable this capability. A tailored subset of the repositories may contain only the messages needing adjustments.

3. Return Codes:

   Since this is a complete re-implementation of the product, it is strictly a coincidence if any return codes are the same. The specific conditions which resulted in a particular 1.4 return code may or may not arise in 1.5, and the required corrections may or may not be the same. DirMaint 1.5.0 has been specifically designed for improved automatic problem diagnosis and correction by an invoking application program, by generating a return code which corresponds to the error message issued. However, as stated above, the message handling routines may substitute an alternate return code from a tailored repository subset.

4. Data Files:

   With the exception of the USER BACKUP file, programs should not attempt to process files as they reside directly on DIRMAINT's disks or directories. Programs which do so are "at risk". Unless otherwise stated, there is no assurance that data file formats will be compatible from release to release, or even from service level to service level within a release.

   - The files obtained by issuing DIRMAINT BACKUP, DIRMAINT USER BACKUP, or DIRMAINT USER WITHPASS are suitable for input to the DIRECT or DIRECTXA commands. The MIXED (and NOMIXMSG) parameter(s) may be required.

     Since they are suitable for input to DIRECT and/or DIRECTXA, they may also be used for input to a DirMaint INITLZ command. The file contains DirMaint control information pertaining to the user entry in addition to z/VM directory information. The issuing user is responsible for either ensuring that this control information is accurate or removing obsolete control information from the file. Application programs that are not designed to process DirMaint's

control information should ignore any record beginning with "*DVH". Depending on the functions provided by the program, "ignore" may imply recognizing and deleting the control records, copying them to a new file without changes, or simply doing what needs to be done without "choking" on them.

- With the exception of the masking of passwords, the file obtained by issuing DIRMAINT USER NOPASS is also suitable for input to the DIRECT or DIRECTXA commands; possibly with the MIXED parameter. To prevent accidental destruction of the object directory with inappropriate data, the file generated by DIRMAINT USER NOPASS will include the following lines:

  ```
  DVHWARN: THIS FILE WAS CREATED BY A "DIRMAINT USER NOPASS" COMMAND.
  DVHWARN: ALL PASSWORDS HAVE BEEN MASKED.
  DVHWARN: IT IS NOT SUITABLE FOR INPUT TO DIRECT, DIRECTXA, OR INITLZ.
  DVHWARN: ISSUE "DIRMAINT USER WITHPASS" TO OBTAIN A FILE
  DVHWARN: FOR THAT PURPOSE.
  ```

  or one or more alternative lines beginning with "DVH".

  Application programs that are not designed to process DirMaint's control information should ignore any record beginning with "*DVH", and possibly those beginning with "DVH". Depending on the functions provided by the program, "ignore" may imply recognizing and deleting the control records, copying them to a new file without changes, or simply doing what needs to be done without "choking" on them.
- The files obtained by issuing DIRMAINT GET commands contain DirMaint control information pertaining to the user entry in addition to the z/VM directory information. When doing a DIRMAINT REPLACE, the issuing user is responsible for either ensuring that this control information is accurate or removing obsolete control information from the file. Application programs that are not designed to process DirMaint's control information should ignore any record beginning with "*DVH". Depending on the functions provided by the program, "ignore" may imply recognizing and deleting the control records, copying them to a new file without changes, or simply doing what needs to be done without "choking" on them.
- With the exception of the masking of passwords, the file obtained by issuing DIRMAINT REVIEW is also suitable for input to the DIRMAINT ADD or REPLACE. To prevent accidental damage to the directory with inappropriate data, the file generated by DIRMAINT REVIEW will include the following lines:

  ```
  DVHWARN: THIS FILE WAS CREATED BY A "DIRMAINT REVIEW" COMMAND.
  DVHWARN: ALL PASSWORDS HAVE BEEN MASKED.
  DVHWARN: THE REFERENCED PROFILE(S) HAVE BEEN INCLUDED INLINE.
  DVHWARN: IT IS NOT SUITABLE FOR INPUT TO DIRMAINT ADD OR REPLACE.
  DVHWARN: ISSUE "DIRMAINT GET" TO OBTAIN A FILE FOR THAT PURPOSE.
  ```

  or one or more alternative lines beginning with "DVH".

  Application programs that are not designed to process DirMaint's control information should ignore any record beginning with "*DVH", and possibly those beginning with "DVH". Depending on the functions provided by the program, "ignore" may imply recognizing and deleting the control records, copying them to a new file without changes, or simply doing what needs to be done without "choking" on them.
- The files obtained by doing a DIRMAINT FREEXT or DIRMAINT USEDEXT have not yet been defined for DirMaint 1.5.0. They may or may not be compatible with DIRMAINT 1.4.0.

These critical compatibility areas have been assigned an external compatibility level identification. The 1.4 compatibility set is designated as "140A". The full function set is designated as "150A". Any incompatible changes to these critical areas will result in a new identifier, beginning with 140B and/or 150B. Any simple additions which do not create incompatibilities with either: the command set, the message repositories, or the various tidbits of control information in the directory files; will keep the existing 140A or 150A designations, and only the internal interface level identifier will be changed as described under Product Specific Program Interface.

Users issuing DirMaint commands from the terminal are expected to be using the 150A command set; and may switch to the 140A command set prior to invoking an application program which requires this level of compatibility; and switch back to the 150A command set when the application is done. Refer to the DIRMAINT GLOBALV CMDLEVEL command. Existing application programs may be modified to include the necessary DIRMAINT GLOBALV CMDLEVEL commands; but new application programs should be written to the 150A level.

Message Numbers Available for Installation-Written Exits

In addition to the messages listed in “Guidelines for Creating or Modifying Exit Routines” on page 177, 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

These 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 43. LASTING GLOBALV Variables for the DVHCX and DVHPX* Exits*

Variable	Description
CMDLEVEL	Specifies whether the user is entering the command syntax from DirMaint 150A level or DirMaint 140A level. The valid values are 150A or 140A.
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. The valid values are AMENG, UCENG, or 1SAPI. DirMaint is enabled for additional languages.
NEEDPASS	Specifies whether the user is required to supply their password for interaction with the DIRMAINT machine. The valid values are YES or NO.
REQUEST	Keeps track of how many requests have been sent to the DIRMAINT service machine for processing. The valid values are 1 through 9999.
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. The 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.

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 44. 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.

Table 44. SESSION GLOBALV Variables for the DVHCX and DVHPX* Exits (continued)*

Variable	Description
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 200201. 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.
LOG_STRING	Specifies the command string being processed with any passwords or other sensitive data. This string should be changed to *XXXs*.
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.
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 and:

- 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 45. LASTING GLOBALV Variables for the DVHX Exits*

Variable	Description
TRACE	Specifies which routines, if any, should be traced and the degree of tracing desired. The 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.

Table 46. 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 command syntax from DirMaint 150A level or DirMaint 140A level. The valid values are 150A or 140A.
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 200201. 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. The valid values are AMENG, UCENG, or 1SAPI.
ORIGNODE	Identifies the *nodeid* where the command originated. An asterisk (*) indicates any node within the local CSE cluster.

Table 46. SESSION GLOBALV Variables for the DVHX Exits (continued)*

Variable	Description
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. The 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. The 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.

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 (DVHXRC EXEC). Those that ARE available are: CMDLEVEL, INTERFACE.DVHXRC, 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.

All IBM defined global variables are stored in the DVH15 or DVH15NDX 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 47. 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.

Table 47. Format of Housekeeping Utility Fields (continued)

Columns	Function

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 Appendix E, “WAKEUP Command,” on page 235.

Chapter 10. Planning for Diagnosis

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 *z/VM: Directory Maintenance Facility Messages*.

Displaying Service Level Information

When solving a DirMaint problem, you may sometimes need to provide IBM with service level information for several or all DirMaint executable modules. IBM supplies DVHSERVL EXEC to help you collect that information. To obtain a report, you must first add one or more SERVICE_LEVEL_INFO configuration statements to a CONFIG* DATADVH file as follows:

```
SERVICE_LEVEL_INFO= ALL | fn EXEC | fn  XEDIT | fn REXX | fn MODULE | CONFIG
```

Then, whenever you wish to see the current service level information, either issue `DIRM CMS DVHSERVL` (from a user ID authorized for the DIRM CMS command) or issue `DVHSERVL` from the DIRMAINT machine command line when the server is not running.

After processing the file set specified with the SERVICE_LEVEL_INFO= configuration statement, DVHSERVL will return a report in the following format:

```
DVHSRV3505I Service Level Information for DirMaint Modules

File                 Part              Status   PITS Status  APAR
-------------------  ----------------  -------  -----------  ---------
filename filetype fm  partname parttype  status   @pitsstatus  @VAnumber
```

where:

filename	is the file name.
filetype	is the file type.
fm	is the file mode.
partname	is the part name.
parttype	is the part type.
status	is the status of the part.
@*pitsstatus*	is the latest PITS or line item changes applied to the part identifier.
@VA*numbr*	is the latest APAR changes applied to the part identifier.

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. his can be done using an External Security Manager (ESM) 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 z/VM systems with external security manager programs other than RACF. The methods of defining the DirMaint product, data disks, and 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 z/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 z/VM, see these publications:

- *z/VM: RACF Security Server Auditor's Guide*
- *z/VM: RACF Security Server Command Language Reference*
- *z/VM: RACF Security Server Diagnosis Guide*
- *z/VM: RACF Security Server General User's Guide*
- *z/VM: RACF Security Server Macros and Interfaces*
- *z/VM: RACF Security Server Messages and Codes*
- *z/VM: RACF Security Server Security Administrator's Guide*
- *z/VM: RACF Security Server System Programmer's Guide*
- *z/VM: Security Server RACROUTE Macro Reference*

Commands

RACF Commands

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
```

CP Commands

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

CMS Commands

Note that when RACF is enabled, DirMaint may ignore CMS commands while doing some asynchronous processing. Use DIRM IMMED to send CMS commands directly to DirMaint. See *z/VM: Directory Maintenance Facility Commands Reference* for more information on the DIRM IMMED command.

RACF-Specific Characteristics in the CONFIG DATADVH File

If your system does have RACF installed as the ESM, note that there are several entries in the CONFIG DATADVH file that will set defaults for exit calls, RACF commands, and other characteristics for RACF functions. See “Step 5. Select RACF-Specific Characteristics” on page 39 for more information.

Enabling Auditing Using RACROUTE

You may enable the DirMaint service machines: DIRMAINT, DATAMOVE, and DIRMSAT to record information in the RACF audit trail.

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)
```

Where:

xxxxxxxx
: Identifies the user ID of the DirMaint service machine.

Note: These commands may fail if they have already been issued before.

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. 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. 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 one of the following conditions is true:

- The RPIUCMS MODULE resides on a public disk, or
- The DirMaint service machines already have UACC (READ) access to that disk, or
- The service machines have been made exempt

then the service machines must be permitted for READ access to that disk. To do that, 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.

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

The DirMaint service machines (DIRMAINT, DATAMOVEs and DIRMSATs) should be made exempt from access checking. Even if access checking 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 access 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(RSTDSEG/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(MDISK/NOCTL)
RAC RALTER   VMXEVENT USERSEL.xxxxxxxx ADDMEM(APPCPWVL/NOCTL)
```

Where:

xxxxxxxx
Identifies user ID of the DirMaint service machine to be made exempt.

Note: These commands may fail if they have already been issued before.

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.

To view the list of events, enter:

```
RAC SETEVENT LIST USERSEL.xxxxxxxx
```

Enabling DirMaint to Access DIAGNOSE X'88'

You must enable the DirMaint service machine for DIAGNOSE X'88' access. If RACF is being used to control DIAGNOSE X'88' access, enable DIAGNOSE X'88' access for DirMaint by completing the following steps:

Step 1. Enable RACF/VM profile protection for DIAGNOSE X'88':

1. Confirm that there are no members called DIAG088/NOCTL in the active VMXEVENT profile:

   ```
   RAC SETEVENT LIST USERSEL.xxxxxxxx
   ```

2. Create a profile called DIAG088 in the VMCMD class with a default access of NONE:

   ```
   RDEFINE VMCMD DIAG088 UACC(NONE)
   ```

3. Ensure that the VMCMD class is active:

   ```
   SETROPTS CLASSACT(VMCMD)
   ```

Note: If you do not enable RACF profile protection, the DIRMAINT server must be defined with OPTION DIAG88 in its directory entry.

Step 2. Give the DIRMAINT server permission to perform password validation using DMSPASS (which uses DIAGNOSE X'88' subcode 8):

```
PERMIT DIAG088 CLASS(VMCMD) ID(DIRMAINT) ACCESS(READ)
```

For more information, see *z/VM: RACF Security Server Security Administrator's Guide*.

Enabling DirMaint to Access User's Minidisks and Readers

You must enable the DirMaint service machine for minidisk and reader access.

Minidisk Access

If RACF is being used to control minidisk access, enable minidisk access for DirMaint by completing the following steps:

Step 1. Access DirMaint primary interface files, enter:

```
RAC RALTER VMMDISK 6VMDIR20.11F UACC(READ)
```

Step 2. Access the secondary interface files and help files for testing, enter:

```
RAC RALTER VMMDISK 6VMDIR20.41F UACC(READ)
RAC RALTER VMMDISK 6VMDIR20.29E UACC(READ)
RAC RALTER VMMDISK 6VMDIR20.29D UACC(READ)
```

Step 3. Permit the DirMaint service machines to the necessary disks, unless they have been made exempt, enter:

```
RAC PERMIT 6VMDIR20.491 CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT 6VMDIR20.492 CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT 6VMDIR20.11F CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT 6VMDIR20.41F CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT 6VMDIR20.29E CLASS(VMMDISK) ID(DIRMAINT) ACCESS(CONTROL)
RAC PERMIT MAINT.123    CLASS(VMMDISK) ID(DIRMAINT) ACCESS(ALTER)

RAC PERMIT 6VMDIR20.491 CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
RAC PERMIT 6VMDIR20.492 CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
RAC PERMIT 6VMDIR20.11F CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
RAC PERMIT 6VMDIR20.41F CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
RAC PERMIT 6VMDIR20.29E CLASS(VMMDISK) ID(DATAMOVE) ACCESS(READ)
```

```
RAC PERMIT 6VMDIR20.491 CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT 6VMDIR20.492 CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT 6VMDIR20.11F CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT 6VMDIR20.41F CLASS(VMMDISK) ID(DIRMSAT) ACCESS(READ)
RAC PERMIT 6VMDIR20.29E 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 Access

If RACF is being used to control reader access, enable reader access for DirMaint by completing the following steps:

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 is not exempted from all reader access control, each user must 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 DATAMOVE and DIRMSAT machines are not exempted from all reader access control, 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.

Improving Performance with RACF

When DIRMAINT is heavily used, you can improve performance by adding the 6VMDIR20 disks needed by general users to the global minidisk table in HCPRWA. Add:

```
GLBLDSK  USERID=6VMDIR20,VADDR=11F,SCOPE=GLOBAL
GLBLDSK  USERID=6VMDIR20,VADDR=29D,SCOPE=GLOBAL
GLBLDSK  USERID=6VMDIR20,VADDR=29E,SCOPE=GLOBAL
GLBLDSK  USERID=6VMDIR20,VADDR=41F,SCOPE=GLOBAL
```

Do *not* 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 on adding this local modification, see "Defining Public Minidisks" in the *RACF Security Server for z/VM Program Directory*.

Enabling Mandatory Access Control for DirMaint Resources

Note: This section applies only if you are using security labels and mandatory access control (MAC).

Example—Defining the DirMaint Service Machines:

If you decide to enable access control of the DirMaint resources, you should define all DirMaint service machines (DIRMAINT, DATAMOVEs, and DIRMSATs) to RACF with a SECLABEL of SYSNONE in order to allow access to all users on the system by entering:

```
RAC PERMIT SYSNONE CLASS(SECLABEL) ID(xxxxxxxx) ACCESS(READ)
RAC ALTUSER xxxxxxxx SECLABEL(SYSNONE)
```

Where:

xxxxxxxx
Identifies the user ID of the service machine.

You can enable access control to DirMaint minidisks and readers.

Minidisk Access Control

If you decide to enable minidisk access control, perform the following steps:

Step 1. Define all of the minidisks owned by the DirMaint machines (6VMDIR20, DIRMAINT, DATAMOVEs, and DIRMSATs) with the appropriate SECLABEL. The disks needed by general users must be defined as SYSLOW. Enter:

```
RAC RALTER VMMDISK 6VMDIR20.11F SECLABEL(SYSLOW)
RAC RALTER VMMDISK 6VMDIR20.29D SECLABEL(SYSLOW)
RAC RALTER VMMDISK 6VMDIR20.29E SECLABEL(SYSLOW)
RAC RALTER VMMDISK 6VMDIR20.41F SECLABEL(SYSLOW)
```

The use of optional national language Help files also must be defined as SYSLOW disks. Enter:

```
RAC RALTER VMMDISK 6VMDIR20.xxx SECLABEL(SYSLOW)
RAC RALTER VMMDISK 6VMDIR20.xxx SECLABEL(SYSLOW)
```

The remaining product code disks may be assigned a SECLABEL of your choice. Enter:

```
RAC RALTER VMMDISK 6VMDIR20.491 SECLABEL(xxxxxxxx)
RAC RALTER VMMDISK 6VMDIR20.492 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 Access Control

If you decide to enable reader access control, 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.

Appendix B. DirMaint Support for Systems Management APIs

This appendix includes crucial information for the client programmer or the server administrator using DirMaint as the directory manager for Systems Management APIs, as documented in *z/VM: Systems Management Application Programming*.

As a starting point, this appendix presumes that DirMaint is up and operational for native CMS users, as described in this book and in the DirMaint Program Directory.

This includes coordination between DirMaint and an ESM (External Security Manager) such as RACF, if necessary.

If full-function DirMaint is enabled after using the limited access directory manager function installed for use with the System Management Application Programming Interface (SMAPI), DirMaint must be manually configured as described in this appendix in order to continue to function successfully with SMAPI.

Check the DirMaint web site:

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

for additional hints and tips.

Linking and Accessing the DirMaint Interface Disk

The owner of the VSMAPI worker servers must ensure that the DirMaint interface code is available to each worker server. In the default configuration, the worker servers would be the VSMWORK1, VSMWORK2 and VSMWORK3 user IDs. The most convenient way to do this is to include the following commands in each worker server machine's server profile exec:

```
'EXEC DIRMAINT EXECLOAD'
'EXEC DIRMAINT DEFAULT MENUS DISABLED'
'EXEC DIRMAINT DEFAULT PROMPTS DISABLED'
```

DIRMAINT should be using the PRODUCTION disks most of the time. However, when applying service, it is customary to put the service onto the TEST disks and to let DIRMAINT run with that level of service for a while (hours or days) before putting the new service onto the PRODUCTION disks. Either all worker servers should be switched to use the same interface disk that the DIRMAINT server is using, or else all worker servers should be shut down when DIRMAINT is using the TEST disks.

If the differences between the PRODUCTION and TEST disks are sufficiently minor, the VSMAPI worker servers and DIRMAINT servers may all continue working normally when the disks don't match, but this is not recommended. To switch disks, edit the ACCESS DATADVH file and change the USE= field for your system to point to the correct disk. Remember to revert to the production level of the ACCESS DATADVH file when DIRMAINT returns to use of the PRODUCTION disks.

Note: Changing the contents of the DirMaint interface disk while any of the VSMAPI worker servers are active, or changing the contents of any disk that is linked and accessed by any active server, may cause that server to encounter unpredictable results, including disk I/O errors and abends. To avoid such problems, the server(s) should be shut down or forced off while the disks are being changed, then restarted when the changes are complete.

Configuring Use of the ASUSER Prefix

Note: This section is applicable only when running the IBM-supplied VSM API server code. ALLOW_ASUSER_NOPASS_FROM= should not be used for other virtual machines running any other applications outside of the trusted API server environment.

DirMaint usually requires that users requesting directory changes must demonstrate their identity by supplying the logon password for the ID they are claiming to be, with each DirMaint directory change request. Authentication for API server requests are handled by the API server. Additional DirMaint authentication of every API server request is not needed. In order for the API server to be able to request directory changes from DirMaint, however, DirMaint must be told to trust the userid for requests made from any of the worker servers. This is accomplished by having a configuration file record for each worker server in the form of:

```
ALLOW_ASUSER_NOPASS_FROM= worker_server_x *|node_name
```

where *worker_server_x* is the name of the worker server.

This configuration file record should be placed in a supplemental configuration file. The first 6 characters of the file name are CONFIG, the 7th and 8th characters may be any valid file identification characters ("SM" is suggested and will be used as an example), and the file type must be DATADVH.

This supplemental CONFIGSM DATADVH file must reside on the DIRMAINT server's 11F disk, because the DIRM command user's virtual machine checks the ALLOW_ASUSER_NOPASS_FROM= statements.

Enabling the Asynchronous Update Notification Exit

In order to receive TCP and/or UDP asynchronous notifications for directory updates enabled by the Asynchronous_Notification_Enable_DM API, one or both of the following statements must be added to a DirMaint override configuration file:

```
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.TCP= DVHXNE EXEC
ASYNCHRONOUS_UPDATE_NOTIFICATION_EXIT.UDP= DVHXNE EXEC
```

For more information on the Asynchronous Update Notification exit, see “Asynchronous Update Notification (DVHXNE)” on page 163.

Coordination Between DirMaint and an External Security Manager

If your z/VM system is using an External Security Manager (ESM), such as RACF, some SMAPI functions require handshaking with the ESM. For example, after creating a new image with the Image_Create_DM function, the new image must be defined to the ESM before using an Image_Activate function. In order to use the Image_Disk_Create_DM or Image_Disk_Share_DM functions, the new disk must be defined to the ESM. DirMaint provides “exit points” where routines to accomplish such coordination may be enabled. Working exit routines are provided by IBM for use with RACF. For other ESMs, refer to the vendor-supplied documentation.

To configure DirMaint for use with RACF, see “Step 5. Select RACF-Specific Characteristics” on page 39, as well as Appendix A, “External Security Manager Considerations,” on page 191.

DirMaint Command Set Authorizations

The entries in the VSMWORK1 AUTHLIST file do not allow users to actually perform the Directory Manager functions; they simply allow the requests to be sent to the DIRMAINT server for further authorization checking.

DirMaint must be configured to allow Systems Management API users to use specific command sets. For more information, seeChapter 8, “Delegating Administrative Authority,” on page 121.

Remember to authorize use of both command levels 140A and 150A, especially for any user authorized for command set D.

DirMaint must be configured to allow the VSMAPI short call worker server (VSMWORK1 by default) to use the command set containing the DIRMAINT NAMESAVE command (command set A by default). This is necessary because in order to grant access to all worker servers to the VSMAPI Server_DCSS, the short call worker server executes the Shared_Memory_Access_Add_DM API in its server profile exit – and as seen in Table 48, the Shared_Memory_Access_Add_DM API issues the DIRMAINT NAMESAVE command.

If you wish to allow a given authenticated userid to perform some directory management functions but not others, you need to know which Systems Management APIs invoke which DirMaint commands, and which DirMaint command sets(s) are used by those commands. The IBM-supplied defaults are as follows:

Table 48. IBM-Supplied Defaults for DirMaint Commands and Command Sets

API	Command(s)	CMDSETs
Asynchronous_Notification_Disable_DM	SUBSCRIBE	G
Asynchronous_Notification_Enable_DM	SUBSCRIBE	G
Asynchronous_Notification_Query_DM	SUBSCRIBE	G
Directory_Manager_Local_Tag_Define_DM	DEFINESTAG	P,S
Directory_Manager_Local_Tag_Delete_DM	DEFINESTAG	P,S
Directory_Manager_Local_Tag_Query_DM	SETSTAG	A
Directory_Manager_Local_Tag_Set_DM	SETSTAG	A
Directory_Manager_Search_DM	SCAN	H,S
Directory_Manager_Task_Cancel_DM	WORKUNIT	D
Image_CPU_Define_DM	SETCPU	A
Image_CPU_Delete_DM	SETCPU	A
Image_CPU_Query_DM	SETCPU	A
Image_CPU_Set_Maximum_DM	SETMACH	A
Image_Create_DM	ADD	A,D,G

Table 48. IBM-Supplied Defaults for DirMaint Commands and Command Sets (continued)

API	Command(s)	CMDSETs
Image_Definition_Create_DM	COMMAND	A
	GET	A
	REPLACE	S
	ADD	A
	AUTHSCIF	G
	DSECUSER	G
	CONSOLE	G
	SECUSER	G
	CPU	G
	SETMACH	A
	DEDICATE	P,S
	INCLUDE	A
	IPL	G
	LINK	G
	DMDISK	D
	AMDISK	D
	RMDISK	D,P
	MDISK	G
	NICDEF	G
	SETOPTN	A
	SETCLASS	A
	MAXSTORAGE	A
	STORAGE	G
	SHARE	A
	SETPW	A,M
	LOCK	A
	UNLOCK	A
	VMRELOCATE	A
Image_Definition_Delete_DM	COMMAND	A
	GET	A
	AUTHSCIF	G
	DROPSCIF	G
	CONSOLE	G
	SECUSER	G
	SETCPU	A
	MACHINE	G
	DEDICATE	P,S
	INCLUDE	A
	IPL	G
	LINK	G
	DMDISK	D
	NICDEF	G
	SETOPTN	A
	PRIVCLASS	A
	MAXSTORAGE	A
	STORAGE	G
	SHARE	A
	SETPW	A,M
	LOCK	A
	REPLACE	S
	UNLOCK	A
	VMRELOCATE	A
Image_Definition_Query_DM	GET	A
	GLOBALOPTS	P

Table 48. IBM-Supplied Defaults for DirMaint Commands and Command Sets (continued)

API	Command(s)	CMDSETs
Image_Definition_Update_DM	COMMAND GET REPLACE ADD AUTHSCIF DSECUSER CONSOLE SECUSER CPU SETMACH DEDICATE INCLUDE IPL LINK DMDISK AMDISK RMDISK MDISK NICDEF SETOPTN SETCLASS MAXSTORAGE STORAGE SHARE SETPW LOCK UNLOCK VMRELOCATE	A A S A G G G G G A P,S A G G D D D,P G G A A A G A A,M A A A
Image_Delete_DM	PURGE	A,D
Image_Device_Dedicate_DM	DEDICATE	A
Image_Device_UnDedicate_DM	DEDICATE	A
Image_Disk_Copy_DM	CLONEDISK	D
Image_Disk_Create_DM	AMDISK	D
Image_Disk_Delete_DM	DMDISK	D
Image_Disk_Share_DM	LINK	G
Image_Disk_UnShare_DM	LINK	G
Image_IPL_Delete_DM	IPL	G
Image_IPL_Query_DM	IPL	G
Image_IPL_Set_DM	IPL	G
Image_Lock_DM	LOCK	A
Image_Name_Query_DM	USERMAP	A
Image_Password_Set_DM	PW	G
Image_Query_DM	GET	A
Image_Replace_DM	REPLACE	S
Image_SCSI_Characteristics_Query_DM	LOADDEV	G
Image_SCSI_Characteristics_Define_DM	LOADDEV	G
Image_Unlock_DM	UNLOCK	A
Image_Volume_Space_Define_DM	DASD	S
Image_Volume_Space_Define_Extended_DM	DASD	S
Image_Volume_Space_Query_DM	DASD	S
Image_Volume_Space_Query_Extended_DM	DASD	S
Image_Volume_Space_Remove_DM	DASD	S
Profile_Create_DM	ADD	A

Table 48. IBM-Supplied Defaults for DirMaint Commands and Command Sets (continued)

API	Command(s)	CMDSETs
Profile_Delete_DM	PURGE	A
Profile_Lock_DM	LOCK	A
Profile_Query_DM	GET	A
Profile_Replace_DM	REPLACE	S
Profile_Unlock_DM	UNLOCK	A
Prototype_Create_DM	CMS FILE	S S
Prototype_Delete_DM	CMS	S
Prototype_Name_Query_DM	CMS	S
Prototype_Query_DM	SEND	H,S
Prototype_Replace_DM	CMS FILE	S S
Query_All_DM	USER	S,H
Query_Asynchronous_Operation_DM	STATUS	S,H
Query_Directory_Manager_Level_DM	none	*
Shared_Memory_Access_Add_DM	NAMESAVE	A
Shared_Memory_Access_Query_DM	NAMESAVE	A
Shared_Memory_Access_Remove_DM	NAMESAVE	A
Shared_Memory_Create	NAMESAVE	A
Shared_Memory_Delete	NAMESAVE	A
Shared_Memory_Replace	NAMESAVE	A
Static_Image_Changes_Activate_DM	ONLINE	S
Static_Image_Changes_Deactivate_DM	OFFLINE	S
Static_Image_Changes_Immediate_DM	DIRECT	A,H
Virtual_Channel_Connection_Create_DM	SPECIAL	G
Virtual_Channel_Connection_Delete_DM	SPECIAL	G
Virtual_Network_Adapter_Create_DM	NICDEF	G
Virtual_Network_Adapter_Create_Extended_DM	NICDEF	G
Virtual_Network_Adapter_Delete_DM	NICDEF	G
Virtual_Network_Adapter_Connect_LAN_DM	NICDEF	G
Virtual_Network_Adapter_Connect_Vswitch_DM	NICDEF	G
Virtual_Network_Adapter_Disconnect_DM	NICDEF	G

These are only defaults, which may be customized. Remember to authorize command set D (DASD Management commands) for both command levels 140A and 150A. It is customary to authorize users for the same command sets in both command levels, but command set D is the critical one if the user will be making ADD (Image_Create_DM) requests.

DASD Management

In addition to the preceding command set authorizations, use of the DASD management space allocation requests (ADD or Image_Create_DM, AMDISK or Image_Disk_Create_DM, CLONEDISK or Image_Disk_Copy_DM, CMDISK, or RMDISK) require the requester to be authorized to allocate space of the specified type (group, region, or volume) in the area named. This is controlled by the AUTHDASD CONTROL file, located on DIRMAINT's 1DF disk. By default, all

command set D users are authorized to allocate DASD space anywhere on the system. Note, however, that some installations may customize the AUTHDASD CONTROL file to restrict this.

Once authorized, DASD space may be allocated on named volumes, in named regions (which are subsets of volumes), or in named groups (which are collections of named regions). The EXTENT CONTROL file contains this information. For more information, see Chapter 6, “DASD Management,” on page 73 and the following commands in the *z/VM: Directory Maintenance Facility Commands Reference*: AMDISK, CLONEDISK, CMDISK, DMDISK, RMDISK, FREEXT, USEDEXT, and DIRMAP.

The REGIONS and GROUPS sections of the EXTENT CONTROL file may be maintained using the Image_Volume_Space_Define_DM, Image_Volume_Space_Remove_DM and Image_Volume_Space_Query_DM functions.

Image Disk Modes

The Image_Disk_Create_DM and Image_Disk_Copy_DM functions accept a limited range of input for the *image_disk_mode* parameters. Valid input to these parameters is limited to the following:

R	RR	W	WR	M	MR	MW
RE	RRE	WE	WRE	ME	MRE	MWE
RS	RRS	WS	WRS	MS	MRS	MWS
RD	RRD	WD	WRD	MD	MRD	MWD
RED	RRED	WED	WRED	MED	MRED	MWED
RSD	RRSD	WSD	WRSD	MSD	MRSD	MWSD
RV	RRV	WV	WRV	MV	MRV	MWV
RVE	RRVE	WVE	WRVE	MVE	MRVE	MWVE
RVS	RRVS	WVS	WRVS	MVS	MRVS	MWVS
RVD	RRVD	WVD	WRVD	MVD	MRVD	MWVD
RVED	RRVED	WVED	WRVED	MVED	MRVED	MWVED
RVSD	RRVSD	WVSD	WRVSD	MVSD	MRVSD	MWVSD

Image_Create_DM: Adding A New Image

Once the requisites have been satisfied, adding a new virtual image is not particularly difficult. The trick is in satisfying all of the requisites:

1. The ADD or Image_Create_DM request must be made by a userid that is authorized to make requests in command sets A, D, and G in command level 150A for ALL users; AND (usually) for command set D in command level 140A for ALL users.
2. The Image_Create_DM request can either supply a buffer containing the virtual image directory entry or else supply the name of an existing PROTODIR file from which the new userid, virtual machine, or image is to be created. The PROTODIR file is created using a DIRM FILE command, or a Prototype_Create_DM API request.
3. If the PROTODIR file contains an INCLUDE statement for a PROFILE, the named PROFILE must already exist.

4. The name of the new image may be the same as the name of a PROTODIR file, but must not be the same as an existing PROFILE or USER name.
5. Unless a password is specified on the Image_Create_DM API function call, the password will be set to NOLOG, making the userid unusable until an Image_Password_Set_DM function has been performed for the userid.
6. Unless an account number is specified on the Image_Create_DM function call, the number specified on the ACCOUNT statement within the PROTODIR file will be used. If none is specified within the PROTODIR file, the ACCOUNT specified in the included PROFILE will be used. If there is no included PROFILE, or the included PROFILE does not specify an ACCOUNT statement, the virtual machine's userid will be used as the account number.
7. DirMaint requires account numbers specified on the DIRM ADD command to be verified before they are accepted. The default sample ACCOUNT_NUMBER_VERIFICATION_EXIT provided by DirMaint (DVHXAV EXEC) accepts all account numbers from the DIRM ADD command, as DIRM ADD is an administrator (privileged) command.

   In order for account numbers to be accepted by the Image_Create_DM API (which uses the DIRM ADD command), you must have the following configuration file record in a supplemental configuration file:

   ```
   ACCOUNT_NUMBER_VERIFICATION_EXIT= DVHXAV EXEC
   ```

   The first six characters of the supplemental configuration file name must be 'CONFIG,' the seventh and eighth characters may be any valid file identification characters ('AV' is suggested and will be used here as an example), and the file type must be DATADVH. This supplemental CONFIGAV DATADVH file should reside on the DIRMAINT server's 1DF disk or on any other non-public disk in its search order.

   You may tailor the sample exit to your needs if you require a more detailed account number verification function. If this exit is not configured and an account number is specified on the Image_Create_DM API, the API will fail with an account policy error (return code RCERR_POLICY_ACCT).

Prototype_Create_DM: Adding A New PROTODIR File

The PROTODIR file may contain almost any valid directory statement, as described in *z/VM: CP Planning and Administration*.

1. Any DASD space allocation statements contained in the PROTODIR file should specify unformatted DASD space.
   - The statement name should be MDISK (rather than AMDISK).
   - If the AMDISK keyword is used, neither the BLKSIZE nor the LABEL keywords should be used.
   - Use of the BLKSIZE or LABEL keywords cause an asynchronous DirMaint process to be spawned for which neither completion nor results can be verified.

   If CMS formatted DASD space is required, the PROTODIR file should omit the space allocation statement(s), and subsequent Image_Disk_Create_DM (AMDISK) or Image_Disk_Copy_DM (CLONEDISK) requests should be made to add and format the space once the Image_Create_DM (ADD) function has completed.
2. All DASD space allocation (MDISK) statements contained in the PROTODIR file must specify some form of automatic allocation, using an AUTOG, AUTOR, AUTOV, T-DISK, V-DISK, or the block size variants of GBLK*nnnn*, RBLK*nnnn*, VBLK*nnnn*, or VDBS*nnnn*. Use of an absolute starting cylinder number is not

supported. If a disk must be allocated at a specific starting cylinder number for some reason, omit the definition of this disk from the PROTODIR file and use an Image_Disk_Create_DM (AMDISK) or Image_Disk_Copy_DM (CLONEDISK) request once the Image_Create_DM (ADD) function has completed.

A sample CMS PROTODIR file:

```
USER CMS NOLOG
INCLUDE CMSDFLT
MDISK 191 XXXX AUTOG 2 CMSGROUP MR
```

A sample LINUX PROTODIR file:

```
USER LINUX NOLOG
INCLUDE LINDFLT
MDISK 191 3390 AUTOG 0050 LINGROUP MR
MDISK 150 3390 AUTOG 3088 LINGROUP MR
MDISK 151 3390 AUTOG 0200 LINGROUP MR
```

Replacing an Image Definition

The Image_Replace_DM function can be used to replace the definition of a virtual image in the source directory, however, its use must be coordinated with other functions to ensure the integrity of the definition. Before invoking Image_Replace_DM, perform the following steps.

1. Call the Image_Lock_DM function to lock the image against changes by anyone else.

   If the image is already locked, your Image_Lock_DM request will fail. Wait until whoever locked it has completed their update and unlocked it. You can use the DirMaint STATUS LOCKED request to find out who created the lock.
2. Call the Image_Query_DM function to obtain the current definition of the virtual image. This will include any changes made by anyone else since your most recent change.
3. Modify the results from your Image_Query_DM request to fit your current requirements for the image definition.

   Caution:

   You should not make changes to disk space allocations (MDISK directory statements). There is no provision in the Image_Replace_DM function for allocating new disk space using automatic allocation, formatting new space using explicit allocation, protection against allocating space already in use, or for security erasure of deleted space. Disk space allocation additions or deletions should be made using one or more of the following functions: Image_Disk_Create_DM, Image_Disk_Copy_DM, Image_Disk_Delete_DM.
4. When ready, call the Image_Replace_DM function to have the Directory Manager update the source directory with your new definition. When the replacement definition has been accepted as the new directory source, the image is implicitly unlocked.

   If someone else has replaced the definition between your Image_Lock_DM and Image_Replace_DM requests, or has unlocked the image in that interval, your Image_Replace_DM request will fail. To ensure integrity of the definition, re-lock and re-query the image, redo your modifications, and then repeat the Image_Replace_DM request.
5. If you change your mind and choose not to replace an image definition after locking it, you must explicitly unlock it with an Image_Unlock_DM function call before any other changes can be made to it.

Asynchronous DASD Management Operations

The results of certain DASD management operations may not always be immediately available to a caller of directory manager APIs. This is because certain DASD management operations complete asynchronously rather than synchronously.

These asynchronous DASD management APIs include Image_Disk_Create_DM, Image_Disk_Copy_DM, and Image_Disk_Delete_DM. When DirMaint initiates an asynchronous operation for one of these functions, the API will return an operation ID as an output parameter. Note that Image_Create_DM and Image_Delete_DM may have associated DASD management processes, which will also return an operation ID.

When a client application receives this operation ID, it should then pass the ID to the Query_Asynchronous_Operation_DM API to determine when the operation completes, and whether it completes successfully. A successful invocation of Query_Asynchronous_Operation_DM will return a value of RC_OK in the return code output parameter, and a value in the reason code output parameter which will indicate completion status (RS_ASYNC_OP_SUCCEEDED, RS_ASYNC_OP_FAILED, or RS_ASYNC_OP_IN_PROGRESS).

Note: If a nonexistent operation ID is specified, Query_Asynchronous_Operation_DM will return a value of RC_OK and a reason code value of RS_ASYNC_OP_SUCCEEDED.

If an asynchronous DASD management operation request does not complete in a timely manner, two courses of action are available to you. You can:

- Use native DirMaint commands to diagnose and repair the cause of the problem in an effort to allow the request to complete

or

- Cancel the asynchronous work unit using the Directory_Manager_Task_Cancel_DM API.

Other Asynchronous Operations

Various other DirMaint operations that are not related to DASD Management usually complete synchronously, but can sometimes complete in what appears to be an asynchronous manner. There are several reasons this can happen:

- Other DirMaint requests (of any kind) from other users may be queued ahead of the current request, so that the current request cannot start executing.
- DirMaint operation has been interrupted by human intervention at the DirMaint console.
- The operation is technically not a true asynchronous operation, but it did not complete within DirMaint's timeout limit. A common cause of this condition is that updating a directory through DIRECTXA can be time-consuming.

In these cases, the return code will be set to 592 (RCERR_ASYNC_DM) and the reason code will be set to a 1- to 4-digit DirMaint request identifier. The Query_Asynchronous_Operation_DM function should be used to check the status of these operations.

Asynchronous Notifications

When using DirMaint as the directory manager, DirMaint fills in the *user_word* field in the asynchronous subscription notification message (as described in the Usage

Notes for Asynchronous_Notification_Enable_DM in *z/VM: Systems Management Application Programming*) with either of the following:

- The directory statement name, if only one directory statement was updated (added, changed, or deleted)
- A DirMaint command operand (such as Replace) if multiple directory statements were updated.

Configure Request Log Processing

The DIRMAINT server keeps a log of the SMAPI requests and their associated return codes in the 1SAPI REQUESTS file on the DIRMAINT 155 disk. This log is used to respond to Query_Asynchronous_Operation_DM API calls. The number of previous days for which to keep request information can be configured using the `1SAPI_REQUESTS_BEHAVIOR=` configuration statement in a DirMaint override configuration file. The size of the DIRMAINT 155 disk should be evaluated for the appropriate space needed to store the typical number of requests that are received within the configured time interval. Sometimes an unexpected number of requests can be received in this time period, which could cause the DIRMAINT 155 disk to fill up and then lead to the DIRMAINT machine shutting down. In order to prevent this, the DIRMAINT server will automatically prune the 1SAPI REQUESTS file if the amount of space left on the 155 disk becomes less than or equal to the amount of space needed to prune the file plus 5 percent of the disk space. The percentage of total requests to prune can also be configured on the `1SAPI_REQUESTS_BEHAVIOR=` statement. For information on how to configure this statement, refer to the “Step 2. Select Restart and Recovery Characteristics” section in “CONFIG DATADVH” on page 28.

Refer to the *DirMaint Program Directory* for information on how to determine the size of the DIRMAINT 155 disk when DirMaint is used with the z/VM Systems Management APIs.

DirMaint Optimization: Static_Image_Changes_Activate_DM, _Deactivate_DM, and _Immediate_DM

Most changes to the source directory require DirMaint to use the DIRECTXA command to “compile” the source directory into the “object” directory. Depending on the size of your source directory, CPU speed, user load, and a variety of other factors, the DIRECTXA command may finish quickly (one second or less), may take several seconds, or may take longer.

- If DIRECTXA takes only a short time to run on your system, IBM recommends use of ONLINE= IMMED in the CONFIG* DATADVH file(s). This is fine for occasional isolated directory changes.
- If and when you, as sole system administrator, are going to make multiple updates to the directory in a short period of time, it may be advantageous to start with a Static_Image_Changes_Deactivate_DM call. This ensures that DirMaint does not make use of DIRECTXA until you are done with your changes. When your batch of changes is complete, make sure that you re-enable use of DIRECTXA with a Static_Image_Changes_Activate_DM call, followed by a Static_Image_Changes_Immediate_DM call.
- If there are multiple system administrators working concurrently, you will want to coordinate your use of Static_Image_Changes_Deactivate_DM and Static_Image_Changes_Activate_DM. The first administrator to begin should make the Deactivate call, and the last one finished should make the Activate call. As each administrator finishes his or her group of requests, a

Static_Image_Changes_Immediate_DM call can be made to put all of the changes accumulated by all of the administrators to that point online with one invocation of DIRECTXA.

- If DIRECTXA takes significantly longer to run on your system, you may want to run with ONLINE= SCHED in the CONFIG* DATADVH file(s), and generally use a DIRECT CONDITIONAL request at scheduled intervals (perhaps every 15 minutes, hourly, or every 4 hours) in the DIRMAINT DATADVH wakeup times control file. Under these circumstances, it is not usually necessary or desirable to make Static_Image_Changes_Deactivate_DM or Static_Image_Changes_Activate_DM calls. If you make a change that needs to be put into effect before the next scheduled update, you may use a Static_Image_Changes_Immediate_DM call to do so at any time.
- If a directory manager function ends with (rc,rs) = (0,8), this indicates that the source directory has been updated but the changes have *not* been put online in the object directory. If you wish to override DirMaint's configuration settings and bring this change online immediately, you may do so with a Static_Images_Changes_Immediate_DM call.

Appendix C. 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 a CONFIG* DATADVH file on the user interface disk, 11F and 21F. Move the service machine related entries into a CONFIG* 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 &DIRM commands daily) to put an `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: CP 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 a CONFIG* DATADVH file on the user interface disk (11F, 21F). Move the service machine related entries into a CONFIG* 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.
- 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: CP 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.
  - 2DF disk if the 1DF disk is backed up nightly.
  - 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).

    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 29.

```
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= SSI FM= F ACC= 551
          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 29. Copying all files from the 155 disk to the new V-disk

These steps 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 28.

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

For more information on these commands, see the *z/VM: CP Commands and Utilities Reference*.

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.

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.

Appendix D. DirMaint Configuration Data Files

This appendix provides a summary of each of the CONFIG* DATADVH entries provided by DirMaint. These data files provide information about the configuration of the DirMaint feature. 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 49 describes many of the CONFIG* DATADVH entries supported by DirMaint.

Table 49. CONFIG DATADVH Entries Summarized*

Entry Name	Possible Operands	Comments	Page
ADD_COMMAND_PROCESSING=	FULL or SHORT	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.	36
ALLOW_ASUSER_NOPASS_FROM=	serverid *|servernode	This statement gives the *userid* and *nodeid* of trusted service machines who can make requests including the ASUSER prefix keyword (which generally forces authentication) without supplying a password and thus without authentication. You must be EXTREMELY careful which servers are granted this capability, since this gives the listed servers the keys to your system (ASUSER DIRMAINT or ASUSER MAINT for examples). A *nodeid* of * may be used to represent any system within the cluster where the DIRMAINT server is running.	37
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.	31
CLASS LIMIT ON USER STATEMENT=	8 | 0 ... 32 | 0 ... 8	Specifies how many CP privilege classes may be included on the USER statement.	31
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.	32
COMMANDS_xxxx=	Handler routine file name	Defines the file name of the handler routine. This will determine what machine will process the command.	108
CYL0_BLK0_CLEANUP=	NO or YES	This entry supports your OBJECT REUSE policy.	34
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.	86
DASD_OWNERSHIP_NOTIFICATION_EXIT=	DVHXDN EXEC	Identifies the exit to be called to issue the necessary RACF commands for processing the DASD-related DirMaint commands (such as DIRMAINT AMDISK, DMDISK, etc.).	42

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
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.	73
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.	113
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.	109
DEFAULT_DELTA_OPTIONS=	Null, TRACE, CALL, RETURN, or USER *id*.	This value sets diagnostic options for delta processing.	N/A
DEFAULT_DIRECT_ACTION=	UNCONDITIONAL or CONDITIONAL	This value determines the default for the optional UNCONDITIONAL or CONDITIONAL parameters on the DIRMAINT DIRECT command. If specified as CONDITIONAL, the DIRECTXA command will not be issued unless there are pending changes to be processed. If specified as UNCONDITIONAL, the DIRECTXA command will be issued, regardless of any pending changes to be processed. (If omitted, the default is UNCONDITIONAL.)	N/A
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 z/VM 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.	31
DISK_CLEANUP=	NO or YES	This entry supports your OBJECT REUSE policy. Specify YES to have DirMaint format disk space from one user before allocating to another to protect privacy.	34

Table 49. CONFIG DATADVH Entries Summarized (continued)*

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.	33
DM_MAXIMUM_RETRIES=	A numeric value in the range of 0 through 9999	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 because the directory change to transfer the minidisk to DATAMOVE has not been put online), then 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. 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 for this value is 10. If a value is specified outside of the valid range of 0 through 9999, then the value will be set to 1.	63
DVHDXD_FLASHCOPY_BEHAVIOR=	0, 1, or 2	A value of 0 causes the exit to end with RC=30, which causes DATAMOVE to revert to use of DDR. A value of 1 causes the exit to issue the FLASHCOPY command. If FlashCopy Version 2 DASD is in use, the copy will be synchronous. If FlashCopy Version 1 DASD is in use, the copy will be asynchronous. A value of 2 causes the exit to issue the FLASHCOPY command with the SYNChronous option if FlashCopy Version 1 DASD is in use. If FlashCopy Version 2 DASD is in use, the copy will be synchronous. If the SYNChronous option is not supported and FlashCopy Version 1 DASD is in use, the CLONEDISK operation will complete when the COMMAND COMPLETE delayed response is received, unless polling FLASHCOPY commands are configured using the DVHDXD_FLASHCOPY_COMPLETION_WAIT configuration statement. Note that configuring DirMaint to poll the background copy associated with a CLONEdisk operation negates the performance benefits of FLASHCOPY. Performance in the polling method will be similar or worse to that of DDR.	N/A

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
DVHDXD_FLASHCOPY_COMPLETION_WAIT=	Two numeric values, each ranging from 0 (no wait) to 3600 (one retry every hour)	These values specify when, in number of seconds, to issue a subsequent FLASHCOPY command to check for the completion of the background copy process of a prior command. The first value is the wait between issuances of a CP FLASHCOPY 0 0 request, and the second is between issuances of a CP FLASHCOPY END END request. **Note:** DVHDXD_FLASHCOPY_COMPLETION_WAIT configures DirMaint to poll the background copy associated with a FLASHCOPY command. This polling method is used only when Enterprise Storage System FlashCopy Version 1 DASD is in use and the SYNChronous option is not supported on the CP FLASHCOPY command. Without this polling, the CLONEDISK operation will be considered complete when the COMMAND COMPLETE response is received from the FLASHCOPY command associated with the original CLONEDISK. At that time, the disks are considered useable by CP, but the disks may not be used in another FLASHCOPY operation until the background copy is complete. Configuring DirMaint to wait for the background copy to complete negates the performance benefits of FLASHCOPY. Performance in the polling method will be similar to or worse than that of DDR.	N/A
DVHDXD_FLASHCOPY_DIAGNOSTICS=	0, 1, 2, 3, 4, or 5	This value specifies the level of detail desired in DVHDXD diagnostics, as follows: **0** No diagnostics **1** Show entry time, date, and parameters plus exit RC **2** Show as per above, plus diagnostic options **3** Show as per above, plus device characteristics **4** Show as per above, plus intermediate RCs **5** Show as per above, plus details of each CP substep – if DVHDXD_FLASHCOPY_BEHAVIOR=2.	N/A
DVHDXD_FLASHCOPY_DIAGFILE_ERASE=	0 or 1	This value specifies whether (1) or not (0) the DVHDXD DIAGFILE is erased before beginning the next FLASHCOPY operation.	N/A

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
DVHDXD_FLASHCOPY_NOREPLY_WAIT=	Two numeric values, the first between 1 and 85400, the second between 1 and 61	The first value specifies the number of seconds the IBM-supplied sample exit will wait after issuing a FLASHCOPY command before presuming that the response is lost. The maximum allowable value is 85400 seconds (24 hours). If omitted, the default is 3660 seconds (61 minutes). The second value specifies the number of seconds the IBM-supplied sample exit will wait after receiving one reply before presuming that there are no subsequent replies. The maximum value is 61 seconds. The default is 3 seconds. **Note:** This configuration value is ignored if either DVHDXD_FLASHCOPY_BEHAVIOR= 1 or FlashCopy Version 2 DASD is in use.	N/A
DVHSAPI_END_MSG.*message*=	Any valid message numbers	Identifies user tailorable choices for when the DVHSAPI routine exits and returns control back to the calling application. The default, if no entries are specified, is to end when message DVHREQ2289I is received.	108
DVHSAPI_ENTER_KEY_ACTION=	END | IGNORE	Specifies whether pressing the ENTER key either terminates DVHSAPI or is ignored. The default (for compatibility) is END.	108
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
DVHXDN_RDEFINE_VMMDISK_DEFAULTS=	Any valid option on the RACF RDEFINE command	Specifies the defaults that will be used by DVHXDN when it issues a RACF RDEFINE command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied defaults are UACC(NONE) AUDIT(ALL(READ)).	152
DVHXUN_ADDUSER_DEFAULTS=	Any valid option on the RACF ADDUSER command	Specifies the defaults that will be used by DVHXUN when it issues a RACF ADDUSER command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE).	175
ESM_LOG_FILTER_EXIT=	DVHXLF EXEC	This entry supports your 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.	35

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
ESM_LOG_RECORDING_EXIT=	DVHESMLR EXEC	This entry supports your AUDITING policy. Use of an ESM with application audit logging capability allows AUDIT information to be located in a single repository.	35
ESM_PASSWORD_AUTHENTICATION_EXIT=	DVHXPA EXEC	This entry supports your AUTHENTICATION policy. A non-blank value is required to use an ESM for authentication.	34, 43
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.	108
LOADABLE_USER_FILE=	User file names	Defines the user file to be made resident or nonresident by using the EXECLOAD and EXECDROP commands.	108
LOGONBY_CHANGE_NOTIFICATION_EXIT=	DVHXLB EXEC	Identifies the exit to be called to issue the necessary RACF commands for DIRMAINT LOGONBY command processing.	42
MAXIMUM_DATAMOVE_AUTOLOGS=	Numeric value in the range of 0 through 99	Before assigning a workunit to a DATAMOVE machine, DirMaint will determine if the DATAMOVE machine is logged on. If the DATAMOVE machine is *not* logged on, then DirMaint will attempt to autolog it. The MAXIMUM_DATAMOVE_AUTOLOGS value specifies the number of times DirMaint will attempt to autolog a DATAMOVE machine which is not logged on to the system before quiescing the machine for manual intervention. Note that the autolog count is reset to zero during DAILY processing, so that if an autolog is successful and no further failures occur before DAILY processing, then MAXIMUM_DATAMOVE_AUTOLOGS will be attempted again. The default for this value is 10. If a value is specified outside of the valid range of 0 through 99, then the default will be used.	86
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.	86

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
MAXIMUM_WORKUNIT_RETRIES=	Numeric value in the range of 0 through 999	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 issued, or because the directory change to transfer the minidisk to DATAMOVE hasn't been put online yet), then the FORMAT/COPY/CLEAN request will be put onto the DATAMOVE machine's retry queue. The MAXIMUM_WORKUNIT_RETRIES value specifies the number of times DirMaint will retry a workunit after the first attempt to process the work unit. Once the number of retries are attempted without success, the workunit will be cancelled and rolled back. There is no default setting. If a value outside of the valid range is specified or if MAXIMUM_WORKUNIT_RETRIES is not configured, then work units will be retried forever.	86
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.	39
MESSAGE_LOG_RETENTION_PERIOD=	*months*	This entry supports your 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.	35
MESSAGE_LOGGING_FILETYPE= MESSAGE_LOGGING_FILTER_EXIT=	TRANSLOG DVHXLF EXEC	These entries support your AUDITING policy. You may control what messages are LOGGED and where they are LOGGED.	35
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.	30
PARSER_xxxx=	Parser file name	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.	108
PASSWORD_CHANGE_NOTIFICATION_EXIT=	DVHXPN EXEC	Identifies the exit to be called to issue the necessary RACF commands for DIRMaint PW and SETPW command processing.	42

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
POSIX_CHANGE_NOTIFICATION_EXIT=	DVHXPESM EXEC	Identifies the exit to be called to issue the necessary RACF commands for DIRMAINT POSIXGLIST, POSIXGROUP, POSIXINFO, POSIXFSROOT, POSIXIUPGM, POSIXIWDIR, POSIXUID, and POSIXOPT command processing.	42
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*.	35
PURGE_COMMAND_PROCESSING=	FULL or SHORT	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 your OBJECT REUSE policy. Use of the `PURGE_COMMAND_PROCESSING= SHORT` will bypass the disk cleanup.	36
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.	38
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.	38

Table 49. CONFIG DATADVH Entries Summarized (continued)*

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.	38
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.	38
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 *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*.	N/A

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
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.	39
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.	39
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.	109
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. **Note:** If the IBM supplied default of 365 DAYS is changed, you need to enable a PASSWORD CHANGE NOTIFICATION EXIT = DVHXPN EXEC statement in the CONFIG* DATADVH file.	109
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).	38
RACF_ADDUSER_DEFAULTS=	Any valid option on the RACF ADDUSER command	Specifies the defaults that will be used by DVHXUN when it issues a RACF ADDUSER command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE).	42
RACF_DISK_OWNER_ACCESS=	Any valid access option on the RACF PERMIT CLASS(VMMDISK) command.	Specifies the access authority that will be used by DVHXDN when it issues the RACF PERMIT command for the owner of a disk that is being added. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is ACC(ALTER).	42
RACF_RDEFINE_SURROGAT_DEFAULTS=	Any valid option on the RACF RDEFINE SURROGAT command	Specifies the defaults that will be used by DVHXUN or DVHXLB when it issues a RACF RDEFINE SURROGAT command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE) AUDIT(FAILURES(READ)).	43
RACF_RDEFINE_VMBATCH_DEFAULTS=	Any valid option on the RACF RDEFINE VMBATCH command	Specifies the defaults that will be used by DVHXUN when it issues a RACF RDEFINE VMBATCH command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE) AUDIT(FAILURES(READ)).	43

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
RACF_RDEFINE_VMMDISK_DEFAULTS=	Any valid option on the RACF RDEFINE VMMDISK command	Specifies the defaults that will be used by DVHXDN when it issues a RACF RDEFINE VMMDISK command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied defaults are UACC(NONE) AUDIT(FAILURES(READ)).	42
RACF_RDEFINE_VMPOSIX_POSIXOPT. QUERYDB=	Any valid option on the RACF RDEFINE VMPOSIX POSIXOPT.QUERYDB command	Specifies the defaults that will be used by DVHXUN or DVHXPESM when it issues a RACF RDEFINE VMPOSIX POSIXOPT.QUERYDB command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(READ).	42
RACF_RDEFINE_VMPOSIX_POSIXOPT.SETIDS=	Any valid option on the RACF RDEFINE VMPOSIX POSIXOPT.SETIDS command	Specifies the defaults that will be used by DVHXUN or DVHXPESM when it issues a RACF RDEFINE VMPOSIX POSIXOPT.SETIDS command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE).	43
RACF_RDEFINE_VMRDR_DEFAULTS=	Any valid option on the RACF RDEFINE VMRDR command	Specifies the defaults that will be used by DVHXUN when it issues a RACF RDEFINE VMBATCH command. (See the *z/VM: RACF Security Server Command Language Reference* for valid options.) The IBM-supplied default is UACC(NONE) AUDIT(FAILURES(READ)).	43
RACF_VMBATCH_DEFAULT_MACHINES=	The names of any batch machines on the system	Identifies the batch machines available on the system.	43
REQUIRED_USER_FILE=	User file names	Defines the files needed in the user's virtual machine to enter any DIRMAINT commands.	108
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.	29
SAMPL_LINESIZE_xxxx=	40-222	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.	108
SAMPL_USER_MSGS_xxxx=		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.	108

Table 49. CONFIG DATADVH Entries Summarized (continued)*

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.	67
SERVICE_LEVEL_INFO=	ALL, *fn* EXEC, *fn* XEDIT, *fn* REXX, *fn* MODULE, or CONFIG	Used to specify a file set for DVHSERVL $EXEC, which is used to display a service level information report. *fn* is a file name of the file to be included in the report. You can specify * for the file name to include all files of the type you specify. You can insert multiple SERVICE_LEVEL_INFO= statements. File identifiers that are complied from multiple SERVICE_LEVEL_INFO= specifications will be combined into a single list.	189
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.	32
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.)	35
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`	32

Table 49. CONFIG DATADVH Entries Summarized (continued)*

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.	32
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.	29
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.	29
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	29

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
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.	37
SPOOL_FILE_SECLABEL=	SYSLOW	This entry supports your ACCESS CONTROL policy.	34
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.	29
TREAT_RAC_RC.4=	0, 4, or 30	This tells DVHXUN, DVHXDN, DVHXPESM, and DVHXLB to treat return code 4 (authorization decision deferred by RACF to z/VM) from the RACF commands as if the return code was either 0 (successful) or 30 (RACF not installed). The default is 4, which denotes no change in interpretation.	152, 175, 43
UPDATE_IN_PLACE=	YES or NO	This value controls whether DIRMAINT will attempt to use DIAGNOSE code X'84' to put directory changes online. It can be changed using the DIRM OFFLINE and DIRM ONLINE commands.	30
USE_RACF=	YES|NO ALL|*exit_name*	Specifies whether automatic communication with RACF is enabled or disabled for user exits. YES ALL indicates that all automatic communication with the RACF server is enabled (except for exits overridden with a USE_RACF= NO statement). NO ALL indicates that all automatic communication with the RACF server is disabled (except for exits overridden with a USE_RACF= YES statement). *exit_name* indicates the file name and file type of a user exit which should be enabled or disabled for automatic RACF communication.	41
USER_CHANGE_NOTIFICATION_EXIT=	DVHXUN EXEC	Identifies the exit to be called to issue the necessary RACF commands for processing the user profile-related DirMaint commands (such as DIRMAINT ADD, PURGE, etc.).	42
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.	32

Table 49. CONFIG DATADVH Entries Summarized (continued)*

Entry Name	Possible Operands	Comments	Page
WRK_UNIT_ONLINE=	NO or YES	This value controls whether DIRMAINT will include 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.	30

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
```

SSI or CSE Cluster Configuration Entries

The following identifies how to route files to other systems within a multiple-system SSI or CSE cluster.

Format:

```
FROM= fromspec DEST= destspec S= spoolid T= tagspec1 U= tagspec2
```

Where:

fromspec
: Identifies the network *nodeid* or service machine *userid* where the transaction originates.

destspec
: Identifies the network *nodeid* or service machine *userid* 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 (DIRMAINT, DATAMOVE, or DIRMSAT) at that node. Otherwise, it is the *userid* of an RSCS network machine or spool file bridge.

tagspec1
Identifies the network *nodeid* or service machine *userid* of the spool file tag.

tagspec2
Identifies the *userid* of the spool file tag.

Notes:

1. Cluster support is enabled by default.
2. *fromspec*, *destspec*, and *tagspec1* are node IDs or user IDs, as determined by the IDENTITY command. *destspec* and *tagspec1* are usually the same.
3. FROM=* means from anywhere.
4. DEST=* means wherever the DIRMAINT server is running, as recorded in the WHERETO DATADVH file.
5. S=* means the network server, as determined by the IDENTITY command.
6. T=* means the same as the *destspec* value.

If the defaults are insufficient with shared spool files in an SSI cluster, the *spoolid* value can be specified as the user ID of the appropriate DIRMSAT machine, as DIRMSAT can work as a spool file bridge to users on remote systems. For sample routing statements using the DIRMSAT machine as a spool file bridge, see “Step 2. Identify the Communication Path” on page 68.

For performance reasons, you may choose to establish a dedicated network for the CSE or SSI cluster. In this case, the *spoolid* value must identify the special server for the cluster, and *tagspec1* must provide the correct tag data for that particular server. Here is an example of using a spool file bridge:

```
FROM= DVHTEST1 DEST= DVHTEST2 S= SFBRIDGE T= DVHTEST2
FROM= DVHTEST1 DEST= DVHTEST3 S= SFBRIDGE T= DVHTEST3
FROM= DVHTEST1 DEST= DVHTEST4 S= SFBRIDGE T= DVHTEST4
FROM= DVHTEST2 DEST= DVHTEST1 S= SFBRIDGE T= DVHTEST1
FROM= DVHTEST2 DEST= DVHTEST3 S= DIRMAINT T= DVHTEST3
FROM= DVHTEST2 DEST= DVHTEST4 S= DIRMAINT T= DVHTEST4
FROM= DVHTEST3 DEST= DVHTEST1 S= SFBRIDGE T= DVHTEST1
FROM= DVHTEST3 DEST= DVHTEST2 S= DIRMAINT T= DVHTEST2
FROM= DVHTEST3 DEST= DVHTEST4 S= DIRMAINT T= DVHTEST4
FROM= DVHTEST4 DEST= DVHTEST1 S= SFBRIDGE T= DVHTEST1
FROM= DVHTEST4 DEST= DVHTEST2 S= DIRMAINT T= DVHTEST2
FROM= DVHTEST4 DEST= DVHTEST3 S= DIRMAINT T= DVHTEST3
```

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 multiple-system CSE cluster. For example:

```
SATELLITE_SERVER= DIRMSAT1 DVHTEST2
SATELLITE_SERVER= DIRMSAT2 DVHTEST2
SATELLITE_SERVER= DIRMSAT3 DVHTEST3
SATELLITE_SERVER= DIRMSAT4 DVHTEST3
SATELLITE_SERVER= DIRMSAT5 DVHTEST4
SATELLITE_SERVER= DIRMSAT6 DVHTEST4
SATELLITE_SERVER= DIRMSAT7 DVHTEST1
SATELLITE_SERVER= DIRMSAT8 DVHTEST1
```

Network Configuration Entries

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
```

System Performance

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*	29
LOADABLE_DATAMOVE_FILE= *filename filetype*	65
LOADABLE_DIRMSAT_FILE= *filename filetype*	70
LOADABLE_DIRMAINT_FILE= *filename filetype*	43

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
```

System Exit Routines Entries

The following identifies the service machine exit routines. For more information, see Chapter 9, “Exit Routines,” on page 125.

```
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_COPY_CMS_EXIT=             DVHDXC EXEC
DATAMOVE_DDR_EXIT=                  DVHDXD EXEC
DATAMOVE_ERASE_EXIT=                DVHDXE EXEC
DATAMOVE_FORMAT_EXIT=               DVHDXF EXEC
DATAMOVE_COPY_NONCMS_EXIT=          DVHDXN 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.

User Performance Entries

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
```

User Exit Entries

The following identifies the user exit routines. For more information, see Chapter 9, “Exit Routines,” on page 125.

```
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 E. 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 50. 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 *z/VM: CMS Commands and Utilities Reference*.

Appendix F. Making Multiple Updates to a Directory

Once a directory has been initialized, it should not be edited directly. Direct editing invariably introduces checksum errors, possibly for every entry if default serialization is allowed to take place. An editor may only be used as part of the cycle of transactions (DIRM FOR *userid* GET, RECEIVE, XEDIT *userid* DIRECT A, DIRM FOR *userid* REPLACE), when applied to a single directory entry at a time.

To make multiple updates to the entire directory (updates that may affect multiple users) the following procedure is recommended:

1. Disable updates to the source directory (DIRM DISABLE)
2. Ensure you have the latest copy of the directory (DIRM USER BACKUP)
3. Receive the monolithic backup (DIRM SEND USER BACKUP G)

   Note: You can optionally combine these three commands into a batch job file (called BULKUPDT PART1 A for this example), containing the following commands:

   ```
   DISABLE
   USER BACKUP
   SEND USER BACKUP G
   ```

   and then submit this batch job file the DIRMAINT server using a DIRM BATCH BLKUPDT PART1 command.
4. Receive the file to your A-minidisk and use your editor to make the changes.
5. Replace the user input file (DIRM FILE USER BACKUP A USER INPUT E) on the DIRMAINT server.
6. Create a batch job file (called BULKUPDT PART2 A for this example), containing the following commands:

   ```
   CMS ERASE USER DIRECT E
   RLDDATA
   ENABLE
   ```

   and submit this batch job file to the DIRMAINT server using a DIRM BATCH BULKUPDT PART2 command.

   Note: This is required. Otherwise, authentication of the DIRM RLDDATA command will fail once the USER DIRECT E file has been erased. (If you've inadvertently done this, you can correct it by logging on to the DIRMAINT server and issuing the RLDDATA and ENABLE commands from the console, then CP DISC.) If you wish, the DIRM ENABLE command can be issued separately.

Appendix G. Test the Installation/Service for DirMaint

This appendix is used for testing the initial installation of DirMaint and its related server machines. It will also be used prior to placing new service into production. You should follow the steps to test each of the server machines that you are using.

Summary of the Installation Process

- This procedure will require the DirMaint server machines to be shutdown, therefore, you should only test when it will least disrupt your production environment.
- If the install ID is logged on you will get messages stating that the minidisks can not be accessed R/W. The install ID should be logged off during this procedure.
- **If testing after installing DirMaint**, then the RUNMODE= entry of the CONFIG DATADVH file (on the 6VMDIR20 41F minidisk) is set to *testing*. You should **never** change the CONFIG DATADVH file. To change the RUNMODE= entry, do so using override files as described in the "CONFIG DATADVH" on page 28.
- **If testing after applying service**, then be aware of what the RUNMODE= entry is set to. You should review whether or not you want the DIRMAINT server to have the ability to place the source directory online while testing and set the RUNMODE= entry accordingly in your override file.
- During these instructions, privileged commands are issued from another user ID. The AUTHFOR CONTROL file should have an entry for the user ID you will be issuing these commands from. For the purposes of these instructions, the test user ID will be called **DIRMUSER**. If you are testing from a different user ID, substitute that name for DIRMUSER.

  Note: To test multiple-system CSE or SSI clusters, you will need a DIRMUSER user ID on a second and third system in the cluster. For testing an SSI cluster, the DIRMUSER user ID must be a multiconfiguration virtual machine defined using IDENTITY and SUBCONFIG entries.
- For the purposes of these instructions, **DIRMAINT**, **DATAMOVE**, and **DIRMSAT** are assumed as the name of the server user ID's. If you are using a different user ID name, substitute that name for the particular server name where applicable in this appendix.
- The test instructions in this appendix are divided into four sections as follows:
  - "Test the DIRMAINT Server Machine" for DIRMAINT,
  - "Test the DIRMSAT Server Machine" on page 249 for DIRMSAT,
  - "Test the DATAMOVE Server Machine" on page 254 for DATAMOVE, and
  - "Quick Test After Installing Service" on page 261 for testing after applying of service.

Following a new installation of DirMaint, complete all relevant test sections then proceed with "Post Test Instructions" on page 260. After installing RSU or corrective service, you may alternatively use the procedure in "Quick Test After Installing Service" on page 261.

Test the DIRMAINT Server Machine

This procedure will test the new DIRMAINT code to see that it functions properly. You will log on a test server machine and access the appropriate disks. The DIRMUSER user ID is required in order to verify the DIRMAINT function.

Notes

1. Starting with step 6, this procedure will have you make updates to the source directory and verify these changes took effect. To do this the RUNMODE= entry within an override file to the CONFIG DATADVH file (on the 6VMDIR20 41F minidisk) must be set to *operational*.

__ 1. Log on to the **DIRMAINT** server.

__ 2. If the DIRMAINT server is currently running, it must be shutdown.

shutdown

__ 3. Link to the 6VMDIR20's test disks

profile test

You will not see a CMS Ready message. Instead, this condition has been replaced by a line with the DIRMAINT service machine's network node id, user id, the date, return code, and time of day.

__ 4. Verify the appropriate disks have been accessed

The disk mode and addresses or directory names should match the configuration specified in the DVHPROFA DIRMAINT file. If not, make the necessary corrections and run the profile exec again.

q accessed

```
Mode  Stat     Files  Vdev  Label/Directory
A      R/W         1  155   DIR155
C      R/W       294  191   DRM492
D      R/W        57  11F   DRM41F
E      R/W        45  1DF   DIR1DF
F      R/O        25  551   DIR551
G      R/W         7  1DB   DIR1DB
H      R/W        10  1AA   DIR1AA
S      R/O       464  190   CMSESA
Y/S    R/O         3  19E   Y-DISK
Z      R/W         0  1FA   DIR1FA
DIRMAINT DVHTEST1 - 2011/04/11; T=0.01/0.01 17:28:35
```

__ 5. Check for any extraneous files

This check verifies that the disks are accessed, that the message handler is functioning correctly, and that there are no back level executable files that have been misplaced in the search order. If you have a large number of duplicate files, or if you have any executable files, use the EXEC DVHUCHK CHECK FILELIST command to resolve the discrepancies before continuing.

exec dvhuchk check itemize

Depending on your environment, the messages issued may contain different information.

```
DVHUCH1371W Filemodes C and E both contain AUTHDASD DATADVH.
DVHUCH1371W Filemodes A and C both contain DIRMAINT DATADVH.
DVHUCH1371W Filemodes A and E both contain DIRMAINT DATADVH.
DVHUCH1371W Filemodes A and G both contain DIRMAINT DATADVH.
DVHUCH1372I There were 4 extraneous files found.
DVHUCH1190I Command CHECK complete; RC= 2.
DIRMAINT DVHTEST1 - 2011/04/11(00002); T=0.40/0.43 18:11:32
```

__ 6. Update the RUNMODE=, ONLINE=, and DATAMOVE_MACHINE= entries.

cms xedit config*nn* datadvh d

Add a RUNMODE= OPERATIONAL entry and an ONLINE= IMMED entry to this override configuration file, where *nn* corresponds to the characters you choose for the override file.

If you will be testing multiple DATAMOVE servers in a CSE cluster, you will also need to add DATAMOVE_MACHINE= entries for each DATAMOVE server you will be testing. Refer to “Step 1. Define a DATAMOVE Service Machine to DIRMAINT” on page 61 for more information on configuring the DATAMOVE_MACHINE= statements. In an SSI environment, the DATAMOVE servers are already defined for you.

Save this file on the D disk (41F).

====> file

__ 7. Start the server machine using the new code and place the new CONFIG* DATADVH override file into use.

The DVHBEGIN EXEC will start the server. This will make a copy of the USER INPUT file (on the primary directory disk, 1DF) as USER BACKUP (on the directory backup disk, 1DB), and then convert the USER BACKUP file from *monolithic* to *clusterized* format. This also initializes all of the other control files DIRMAINT needs to begin updating the source directory. The operation of a substantial portion of the DirMaint product code will have been verified by this step.

Note: It is recommended that a copy of the USER BACKUP file be made prior to issuing the DVHBEGIN command.

dvhbegin

The DIRMAINT service machine should now be waiting for work.

```
DVHILZ3510I Starting DVHINITL with directory: USER INPUT E
DVHILZ3510I DVHINITL Parms: BLDMONO
DVHILZ3509I Monolithic backup now exists as: USER BACKUP G.
DVHILZ3509I Continuing with execution.
DVHILZ3510I Starting DVHINITL with directory: USER BACKUP G
DVHILZ3510I DVHINITL Parms: BLDCLUSTER BLDLINK BLDDASD
DIRMAINT DVHTEST1 - 2011/04/11; T=4.30/4.69 18:58:06
DVHWAI2140I Waiting for work on 2011/04/11 at 18:58:06.
```

__ 8. Verify the operation of DirMaint's console interrupt handler, the command syntax parser, and part of the authorization checking.

cp query files

```
DVHWAI2146I Wakeup caused by console attention on 2011/04/11 at 19:20:00.
DVHREQ2290I Request is: CP query files
DVHREQ2288I Your CP request for DIRMAINT at * has been accepted.
FILES:   NO RDR, 0001 PRT,   NO PUN
DVHREQ2289I Your CP request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/11; T=0.56/0.61 19:20:05
DVHWAI2140I Waiting for work on 2011/04/11 at 19:20:05.
```

__ 9. Test DirMaint's automatic restart and recovery.

__ a. Create a CP disabled wait state.

If a restart is not successful, check the `SHUTDOWN_REIPL_COMMAND= CP IPL CMS PARM AUTOCR` entry in the CONFIG* DATADVH file(s). You may need to correct the name of the CMS saved system, or change it to IPL by device address rather than by system name. It should

match the IPL statement you included in the DIRMAINT machine's directory entry, or the IPL statement included in the profile used by the DIRMAINT machine's directory entry.

cp system clear

```
DVHWAI2146I Wakeup caused by console attention on 2011/05/16 at 21:51:50.
DVHREQ2290I Request is: CP system clear
DVHREQ2288I Your CP request for DIRMAINT at * has been accepted.
Storage cleared - system reset.
DMSWSP314W Automatic re-IPL by CP due to disabled wait; PSW 000A0000 00000000
z/VM V6.2.0    2011-05-01 10:40
PRT FILE 1019 SENT FROM DIRMAINT CON WAS 1019 RECS 0049 CPY  001 0 HOLD   NOKEEP
   ⋮
PRODUCT:
IBM Directory Maintenance Facility for z/VM (DirMaint)
5741-A05 (C) Copyright IBM Corporation 1979, 2011.
Function Level 620 Service Level nnnn.
DMSACC724I 155 replaces A (191)

DVHPRO2008I ROLE = DIRMAINT

DVHPRO2010I TESTING USE OF MSGNOH ...
DVHPRO2002A Manual start is required for DIRMAINT.
DVHPRO2002A Enter DVHBEGIN when ready to start.
DIRMAINT DVHTEST1 - 2011/05/16; T=0.38/0.46 21:54:49
```

__ b. Start the server machine again

If the DIRMAINT machine was running disconnected, the DVHPROF EXEC would have automatically issued an *EXEC DVHBEGIN* command; manual intervention would not have been required. To start DirMaint manually, enter the following commands:

profile test
dvhbegin

__ 10. Verify the DIRMAINT machine will update the source directory.

Complete the test instructions from this point only when you are ready to allow DirMaint to update the source directory.

__ a. Determine your present distribution code

distrib ?

Write down the value returned from this command, it will be used in step 10d on page 247 when restoring it.

```
DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 20:00:31.
DVHREQ2290I Request is: DISTRIB ?
DVHREQ2288I Your DISTRIB request for DIRMAINT at * has been accepted.
DVHDIS3248I The current distribution code for DIRMAINT is oldvalue
DVHREQ2289I Your DISTRIB request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/16; T=0.97/1.06 20:00:40
DVHWAI2140I Waiting for work on 2011/04/16 at 20:00:40.
```

__ b. Change your present distribution code

distrib *value*

value can be any value you wish, you will be changing it back shortly.

```
DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 20:05:57.
DVHREQ2290I Request is: DISTRIB value
DVHREQ2288I Your DISTRIB request for DIRMAINT at * has been accepted.
DVHBIU3423I The source for directory entry DIRMAINT has been updated.
```

```
DVHBIU3423I The next ONLINE will take place via Diagnose 84.
DVHBIU3428I Changes made to directory entry DIRMAINT have been placed online.
DVHREQ2289I Your DISTRIB request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/16; T=1.26/1.39 20:06:09
DVHWAI2140I Waiting for work on 2011/04/16 at 20:06:10.
```

__ c. Check if the distribution code has been changed

distrib ?

```
DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 20:06:12.
DVHREQ2290I Request is: DISTRIB ?
DVHREQ2288I Your DISTRIB request for DIRMAINT at * has been accepted.
DVHDIS3248I The current distribution code for DIRMAINT is value
DVHREQ2289I Your DISTRIB request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/16; T=0.97/1.06 20:06:20
DVHWAI2140I Waiting for work on 2011/04/16 at 20:06:20.
```

__ d. Restore the original distribution code

distrib *oldvalue*
oldvalue is the original distribution code, determined in step 10a on page 246.

```
distrib oldvalue
DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 20:05:57.
DVHREQ2290I Request is: DISTRIB oldvalue
DVHREQ2288I Your DISTRIB request for DIRMAINT at * has been accepted.
DVHBIU3423I The source for directory entry DIRMAINT has been updated.
DVHBIU3423I The next ONLINE will take place via Diagnose 84.
DVHBIU3428I Changes made to directory entry DIRMAINT have been placed online.
DVHREQ2289I Your DISTRIB request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/16; T=1.26/1.39 20:06:09
DVHWAI2140I Waiting for work on 2011/04/16 at 20:06:10.
```

__ 11. Test a directory update that does not get placed online with diagnose X'84'

spool 00b 1403
spool 00b ?
spool 00b delete

```
DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 20:41:07.
DVHREQ2290I Request is: SPOOL 000B 1403
DVHREQ2288I Your SPOOL request for DIRMAINT at * has been accepted.
DVHBIU3424I The source for directory entry DIRMAINT has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHBIU3428I Changes made to directory entry DIRMAINT have been placed
DVHBIU3428I online.
DVHBIU3427I Changes made to directory entry
DVHBIU3427I DIRMAINT by DIRMAINT at DVHTEST1 have
DVHBIU3427I been placed online.
DVHREQ2289I Your SPOOL request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/16; T=1.20/1.32 20:41:18
DVHWAI2140I Waiting for work on 2011/04/16 at 20:41:18.


DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 20:43:36.
DVHREQ2290I Request is: SPOOL 000B ?
DVHREQ2288I Your SPOOL request for DIRMAINT at * has been accepted.
DVHSPL3320I The spool statement in DIRMAINT associated with virtual
DVHSPL3320I address 000B is:
DVHSPL3320I 1403
DVHREQ2289I Your SPOOL request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/16; T=1.00/1.10 20:43:45
DVHWAI2140I Waiting for work on 2011/04/16 at 20:43:45.


DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 20:49:50.
```

```
DVHREQ2290I Request is: SPOOL 000B DELETE
DVHREQ2288I Your SPOOL request for DIRMAINT at * has been accepted.
DVHBIU3424I The source for directory entry DIRMAINT has been updated.
DVHBIU3424I The next ONLINE will take place immediately.
DVHBIU3428I Changes made to directory entry DIRMAINT have been placed
DVHBIU3428I online.
DVHBIU3427I Changes made to directory entry
DVHBIU3427I DIRMAINT by DIRMAINT at DVHTEST1 have
DVHBIU3427I been placed online.
DVHREQ2289I Your SPOOL request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/16; T=1.35/1.49 20:50:04
DVHWAI2140I Waiting for work on 2011/04/16 at 20:50:04.
```

__ 12. Verify the directory has not been corrupted by any changes made

chksum

```
chksum
DVHWAI2146I Wakeup caused by console attention on 2011/04/22 at 14:24:32.
DVHREQ2290I Request is: CHKSUM
DVHREQ2288I Your CHKSUM request for DIRMAINT at * has been accepted.
DVHILZ3510I Starting DVHINITL with directory: USER DIRECT E
DVHILZ3510I DVHINITL Parms: BLDMONO
DVHREQ2289I Your CHKSUM request for DIRMAINT at * has completed; with RC = 0.
DIRMAINT DVHTEST1 - 2011/04/22; T=0.91/0.99 14:24:38
```

__ 13. Make sure the DIRMUSER user ID is authorized to issue privileged commands, then disconnect the DIRMAINT server user ID.

__ a. Issue the AUTHFOR command to add the DIRMUSER user ID as a privileged user to the AUTHFOR CONTROL file.

Note: If you are testing from a different user ID, substitute that name for DIRMUSER.

for dirmuser authfor dirmuser cmdset adghopsm cmdl 140a
for dirmuser authfor dirmuser cmdset adghopsm cmdl 150a

__ b. Disconnect the DIRMAINT server.

#cp disc

__ 14. Log on to the DIRMUSER user ID. This user ID is required to be a DirMaint privileged user in order to issue privileged DirMaint commands to properly validate your installation of DirMaint.

__ a. Link to the disk or SFS directory containing test code for the MAINT 19E minidisk

__ 1) If using minidisks:

link 6VMDIR20 29E *vaddr* **rr**
access *vaddr* **b**

Where *vaddr* is any free virtual address. This disk contains the DIRMAINT EXEC which will be used to verify that DirMaint code is satisfactory.

__ 2) If using SFS:

access VMPSFS:6VMDIR20.DIRM.MAINT19E b

This directory contains the DIRMAINT EXEC which will be used to verify that DirMaint code is satisfactory.

__ b. Edit the ACCESS DATADVH file on the B disk and change the entry for the 11F disk to 21F and file on your A disk.

```
xedit access datadvh b
locate on= *
change /11F/21F/
====> file = = a
```

If the 21F minidisk (6VMDIR20's 41F minidisk) was defined with a password other than 'ALL', be sure to make the appropriate password correction here as well.

__ c. Verify correct installation of the user machine code, issue:

```
dirmaint globalv ? interface
```

```
DVHGLB1341I The current setting for INTERFACE is 199701.150A
DVHGLB1190I Command GLOBALV complete; RC= 0.
```

__ d. Verify correct installation of the DIRMAINT service machine code and configuration of the DIRMAINT service virtual machine and issue:

```
dirmaint globalv cmdlevel 150a
dirmaint query dvhlevel
```

Where *yynn* will be the year and number of the latest RSU applied to DirMaint. If *0000*, then there have not been any RSUs applied to DirMaint.

```
DVHGLB3844I IBM Directory Maintenance Facility for z/VM (DirMaint)
DVHGLB3844I 5741-A07 (C) Copyright IBM Corporation 1979, 2011.
DVHGLB3844I Function Level 620 Service Level nnnn.
DVHREQ2289I Your QUERY request for DIRMUSER at * has completed; with RC = 0.
```

__ e. Repeat steps 14a on page 248 through 14c from the DIRMUSER user ID on each system in the CSE or SSI cluster. This will allow the test user interface code to be available to the DIRMUSER user ID on each system in the CSE or SSI cluster. Do *not* log off or disconnect from the DIRMUSER user ID yet, as it will be used to test the other DirMaint functions.

Test the DIRMSAT Server Machine

This procedure will test the new DIRMSAT code to see that it functions properly. You will log the server machine on and access the appropriate disks. The DIRMUSER user ID is required in order to verify the DIRMSAT function.

Notes

1. Log on the satellite server on a system within the cluster where the DIRMAINT server is *not* running.
   - If you are testing in a CSE cluster, you should have already defined a DIRMSAT server. (For more information on defining a DIRMSAT server, see “Directory Statements for the DIRMSAT Virtual Machine” on page 18 and “Step 1. Define a Satellite Service Machine to DIRMAINT” on page 67.)
   - If you are testing in an SSI cluster, the satellite servers are already defined for you. A satellite server named DIRMSAT is defined for the member system in slot 1 in the SSI cluster, as identified by the CP QUERY SSI command. Satellite servers named DIRMSAT2, DIRMSAT3, and DIRMSAT4 are defined for the member system in slots 2, 3 and 4, respectively.
2. Starting with step 12 on page 252, this procedure will have you make updates to the source directory and verify these changes took effect. To do this the RUNMODE= entry within an **override file** to the CONFIG DATADVH file (on the 6VMDIR20 41F minidisk) must be set to *operational*.

__ 1. Log on to the **DIRMSAT** server. See note 1 above.

__ 2. If the DIRMSAT server is currently running, it must be shutdown.

shutdown

__ 3. Link to the 6VMDIR20's test disks

profile test

You will not see a CMS Ready message. Instead, this condition has been replaced by a line with the DIRMSAT service machine's network node id, user id, the date, return code, and time of day.

__ 4. Verify the appropriate disks have been accessed

The disk mode and addresses or directory names should match the configuration specified in the DVHPROFA DIRMSAT file. If not, make the necessary corrections and run the profile exec again.

q accessed

```
Mode  Stat     Files  Vdev  Label/Directory
A      R/W         1  155   DST155
C      R/O       294  191   DRM492
D      R/O        57  11F   DRM41F
E      R/O        45  1DF   DIR1DF
F      R/O        25  551   DIR551
S      R/O       464  190   CMSESA
Y/S    R/O         3  19E   Y-DISK
Z      R/W         0  1FA   DST1FA
DIRMSAT DVHTEST1  - 2011/04/11; T=0.01/0.01 17:28:35
```

__ 5. Check for any extraneous files

This check verifies that the disks are accessed, that the message handler is functioning correctly, and that there are no back level executable files that have been misplaced in the search order. If you have a large number of duplicate files, or if you have any executable files, use the EXEC DVHUCHK CHECK FILELIST command to resolve the discrepancies before continuing.

exec dvhuchk check itemize

```
DVHUCH1371W Filemodes C and E both contain AUTHDASD DATADVH.
DVHUCH1371W Filemodes A and C both contain DIRMSAT  DATADVH.
DVHUCH1372I There were 2 extraneous files found.
DVHUCH1190I Command CHECK complete; RC= 2.
DIRMSAT DVHTEST1  - 2011/04/11(00002); T=0.40/0.43 18:11:32
```

__ 6. Start the server machine using the new code.

The DVHBEGIN EXEC will start the server.

dvhbegin

The DIRMSAT service machine should now be waiting for work.

```
dvhbegin
DIRMSAT DVHTEST1  - 2011/04/11; T=4.30/4.69 18:58:06
DVHWAI2140I Waiting for work on 2011/04/11 at 18:58:06.
```

__ 7. Verify the operation of DIRMSAT's console interrupt handler, the command syntax parser, and part of the authorization checking.

cp query files

```
DVHWAI2146I Wakeup caused by console attention on 2011/04/11 at 19:20:00.
DVHREQ2290I Request is: CP query files
DVHREQ2288I Your CP request for DIRMSAT at * has been accepted.
FILES:   NO RDR, 0001 PRT,   NO PUN
DVHREQ2289I Your CP request for DIRMSAT at * has completed; with RC = 0.
DIRMSAT DVHTEST1  - 2011/04/11; T=0.56/0.61 19:20:05
DVHWAI2140I Waiting for work on 2011/04/11 at 19:20:05.
```

__ 8. Test DirMaint's automatic restart and recovery of the DIRMSAT server

__ a. Create a CP disabled wait state.

If a restart is not successful, check the `SHUTDOWN_REIPL_COMMAND= CP IPL CMS PARM AUTOCR` entry in the CONFIG* DATADVH file(s). You may need to correct the name of the CMS saved system, or change it to IPL by device address rather than by system name. It should match the IPL statement you included in the DIRMSAT machine's directory entry, or the IPL statement included in the profile used by the DIRMSAT machine's directory entry.

cp system clear

```
cp system clear
DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 21:51:50.
DVHREQ2290I Request is: CP system clear
DVHREQ2288I Your CP request for DIRMSAT at * has been accepted.
Storage cleared - system reset.
DMSWSP314W Automatic re-IPL by CP due to disabled wait; PSW 000A0000 00000000
z/VM V6.2.0    2011-05-23 10:40
PRT FILE 1019 SENT FROM DIRMSAT CON WAS 1019 RECS 0049 CPY  001 0 HOLD    NOKEEP
  ⋮

PRODUCT:
IBM Directory Maintenance Facility for z/VM (DirMaint)
5741-A05 (C) Copyright IBM Corporation 1979, 2011.
Function Level 620 Service Level nnnn.
DMSACC724I 155 replaces A (191)

DVHPRO2008I ROLE = DIRMSAT

DVHPRO2010I TESTING USE OF MSGNOH ...
DVHPRO2002A Manual start is required for DIRMSAT.
DVHPRO2002A Enter DVHBEGIN when ready to start.
  ⋮
DIRMSAT DVHTEST1  - 2011/04/16; T=0.38/0.46 21:54:49
```

__ b. Start the server machine again

If the DIRMSAT machine was running disconnected, the DVHPROF EXEC would have automatically issued an *EXEC DVHBEGIN* command; manual intervention would not have been required.

profile test
dvhbegin

__ 9. Disconnect the user ID

#cp disc

__ 10. Repeat steps 1 on page 249 through 9 for each satellite server defined for each system in the CSE or SSI cluster where the DIRMAINT machine is *not* running.

__ 11. Verify multiple-system CSE or SSI cluster routing is properly working. Logon to a DIRMUSER user ID on a system in your cluster **other than the one where DIRMAINT is running**, and:

__ a. Send a command to the DIRMAINT server in the cluster and verify a response is received by issuing:

```
dirm distrib ?
```

__ b. Send a file to the DIRMAINT server and verify the file is returned by issuing:

```
dirm  batch
input  review  noprof
file
```

__ c. Peek the reader file returned, and verify for accuracy by issuing:

```
peek nnnn
discard
logoff
```

Where *nnnn* is the spool ID of the reader file returned from DIRMAINT.

__ 12. **Go back to the DIRMUSER user ID for the system in your cluster where DIRMAINT is running** (the user ID which you tested DIRMAINT functions). If you had previously logged off this user ID, make sure to complete the instructions in step 14a on page 248 to access test code for the MAINT 19E minidisk.

Complete the test instructions from this point only when you are ready to allow DirMaint to update the source directory.

There are two reasons for using a satellite server. If you are using a satellite server to update multiple object directories in a multiple-system CSE or SSI cluster, you can verify that a directory change has taken place by logging on to the user ID on each system in turn and checking the necessary characteristic. If you are using a satellite server to maintain a duplicate object directory on a stand-alone system, it is more difficult to completely verify successful operation. You will need to do a CP SHUTDOWN of your entire system, IPL from the alternate system residence volume, logon to the DIRMUSER id and verify that the directory changes have in fact taken place, then SHUTDOWN your system again and re-IPL from your primary system residence volume. These instructions will not have you do the system SHUTDOWN/IPL, do this at your discretion.

__ 13. To verify that the DIRMSAT service machine correctly updates the object directory, do the following:

__ a. Determine your current DATEFORMAT setting:

```
#cp query dateformat
```

Write down the value returned from this command, it will be used in step 13f on page 253 when restoring it.

__ b. Change your present DATEFORMAT setting:

```
dirmaint for dirmuser dateformat isodate
```

__ c. Verify that the DATEFORMAT setting has been changed by logging off DIRMUSER, logging back on to DIRMUSER, and issuing:

```
#cp  query  dateformat
```

Verify that the date format returned matches the value specified in the previous DIRM command (issued in step 13b).

__ d. Log on to the DIRMUSER user ID on each system where a DIRMSAT or DIRMSATn server is running and issue:

#cp query dateformat

Verify that the date format returned matches the value specified in the previous DIRM command (issued in step 13b on page 252).

Note: If you were already logged on to the DIRMUSER user ID on this system before the DIRM command in step 13b on page 252 was issued, you will have to log off and log back on for the change to take effect.

__ e. Log off the DIRMUSER user ID on each system in the cluster, except the system where DIRMAINT is running:

#cp logoff

__ f. Restore the original DATEFORMAT setting:

dirm for dirmuser dateformat *oldvalue*

Where *oldvalue* is the original date format, as determined in step 13a on page 252.

Or:

dirm for dirmuser dateformat delete

If you wish to use the system default.

__ 14. If your system is running in a multiple-system SSI cluster, continue with the following multiconfiguration virtual machine test verification steps:

__ a. Log on to the MAINT user ID and determine the system IDs associated with each slot in the SSI cluster:

#cp query ssi

Write down the value returned for the SYSTEMID associated with each slot.

Note: Following this step, *sys1*, *sys2*, *sys3*, and *sys4* will now correspond to the member system associated with slots 1, 2, 3 and 4, respectively in the QUERY SSI output.

__ b. Log on to the DIRMUSER user ID on the system associated with SLOT 1 in the QUERY SSI output. The DIRMUSER user ID should be a multiconfiguration machine defined with IDENTITY and SUBCONFIG entries.

__ c. Query the current DATEFORMAT setting for the DIRMUSER user ID on each system in the SSI cluster:

#cp query dateformat

You must issue this command separately on each member of the cluster. Write down the DATEFORMAT setting returned on each system.

__ d. Change the DATEFORMAT setting for the DIRMUSER user ID at the slot 2 member. Issue the following command from the DIRMUSER user ID on any member system in the cluster:

dirmaint for dirmuser at *sys2* **dateformat isodate**

__ e. Log off and back on to the DIRMUSER user ID on *sys2*.

__ f. Verify that the DATEFORMAT setting changed on *sys2*:

#cp query dateformat

Verify that the DATEFORMAT setting matches that used in the DIRM command in step 14d on page 253.

__ g. Log off and back on to the DIRMUSER user ID on *sys1*.

__ h. Verify that the DATEFORMAT setting did not change on *sys1*.

#cp query dateformat

Verify that the DATEFORMAT setting is the same as it was for *sys1* in step 14c on page 253.

__ i. Verify that the DATEFORMAT setting did not change on *sys3* and *sys4*. Repeat steps 14g and 14h for the DIRMUSER user ID on *sys3* and *sys4*.

__ j. Restore the original DATEFORMAT setting:

dirm for dirmuser at *sys2* **dateformat** *oldvalue*

Where *oldvalue* is the original date format on *sys2*, as determined in step 14c on page 253.

Or:

dirm for dirmuser at *sys2* **dateformat delete**

If you wish to use the system default.

__ 15. Do not log off or disconnect from the DIRMUSER user ID on the system where DIRMAINT is running, as it will be used to test the DATAMOVE server functions.

Test the DATAMOVE Server Machine

This procedure will test the new DATAMOVE code to see that it functions properly. You will log on a server machine and access the appropriate disks. The DIRMUSER user ID is required in order to verify the DATAMOVE function.

Notes

1. You will first need to log on to the appropriate DATAMOVE server.
   a. In a CSE cluster, log on to the DATAMOVE server at the node you defined on the first DATAMOVE_MACHINE= statement in your override configuration file in step 6 on page 244.
   b. In an SSI cluster, log on the DATAMOVE server at the member system associated with slot 1 in the QUERY SSI output. Refer to the DATAMOVE_MACHINE= statements in the CONFIGSS DATADVH file on the DIRMAINT 11F disk to determine which DATAMOVE server is defined for each system in the cluster.
2. To test multiple-system CSE or SSI clusters, you will need a DIRMUSER user ID on a second system in the cluster, and you should have already verified the DIRMSAT operation in "Test the DIRMSAT Server Machine" on page 249.
3. Starting with step 11 on page 257, this procedure will have you make updates to the source directory and verify these changes took effect. To do this the RUNMODE= entry within an override file to the CONFIG DATADVH file (on the 6VMDIR20 41F minidisk) must be set to *operational*.

__ 1. Log on to the **DATAMOVE** server. See note 1 above.

__ 2. If the DATAMOVE server is currently running, it must be shutdown.

shutdown

__ 3. Link to the 6VMDIR20's test disks

profile test

You will not see a CMS Ready message. Instead, this condition has been replaced by a line with the DATAMOVE service machine's network node id, user id, the date, return code, and time of day.

__ 4. Verify the appropriate disks have been accessed

The disk mode and addresses or directory names should match the configuration specified in the DVHPROFM DATADVH file. If not, make the necessary corrections and run the profile exec again.

q accessed

```
q accessed
Mode  Stat     Files  Vdev  Label/Directory
A      R/W         1  155   DAT155
C      R/O       294  191   DRM492
D      R/O        57  11F   DRM41F
F      R/O         5  551   PMA551
S      R/O       464  190   CMSESA
Y/S    R/O         3  19E   Y-DISK
Z      R/W         0  1FA   DAT1FA
DATAMOVE DVHTEST1 - 2011/04/11; T=0.01/0.01 17:28:35
```

__ 5. Check for any extraneous files

This check verifies that the disks are accessed, that the message handler is functioning correctly, and that there are no back level executable files that have been misplaced in the search order. If you have a large number of duplicate files, or if you have any executable files, use an *EXEC DVHUCHK CHECK FILELIST* command to resolve the discrepancies before continuing.

exec dvhuchk check itemize

```
exec dvhuchk check itemize
DVHUCH1371W Filemodes C and E both contain AUTHDASD DATADVH.
DVHUCH1371W Filemodes A and C both contain DATAMOVE DATADVH.
DVHUCH1372I There were 2 extraneous files found.
DVHUCH1190I Command CHECK complete; RC= 2.
DATAMOVE DVHTEST1 - 2011/04/11(00002); T=0.40/0.43 18:11:32
```

__ 6. Start the server machine using the new code.

The DVHBEGIN EXEC will start the server.

dvhbegin

The DATAMOVE service machine should now be waiting for work.

```
dvhbegin
DATAMOVE DVHTEST1 - 2011/04/11; T=4.30/4.69 18:58:06
DVHWAI2140I Waiting for work on 2011/04/11 at 18:58:06.
```

__ 7. Verify the operation of DATAMOVE's console interrupt handler, the command syntax parser, and part of the authorization checking.

cp query files

```
cp query files
DVHWAI2146I Wakeup caused by console attention on 2011/04/11 at 19:20:00.
DVHREQ2290I Request is: CP query files
DVHREQ2288I Your CP request for DATAMOVE at * has been accepted.
FILES:   NO RDR, 0001 PRT,   NO PUN
DVHREQ2289I Your CP request for DATAMOVE at * has completed; with RC = 0.
DATAMOVE DVHTEST1 - 2011/04/11; T=0.56/0.61 19:20:05
DVHWAI2140I Waiting for work on 2011/04/11 at 19:20:05.
```

__ 8. Test DirMaint's automatic restart and recovery of the DATAMOVE server

__ a. Create a CP disabled wait state.

If a restart is not successful, check the SHUTDOWN_REIPL_COMMAND= CP IPL CMS PARM AUTOCR entry in the CONFIG* DATADVH file(s). You may need to correct the name of the CMS saved system, or change it to IPL by device address rather than by system name. It should match the IPL statement you included in the DATAMOVE machine's directory entry, or the IPL statement included in the profile used by the DATAMOVE machine's directory entry.

cp system clear

```
cp system clear
DVHWAI2146I Wakeup caused by console attention on 2011/04/16 at 21:51:50.
DVHREQ2290I Request is: CP system clear
DVHREQ2288I Your CP request for DATAMOVE at * has been accepted.
Storage cleared - system reset.
DMSWSP314W Automatic re-IPL by CP due to disabled wait; PSW 000A0000 00000000
z/VM V6.2.0    2011-05-23 10:40
PRT FILE 1019 SENT FROM DATAMOVE CON WAS 1019 RECS 0049 CPY  001 0 HOLD   NOKEEP
  ⋮

PRODUCT:
IBM Directory Maintenance Facility for z/VM (DirMaint)
5741-A05 (C) Copyright IBM Corporation 1979, 2011.
Function Level 620 Service Level nnnn.
DMSACC724I 155 replaces A (191)

DVHPRO2008I ROLE = DATAMOVE

DVHPRO2010I TESTING USE OF MSGNOH ...
DVHPRO2002A Manual start is required for DATAMOVE.
DVHPRO2002A Enter DVHBEGIN when ready to start.
  ⋮
DATAMOVE DVHTEST1 - 2011/04/16; T=0.38/0.46 21:54:49
```

__ b. Start the server machine again

If the DATAMOVE machine was running disconnected, the DVHPROF EXEC would have automatically issued an *EXEC DVHBEGIN* command; manual intervention would not have been required.

profile test
dvhbegin

__ 9. Disconnect the user ID

#cp disc

__ 10. Repeat steps 1 on page 254 through 9 for each DATAMOVE server defined for each system in the CSE or SSI cluster. Refer to the DATAMOVE_MACHINE= statements configured in the associated override configuration file to determine which DATAMOVE servers are defined for which systems in the cluster.

__ 11. Log on to the DIRMUSER user ID on any system in the CSE or SSI cluster. If you had previously logged off this user ID, make sure to complete the instructions in step 14a on page 248 to access test code for the MAINT 19E minidisk.

Complete the test instructions from this point only when you are ready to allow DirMaint to update the source directory.

You will also want to make sure directory changes are placed online immediately for verification steps.

__ a. To define available DASD extents for creating test disks to the DIRMAINT server, issue the following:

dirmaint dasd add region *regname volid devtype size start*

Add a region of free DASD space to be used to test DATAMOVE server functions to the DirMaint EXTENT CONTROL file – where *regname* is the name of the region, *volid* specifies the volume ID placed in the *volser* field of any volume with at least 1 cylinder (on any CKD device) or 64 blocks (on any FB-512 device) of available space, *devtype* specifies the DASD type associated with the volume, *size* specifies the number of blocks or cylinders associated with the free space, and *start* specifies the starting block or cylinder.

dirmaint rldext

Place the EXTENT CONTROL file changes into use.

__ b. Verify that the DIRMAINT server will allocate DASD by issuing the following:

dirmaint online immed
dirmaint for dirmuser amdisk *nnn* **x autor 1** *regname*

Where *nnn* is the device address that does not already exist for the DIRMUSER user ID, and *regname* is the region name defined in step 11a. If using FB-512 DASD, substitute 64 for 1.

__ c. Wait for the messages to indicate that the minidisk has been added and transferred to the DIRMUSER user ID, then issue the following:

cp link * *nnn nnn* **mr**
access *nnn* **z**

Where *nnn* is the device address used in the previous AMDISK command for the DIRMUSER user ID. This will verify that the minidisk exists and has not been formatted.

__ d. Have DIRMAINT delete the requested DASD, issue the following:

dirmaint for dirmuser dmdisk *nnn* **noclean**

Where *nnn* is the device address used in the previous AMDISK command for the DIRMUSER user ID.

The expected result will be messages from DIRMAINT indicating that the specified device has been deleted and that the space is available for re-allocation.

__ e. Wait for the messages to indicate that the minidisk has been deleted, then issue:

cp detach *nnn*
cp link * *nnn nnn* **mr**

Where *nnn* is the device address used in the previous AMDISK command for the DIRMUSER user ID. The minidisk should not exist.

__ f. Verify that the DIRMAINT server will allocate and DATAMOVE server will format DASD by issuing:

dirmaint for dirmuser amdisk *nnn* **x autor 1** *regname* **label** *label*

Where *nnn* is any legal device address that does not already exist for the DIRMUSER user ID, *regname* is the region name defined in step 11a on page 257, and *label* is the label to be assigned to the minidisk being allocated.

__ g. Wait for the messages to indicate that the minidisk has been added to DATAMOVE, the minidisk has been formatted, and transferred to the DIRMUSER user ID, then issue:

cp link * *nnn nnn* **mr**
access *nnn* **z**

Where *nnn* is the device address used in the previous AMDISK command for the DIRMUSER user ID. This will verify that the minidisk exists and has been formatted by DATAMOVE.

__ h. Have DATAMOVE format the requested DASD being deleted, issue:

cp detach *nnn*
dirmaint for dirmuser dmdisk *nnn* **clean**

Where *nnn* is the device address used in the previous AMDISK command for the DIRMUSER user ID.

The expected result will be a message from DIRMAINT indicating that the specified device has been transferred to the DATAMOVE machine for DATA security clean-up, followed by more messages from DIRMAINT indicating that the specified device has been deleted from the DATAMOVE machine and that the space is available for re-allocation.

__ i. Wait for the messages to indicate that the minidisk has been deleted, then issue:

cp link * *nnn nnn* **mr**

Where *nnn* is the device address used in the previous AMDISK command for the DIRMUSER user ID. The minidisk should not exist.

__ 12. If your system is running in a multiple system SSI cluster, continue with the following SSI cluster test verification steps. You can issue the following commands from the DIRMUSER user ID on any of the member systems in the cluster. This step will test the DATAMOVE server defined for the member systems associated with slots 2, 3, and 4 in the QUERY SSI output. This step should be repeated for all remaining slots in your SSI cluster.

__ a. Verify on a specific system in the cluster that the DIRMAINT server will allocate and the DATAMOVE server will format DASD, by issuing:

dirmaint for *subconfig-n* **at** *sysn* **amdisk** *nnn* **x autor 1** *regname* **label** *label*

Where *sysn* is the system ID of the member system in slot 2, 3, or 4 of the QUERY SSI output, *subconfig-n* is the name of the SUBCONFIG entry for the DIRMUSER user ID associated with system *sysn*, *nnn* is any legal device address that does not already exist for the DIRMUSER user ID, *regname* is the region name defined in step 11a on page 257, and *label* is the label to be assigned to the minidisk being allocated.

__ b. Wait for the messages to indicate that the minidisk has been added to DATAMOVE, the minidisk has been formatted, and transferred to the DIRMUSER user ID on the *sysn* system in the cluster. Then issue the following from the DIRMUSER user ID on the *sysn* system in the cluster:

cp detach *nnn*
cp link * *nnn nnn* **mr**
access *nnn* **z**

Where *nnn* is the device address used in the previous AMDISK command for the DIRMUSER user ID on the *sysn* system. This will verify that the minidisk exists and has been formatted by DATAMOVE.

__ c. Verify that the minidisk was *not* added to the other member systems in the cluster. Log on to the DIRMUSER user ID on each of the other systems in the cluster (besides *sysn*) and issue the following:

cp link * *nnn nnn* **mr**

This will verify that the minidisk does *not* exist.

__ d. Have the DATAMOVE server on the *sysn* system in the cluster format the requested DASD being deleted, by issuing:

cp detach *nnn*
dirmaint for *subconfig-n* **at** *sysn* **dmdisk** *nnn* **clean**

Where *subconfig-n*, *sysn* and *nnn* are the SUBCONFIG name, system ID and device address used in the previous AMDISK command for the DIRMUSER user ID in step 12a on page 259.

__ e. Wait for the messages to indicate that the minidisk has been transferred to DATAMOVE, the minidisk has been formatted, and has been deleted from DATAMOVE on the *sysn* system in the cluster. Then issue the following from DIRMUSER on the *sysn* system in the cluster:

cp link * *nnn nnn* **mr**

Where *nnn* is the device address used in the previous AMDISK command for the DIRMUSER user ID in step 12a on page 259.

__ f. Log off the DIRMUSER user ID on the *sysn* system in the cluster.

logoff

Logoff should only be issued if you are finished testing the DATAMOVE function for each system in the cluster.

Post Test Instructions

Verification of DirMaint is now complete. You will need to do some final clean up.

__ 1. Erase the ACCESS DATADVH file created on DIRMUSER's A disk. This should be done on each system in the cluster.

erase access datadvh a

__ 2. You can log off the DIRMUSER user ID

__ 3. At this point, it is advisable to log off all DirMaint server machines.

__ 4. Once DirMaint is in production, the supplied sample files can be modified and placed in production without bringing down the DirMaint server's. This way, the system files can be maintained on the test disk and immediately placed into production without the need for shutting the servers down.

From the 6VMDIR20 user ID, issue:

__ a. Make the desired changes to tailorable system files on the 492 or 41F test minidisk.

__ b. Send the updated system file to the DIRMAINT server and have it replace the file on the appropriate production minidisk.

dirm file *fn ft fm*

Where *fn ft fm* is the file identification and location of the system file you wish to replace on the appropriate production minidisk.

__ c. Have the DIRMAINT server place the new system files into use

dirm rlddata

What's Next?

Proceed with:

To place the new DirMaint code into production after testing during initial installation, see "Section 6.4, Place DirMaint Tailored Files Into Production" of the *Dirmaint Program Directory*.

OR

To replace DirMaint code running in production with the new code after testing while servicing your system, see "Section 7.5, Place Serviced DirMaint Into Production" of the *DirMaint Program Directory*.

Quick Test After Installing Service

The steps in "Test the DIRMAINT Server Machine" on page 243, "Test the DIRMSAT Server Machine" on page 249, and "Test the DATAMOVE Server Machine" on page 254 are specifically designed to systematically test the functions of the DirMaint product in a sequence such that the point of failure, if any, isolates the cause and identifies the necessary corrective action. After a period of production, there is small likelihood that this procedure will encounter any problem, and is unnecessarily involved for routine regression testing.

This procedure provides an easier alternative. If the test is successful, testing is complete. If the test fails, there is no indication of the cause; return to "Test the DIRMAINT Server Machine" on page 243 and follow the more detailed steps to isolate the cause and identify the solution.

__ 1. Switch the DirMaint service machines from using the production code to use the code to be tested. Create a PROD2TST DVHBATCH file, for example, containing the following commands:

```
OFFLINE
DATAMOVE SHUTDOWN
SATELLITE SHUTDOWN
FOR DATAMOVE LINK 6VMDIR20 491 191 DELETE
FOR DATAMOVE LINK 6VMDIR20 11F 11F DELETE
FOR DATAMOVE LINK 6VMDIR20 492 192 DELETE
FOR DATAMOVE LINK 6VMDIR20 41F 21F DELETE
FOR DIRMSAT LINK 6VMDIR20 491 191 DELETE
FOR DIRMSAT LINK 6VMDIR20 11F 11F DELETE
FOR DIRMSAT LINK 6VMDIR20 492 192 DELETE
FOR DIRMSAT LINK 6VMDIR20 41F 21F DELETE
FOR DIRMAINT LINK 6VMDIR20 491 191 DELETE
FOR DIRMAINT LINK 6VMDIR20 11F 11F DELETE
FOR DIRMAINT LINK 6VMDIR20 492 192 DELETE
FOR DIRMAINT LINK 6VMDIR20 41F 21F DELETE
FOR DATAMOVE LINK 6VMDIR20 492 191 RR
FOR DATAMOVE LINK 6VMDIR20 41F 11F RR
FOR DATAMOVE LINK 6VMDIR20 491 192 RR
FOR DATAMOVE LINK 6VMDIR20 11F 21F RR
FOR DIRMSAT LINK 6VMDIR20 492 191 RR
FOR DIRMSAT LINK 6VMDIR20 41F 11F RR
FOR DIRMSAT LINK 6VMDIR20 491 192 RR
FOR DIRMSAT LINK 6VMDIR20 11F 21F RR
FOR DIRMAINT LINK 6VMDIR20 492 191 MR
FOR DIRMAINT LINK 6VMDIR20 41F 11F MR
FOR DIRMAINT LINK 6VMDIR20 491 192 MR
FOR DIRMAINT LINK 6VMDIR20 11F 21F MR
FOR DATAMOVE REVIEW NOPROF
FOR DIRMSAT REVIEW NOPROF
FOR DIRMAINT REVIEW NOPROF
CP MSG <your_id> PROD2TST is done!
```

Note: The `FOR DATAMOVE` commands in this batch file must be duplicated for each DATAMOVE server, and the `FOR DIRMSAT` commands must be duplicated for each defined satellite server.

__ 2. Submit this batch job to the DIRMAINT server using these commands:

```
DIRM BATCH PROD2TST DVHBATCH
```

__ 3. When the job has completed successfully, and you have reviewed the returned reader files verifying the LINK statements are correct, issue:

```
DIRM DIRECT
DIRM SHUTDOWN
```

__ 4. Restart the DIRMAINT and DIRMSAT servers. Note that the DATAMOVE machines will be autologged by DirMaint when they are used.

a. Issue the following commands:

```
CP XAUTOLOG DIRMAINT
CP SLEEP 1 MIN
```

b. Then issue the following command for each DIRMSAT server configured on each system where DIRMAINT is not running.

CP AT *sysn* **CMD XAUTOLOG** *dirmsat_id*

Where *dirmsat_id* is the user ID of a DIRMSAT server where DIRMAINT is not running and *sysn* is the system associated with the DIRMSAT server on the SATELLITE_SERVER= statement in your override configuration file.

__ 5. Run a test that exploits most of DirMaint's critical code paths.
Issue these commands:

DIRM ONLINE
DIRM FOR <any_id> AMDISK <555> <3390> AUTOG <1> <groupname> M

You may substitute other values for the virtual address, device type, size, or groupname; as appropriate for your installation. This tests the communication paths between the originating user and the DIRMAINT server, the parser, message handler, and the path between DIRMAINT and the DIRMSAT servers, if any.

__ 6. Reverse the previous step by issuing these command:

DIRM FOR <any_id> DMDISK <555> CLEAN

This retests the same paths as the previous step, and adds the communication paths between DIRMAINT and the DATAMOVE server, and from DATAMOVE back to DIRMAINT.

__ 7. Switch the DirMaint service machines from using the test code back to use the production code to be tested.
Create a TST2PROD DVHBATCH file, for example, containing these commands:

```
OFFLINE
DATAMOVE SHUTDOWN
SATELLITE SHUTDOWN
FOR DATAMOVE LINK 6VMDIR20 492 191 DELETE
FOR DATAMOVE LINK 6VMDIR20 41F 11F DELETE
FOR DATAMOVE LINK 6VMDIR20 491 192 DELETE
FOR DATAMOVE LINK 6VMDIR20 11F 21F DELETE
FOR DIRMSAT LINK 6VMDIR20 492 191 DELETE
FOR DIRMSAT LINK 6VMDIR20 41F 11F DELETE
FOR DIRMSAT LINK 6VMDIR20 491 192 DELETE
FOR DIRMSAT LINK 6VMDIR20 11F 21F DELETE
FOR DIRMAINT LINK 6VMDIR20 492 191 DELETE
FOR DIRMAINT LINK 6VMDIR20 41F 11F DELETE
FOR DIRMAINT LINK 6VMDIR20 491 192 DELETE
FOR DIRMAINT LINK 6VMDIR20 11F 21F DELETE
FOR DATAMOVE LINK 6VMDIR20 491 191 RR
FOR DATAMOVE LINK 6VMDIR20 11F 11F RR
FOR DATAMOVE LINK 6VMDIR20 492 192 RR
FOR DATAMOVE LINK 6VMDIR20 41F 21F RR
FOR DIRMSAT LINK 6VMDIR20 491 191 RR
FOR DIRMSAT LINK 6VMDIR20 11F 11F RR
FOR DIRMSAT LINK 6VMDIR20 492 192 RR
FOR DIRMSAT LINK 6VMDIR20 41F 21F RR
FOR DIRMAINT LINK 6VMDIR20 491 191 MR
FOR DIRMAINT LINK 6VMDIR20 11F 11F MR
FOR DIRMAINT LINK 6VMDIR20 492 192 MR
FOR DIRMAINT LINK 6VMDIR20 41F 21F MR
FOR DATAMOVE REVIEW NOPROF
FOR DIRMSAT REVIEW NOPROF
FOR DIRMAINT REVIEW NOPROF
CP MSG <your_id> TST2PROD is done!
```

Note: The `FOR DATAMOVE` commands in this batch file must be duplicated for each DATAMOVE server, and the `FOR DIRMSAT` commands must be duplicated for each defined satellite server.

__ 8. Submit this batch job to the DIRMAINT server by issuing these command:

```
DIRM BATCH TST2PROD DVHBATCH
```

__ 9. When the job has completed successfully, and you have reviewed the returned reader files verifying the LINK statements are correct, issue these commands:

```
DIRM DIRECT
DIRM ONLINE
DIRM SHUTDOWN
```

What's Next?

To replace DirMaint code running in production with the new code after testing while servicing your system, see see “Section 7.5, Place Serviced DirMaint Into Production” of the *DirMaint Program Directory*.

Appendix H. DirMaint Tailorable and Non-Tailorable System Files

This table lists the DirMaint files that may be tailored during the installation procedure.

Table 51. DirMaint Tailorable System Files

File Name	Description	Page
ACCESS DATADVH	Identifies where the DirMaint common user code resides.	For more information, see "The ACCESS DATADVH File" on page 107.
AUTHDASD DATADVH	Determines who can allocate space in what DASD groups, regions, or volumes.	For more information, see "The AUTHDASD DATADVH Control File" on page 84.
AUTHFOR CONTROL	Identifies what user ID's have delegated authority for another user ID to act for them, and what command sets are included in that authority. Maintained by using the DIRM AUTHFOR and DIRM DROPFOR commands.	For more information, see "AUTHFOR CONTROL File" on page 123. For more information on the AUTHFOR and DROPFOR commands, see the *z/VM: Directory Maintenance Facility Commands Reference*.
AUTHLINK CONTROL	Identifies what user ID's are allowed to issue DIRM LINK commands without having to know the minidisk password for the disk being linked. Maintained by using the DIRM AUTHLINK command.	For more information on the AUTHLINK command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
AUTHSCIF CONTROL	Contains a list of user ID's which are allowed to issue the DIRM SECUSER command. This file is maintained by the DIRM AUTHSCIF command.	For more information on the AUTHSCIF and SECUSER commands, see the *z/VM: Directory Maintenance Facility Commands Reference*.
AUTOMAIL DATAADVH	Contains the minidisk audit notice file. The Uppercase English version of this file has a filetype of DATAUDVH.	For more information, see "Language Dependent Configuration Entries" on page 229. For more information, see the AUTOMAIL $DATADVH source file.
CONFIG DATADVH	Is the configuration file for the DirMaint servers. Multiple configuration files are allowed, providing the ability to override the supplied sample configuration file without actually altering it. Configuration file records are read starting with filenames CONFIG99-CONFIG0, CONFIGZZ-CONFIGA, then CONFIG DATADVH.	For more information, see "CONFIG DATADVH" on page 28.
DATAMOVE DATADVH	Contains a schedule of events for the DATAMOVE server.	For more information, see "DATAMOVE DATADVH" on page 65.
DIRECTXA DATADVH	Contains a list of all z/VM directory statements.	For more information, see the DIRECTXA $DATADVH source file.

Table 51. DirMaint Tailorable System Files (continued)

File Name	Description	Page
DIRMAINT DATADVH	Contains a schedule of events for the DIRMAINT server.	For more information, see “DIRMAINT DATADVH” on page 43.
DIRMAINT NEWMAIL	Is a mail file which may contain system information about new or changed function or installation policy or any other general DirMaint information an installation would like distributed to users. Maintained by using the DIRM MAIL command.	For more information on the MAIL command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
DIRMSAT DATADVH	Contains a schedule of events for the DIRMSAT server.	For more information, see “DIRMSAT DATADVH” on page 70.
DVHBHEAD DATAADVH	Contains the batch header file. The Uppercase English version of this file has a filetype of DATAUDVH. The Kanji version of this file has a filetype of DATAKDVH.	For more information, see “Language Dependent Configuration Entries” on page 229. For more information, see the DVHBHEAD $DATADVH source file.
DVHLINK EXCLUDE	Contains a list of minidisk addresses, and their owners, that are excluded from the DVHLINK FILE (contains a list of user ID's who have a link to a minidisk). Maintained by using the DIRM USEROPTN command.	For more information, see “DVHLINK EXCLUDE” on page 48. For more information on the USEROPTN command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
DVHMENUS DATAADVH	Contains the DirMaint menu source. The Uppercase English version of this file has a filetype of DATAUDVH.	For more information, see “National Language Support” on page 113.
DVHNAMES DATADVH	Identifies user ID's which are to be notified when any significant event occurs in any of the DirMaint servers.	For more information, see “DVHNAMES DATADVH” on page 46.
DVHPROFA *	Determines which minidisks (or SFS Directories) are accessed by the DIRMAINT, DATAMOVE, and DIRMSAT servers and at what filemode during initialization. The filetype **must be** the same as the user ID name the server is running on.	For more information, see “DVHPROFA DIRMAINT” on page 28.
DVHPROFM DATADVH	Determines which minidisks (or SFS Directories) are accessed by the DATAMOVE server and at what filemode during initialization. This file is used if there is no DVHPROFA file with a filetype matching the user ID name the DATAMOVE server is running on.	For more information, see “DVHPROFA DIRMAINT” on page 28.
EXTENT CONTROL	Provides information required for DirMaint DASD management functions.	For more information, see “The Extent Control File” on page 74.
PROFILE EXEC	Contains the PROFILE EXEC for the DirMaint servers.	For more information, see the PROFILE $EXEC source file.
PROFILE XEDIT	Determines characteristics of your edit sessions.	For more information, see “PROFILE XEDIT” on page 27.

Table 51. DirMaint Tailorable System Files (continued)

File Name	Description	Page
PWMON CONTROL	Contains a list of user ID's whose passwords do not expire, or do not receive password expiration notices, or have their password expiration notices sent to an alternate user ID. Maintained by using the DIRM PWMON command.	For more information, see "PWMON CONTROL" on page 49. For more information on the PWMON command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
PWLNOLCK DATAADVH	Contains the password expired (without lockout) notice file. The Uppercase English version of this file has a filetype of DATAUDVH.	For more information, see "Language Dependent Configuration Entries" on page 229 and the PWLNOLCK $DATAADV source file.
PWLOCKED DATAADVH	Contains the password expired (with lockout) notice file, sent to the affected user ID. The Uppercase English version of this file has a filetype of DATAUDVH.	For more information, see "Language Dependent Configuration Entries" on page 229 and the PWLOCKED $DATAADV source file.
PWLOTHER DATAADVH	Contains the password expired (with lockout) notice file, sent to the surrogate user ID listed in the PWMON CONTROL file. The Uppercase English version of this file has a filetype of DATAUDVH.	For more information, see "Language Dependent Configuration Entries" on page 229 and the PWLOTHER $DATAADV source file.
PWWB4LCK DATAADVH	Contains the password will expire (with lockout) notice file, sent to the affected user ID. The Uppercase English version of this file has a filetype of DATAUDVH.	For more information, "Language Dependent Configuration Entries" on page 229 and the PWWB4LCK $DATAADV source file.
PWWNOLCK DATAADVH	Contains the password will expire (without lockout) notice file, sent to the affected user ID. The Uppercase English version of this file has a filetype of DATAUDVH.	For more information, see "Language Dependent Configuration Entries" on page 229 and the PWWNOLCK $DATAADV source file.
PWWOTHER DATAADVH	Contains the password will expire (without lockout) notice file, sent to the surrogate user ID listed in the PWMON CONTROL file. The Uppercase English version of this file has a filetype of DATAUDVH.	For more information, see "Language Dependent Configuration Entries" on page 229 and the PWWOTHER $DATAADV source file.
RPWLIST DATA	Contains a list of prohibited passwords.	For more information, see "RPWLIST DATA" on page 50 and see *z/VM: CP Commands and Utilities Reference*.
SUBSCRIB DATADVH	Contains a list of subscribers to be notified when any userid, or particular userids, are changed. Maintained by the SUBSCRIBE command.	For more information, see "SUBSCRIB DATADVH" on page 51, and for more information on the SUBSCRIBE command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
USER INPUT	Is the existing source directory file.	For more information, see "USER INPUT" on page 55 and see *z/VM: CP Planning and Administration*.

Table 51. DirMaint Tailorable System Files (continued)

File Name	Description	Page
150ASERV MSGADVH	Contains the American English message repository for the DirMaint servers.	For more information, see “Language Dependent Configuration Entries” on page 229 and “Overriding and Supplementing DirMaint Messages” on page 59.
150AUSER MSGADVH	Contains the American English message repository for the user interface code.	For more information, see “Language Dependent Configuration Entries” on page 229 and “Overriding and Supplementing DirMaint Messages” on page 59.
150ASERV MSGKDVH	Contains the Kanji message repository for the DirMaint servers.	For more information, see “Language Dependent Configuration Entries” on page 229.
150AUSER MSGKDVH	Contains the Kanji message repository for the user interface code.	For more information, see “Language Dependent Configuration Entries” on page 229.
140CMDS DATADVH	Describes the 1.4 compatibility command set.	For more information, see the 140CMDS $SAMPDVH source file.
150CMDS DATADVH	Describes the 1.5 compatibility command set.	For more information, see the 150CMDS $SAMPDVH source file.

This table lists the DirMaint files which are created and maintained by the DirMaint servers. These files **should not** be directly modified.

Table 52. DirMaint Non-Tailorable System Files

File Name	Description	Page
ACTIVE DATADVH	Contains a list of all directory statements from the DIRECTXA DATADVH file plus any additional added by the DIRM DEFINESTAG command.	For more information, see the DIRECTXA $DATADVH source file. For more information on the DEFINESTAG command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
AUTOBLK CONTROL	Generated from the :AUTOBLOCK. section of the EXTENT CONTROL file.	For more information, see “The Extent Control File” on page 74.
CLST* CLUSTER	Contains clusters of user ID's from the monolithic source directory. This file is built using the load list format of the directory. The last 4-characters of the filename will be the cluster number starting at offset 1.	N/A
CMDSTATE FILE	Used for maintaining the state of command processing.	N/A
DATAMOVE CONTROL	Contains the primary control structure for managing interaction with the DATAMOVE machine. Built from entries contained in the CONFIG* DATADVH files.	For more information, see “DATAMOVE Control File” on page 92.

Table 52. DirMaint Non-Tailorable System Files (continued)

File Name	Description	Page
DEFAULT CONTROL	Generated from the :DEFAULTS. section of the EXTENT CONTROL file.	For more information, see "The Extent Control File" on page 74.
DEFAULTS DATADVH	Defines the default maximum size for various DASD devices.	For more information, see "The Extent Control File" on page 74.
DIRMAINT TRANSLOG	Contains the DirMaint servers transaction message logs for the latest month.	For more information, see the CONFIG $SAMPDVH source file.
DIRMAINT TLOG*yymm*	Contains the DirMaint servers transaction message logs for previous months.	For more information, see the CONFIG $SAMPDVH source file.
DIRMSAT CONTROL	Contains the primary control structure for managing interaction with the DIRMSAT Server. Built from entries contained in the CONFIG* DATADVH files.	For more information, see the CONFIG $SAMPDVH source file.
DISABLE CONTROL	Contains the date and time when the DIRM DISABLE command was issued. This file is erased when the DIRM ENABLE command is issued.	For more information on the ENABLE and DISABLE commands, see the *z/VM: Directory Maintenance Facility Commands Reference*.
DVHBATCH CONTROL	Used for maintaining the state of batch processing.	N/A
DVHBATCH QUEUE	Contains the batch requests which have not yet been processed. This file is appended to when a reader file is received after a DIRM BATCH command is issued.	For more information on the BATCH command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
DVHCEXIT CALLINFO	Contains a timestamped record of exits called by a DirMaint server, or user interface code, and the return code passed back from the exit.	N/A
DVHLINK ATTEMPT	Contains a list of minidisk link attempts which were rejected. This file is appended to when a DIRM LINK command fails. The DIRM ELINK command can be used to provide maintenance on this file.	For more information on the ELINK and LINK commands, see the *z/VM: Directory Maintenance Facility Commands Reference*.
DVHLINK FILE	Contains a list of of user ID's who have a LINK to a minidisk. Maintained by using the DIRM DLINK and DIRM LINK commands.	For more information on the DLINK and LINK commands, see the *z/VM: Directory Maintenance Facility Commands Reference*.
DVHLOCK DATADVH	Contains a list of device addresses or user ID's which are locked. Maintained by the DIRM LOCK and DIRM UNLOCK commands.	For more information on the LOCK and UNLOCK commands, see the *z/VM: Directory Maintenance Facility Commands Reference*.

Table 52. DirMaint Non-Tailorable System Files (continued)

File Name	Description	Page
DVHPWUSE DATADVH	Contains a history of user ID passwords for the timeframe specified by the PW_REUSE_INTERVAL= configuration file entry. Passwords for all user ID's on the system are contained in this file. This file is maintained by the supplied (DVHPXV) local password screening exit, and by the supplied (DVHXPN) local logon password notification exit.	For more information, see the CONFIG $SAMPDVH source file.
DVHRETRY QUEUE	Contains a list of work in progress when DirMaint previously shutdown due to an error encountered.	For more information, see “Restart or Shutdown Processing After Encountering an Error” on page 60.
DVHSCAN OUTPUT	Created during DIRM SCAN command processing to hold the requested output of that command.	For more information on the SCAN command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
DVHSHUTX CONTROL	Contains two counters and an action indicator used during shutdown/restart processing when errors are encountered.	For more information, see “Restart or Shutdown Processing After Encountering an Error” on page 60.
DVHSLVL DATADVH	Contains the current RSU service level for the DirMaint server's.	Maintained on the RSU only.
DVHULVL DATADVH	Contains the current RSU service level for the user interface code.	Maintained on the RSU only.
EXCLUDE CONTROL	Generated from the :EXCLUDE. section of the EXTENT CONTROL file.	For more information, see “The Extent Control File” on page 74.
GROUP CONTROL	Generated from the :GROUPS. section of the EXTENT CONTROL file.	For more information, see “The Extent Control File” on page 74.
LOCAL DATADVH	Contains a list of all directory statements added by the DIRM DEFINESTAG command.	For more information on the DEFINESTAG command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
OFFLINE CONTROL	Contains the date and time when the DIRM OFFLINE command was issued. This file is erased when the DIRM ONLINE command is issued.	For more information on the OFFLINE and ONLINE commands, see the *z/VM: Directory Maintenance Facility Commands Reference*.
OPTIONS DATADVH	Defines the keywords on the OPTION directory statement. Is used during DIRM OPTION command processing.	For more information on the OPTION command, see the *z/VM: Directory Maintenance Facility Commands Reference*. For more information, see the AUTOMAIL $DATADVH source file.
PASSCHNG DATADVH	Contains a list of minidisk password change requests. This file is maintained by the supplied (DVHXMN) minidisk password notification exit.	N/A

Table 52. DirMaint Non-Tailorable System Files (continued)

File Name	Description	Page
PASSWORD CHANGE	Created when the DIRM PWGEN command is issued. This file contains a list of randomly generated passwords to alter the existing user ID passwords.	For more information on the PWGEN command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
PENDING MESSAGES	Contains a list of messages to be issued when the next directory online takes place.	N/A
PENDING FREEDEV	Contains a list of devices to be freed from a device table (*FDEV DEVTABLE).	For more information, see "xxxxFDEV DVHTABLE File" on page 93.
PURGES PENDING	Contains a list of user ID purge requests which have not yet been processed. This file is appended to when a DIRM PURGE command is received.	For more information on the PURGE command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
PWMINFO LOCKLIST	Contains the distribution list of user ID's to be sent a password expired notice and have their passwords changed to NOLOG. This list of users will be sent the password expired notice and be NOLOGed if the CONFIG DATADVH entry `PW_LOCK_MODE=AUTOMATIC`.	For more information, see "Step 4. Select Password Control Characteristics" on page 37.
PWMINFO WARNLIST	Contains the distribution list of user ID's to be sent a password warning notice. This list of users will be sent the password warning notice if the CONFIG DATADVH entry `PW_WARN_MODE=AUTOMATIC`.	For more information, see "Step 4. Select Password Control Characteristics" on page 37.
PWMON LOCKLIST	Contains a list of user ID's to be locked out of the system. This list is sent to, intended to be edited by, the system administrator. The DIRM PWMON command is used to send this file back to DirMaint for processing.	For more information, see "Step 4. Select Password Control Characteristics" on page 37. For more information on the PWMON command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
REGION CONTROL	Generated from the :REGIONS. section of the EXTENT CONTROL file.	For more information, see "The Extent Control File" on page 74.
SSIVOL CONTROL	Generated from the :SSI_VOLUMES. section of the EXTENT CONTROL file.	For more information, see "The Extent Control File" on page 74.
UNASSIGN DVHQUEUE	Contains a list of unassigned Work Unit Control Files (WUCF).	For more information, see "Unassigned Queue" on page 93.
USER BACKOLD	Is the previous backup of the monolithic version of the source directory file.	For more information on the BACKUP command, see the *z/VM: Directory Maintenance Facility Commands Reference*.

Table 52. DirMaint Non-Tailorable System Files (continued)

File Name	Description	Page
USER BACKUP	Is the backup of the monolithic version of the source directory file.	For more information on the BACKUP command, see the *z/VM: Directory Maintenance Facility Commands Reference*.
USER DIRECT	Manages the cluster files, containing pointers to the cluster files for each user ID in the directory.	N/A
WHERETO DATADVH	Identifies the node in a multi-system cluster on which the DIRMAINT server is currently running.	N/A
$DIRGRP$ DIRMPART	Contains the GLOBALDEFS statement which denotes the start of the global definition section of the source directory. The GLOBALDEFS Directory Control Statement contains the GLOBALOPTS and POSIXGROUP Directory Control Statements. Maintained by using the DIRM GLOBALOPTS and DIRM POSIXGROUP commands.	For more information on the GLOBALOPTS and POSIXGROUP commands, see the *z/VM: Directory Maintenance Facility Commands Reference*.
* DIRMPART	Contains additions or changes to a user directory. The filename will be the user ID of the directory entry. All DIRMPART files are merged into the CLUSTER files during nightly back up processing.	N/A
* DVHPWUSE	Contains a history of user ID passwords, one file for each user ID in the system, for the timeframe specified by the `PW_REUSE_INTERVAL=` configuration file entry. These files are maintained by the supplied (DVHPXV) local password screening exit, and by the supplied (DVHXPN) local logon password notification exit.	N/A
* NAMES	Contains the distribution list to be notified when a DirMaint server starts up or shuts down. Generated from the DVHNAMES DATADVH file. The filename will be DIRMAINT, DIRMSAT, and DATAMOVE for the three different servers.	For more information, see “DVHNAMES DATADVH” on page 46.
* RDRFILE	Contains the data file associated with a reader file request. The filename will be the spool file ID of the reader file request.	N/A
* VCONTROL	Contains information that pertains to a particular DASD volume. One of these files exists for each known DASD volume on the system. The filename will be the DASD volume ID.	For more information, see “Work Unit Control File” on page 89.

Table 52. DirMaint Non-Tailorable System Files (continued)

File Name	Description	Page
* WORKSAVE	Contains the WORKUNIT file after completion of processing if saving was specified in the CONFIG* DATADVH files.	For more information, see “Work Unit Control File” on page 89.
* WORKUNIT	Contains the basic unit of work managed on the DASD subsystem, also known as the Work Unit Control File (WUCF). The filename will be an eight digit number uniquely identifying the work unit.	For more information, see “Work Unit Control File” on page 89.
* WUCFFAIL	Contains a history of what commands were performed and which commands failed for a basic unit of work managed on the DASD subsystem. The filename will be an eight digit number uniquely identifying the work unit.	For more information, see “Work Unit Control File” on page 89.
*FDEV DVHTABLE	Contains a bit map describing what devices are currently in use in the associated DATAMOVE machine. The first 4-characters of the filename will be a number which is assigned to a specific DATAMOVE machine.	For more information, see “xxxxFDEV DVHTABLE File” on page 93.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant 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 character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsurama, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you 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:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this 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 of 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 illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Programming Interface Information

This 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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at IBM copyright and trademark information - United States (www.ibm.com/legal/us/en/copytrade.shtml).

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Glossary

For a list of z/VM terms and their definitions, see *z/VM: Glossary*.

The z/VM glossary is also available through the online z/VM HELP Facility. For example, to display the definition of the term “dedicated device”, issue the following HELP command:

```
help glossary dedicated device
```

While you are in the glossary help file, you can do additional searches:

- To display the definition of a new term, type a new HELP command on the command line:

  ```
  help glossary newterm
  ```

  This command opens a new help file inside the previous help file. You can repeat this process many times. The status area in the lower right corner of the screen shows how many help files you have open. To close the current file, press the Quit key (PF3/F3). To exit from the HELP Facility, press the Return key (PF4/F4).
- To search for a word, phrase, or character string, type it on the command line and press the Clocate key (PF5/F5). To find other occurrences, press the key multiple times.

  The Clocate function searches from the current location to the end of the file. It does not wrap. To search the whole file, press the Top key (PF2/F2) to go to the top of the file before using Clocate.

Bibliography

See the following publications for additional information about z/VM. For abstracts of the z/VM publications, see *z/VM: General Information*.

Where to Get z/VM Information

z/VM product information is available from the following sources:

- z/VM V6R2 Information Center (publib.boulder.ibm.com/infocenter/zvm/v6r2/)
- IBM: z/VM Internet Library (www.ibm.com/vm/library/)
- IBM Publications Center (www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss)
- *IBM Online Library: z/VM Collection*, SK5T-7054

z/VM Base Library

Overview

- *z/VM: General Information*, GC24-6193
- *z/VM: Glossary*, GC24-6195
- *z/VM: License Information*, GC24-6200

Installation, Migration, and Service

- *z/VM: Installation Guide*, GC24-6246
- *z/VM: Migration Guide*, GC24-6201
- *z/VM: Service Guide*, GC24-6247
- *z/VM: VMSES/E Introduction and Reference*, GC24-6243

Planning and Administration

- *z/VM: CMS File Pool Planning, Administration, and Operation*, SC24-6167
- *z/VM: CMS Planning and Administration*, SC24-6171
- *z/VM: Connectivity*, SC24-6174
- *z/VM: CP Planning and Administration*, SC24-6178
- *z/VM: Getting Started with Linux on System z*, SC24-6194
- *z/VM: Group Control System*, SC24-6196
- *z/VM: I/O Configuration*, SC24-6198
- *z/VM: Running Guest Operating Systems*, SC24-6228
- *z/VM: Saved Segments Planning and Administration*, SC24-6229
- *z/VM: Secure Configuration Guide*, SC24-6230
- *z/VM: TCP/IP LDAP Administration Guide*, SC24-6236
- *z/VM: TCP/IP Planning and Customization*, SC24-6238
- *z/OS and z/VM: Hardware Configuration Manager User's Guide*, SC33-7989

Customization and Tuning

- *z/VM: CP Exit Customization*, SC24-6176
- *z/VM: Performance*, SC24-6208

Operation and Use

- *z/VM: CMS Commands and Utilities Reference*, SC24-6166
- *z/VM: CMS Pipelines Reference*, SC24-6169
- *z/VM: CMS Pipelines User's Guide*, SC24-6170
- *z/VM: CMS Primer*, SC24-6172
- *z/VM: CMS User's Guide*, SC24-6173
- *z/VM: CP Commands and Utilities Reference*, SC24-6175
- *z/VM: System Operation*, SC24-6233
- *z/VM: TCP/IP User's Guide*, SC24-6240
- *z/VM: Virtual Machine Operation*, SC24-6241
- *z/VM: XEDIT Commands and Macros Reference*, SC24-6244
- *z/VM: XEDIT User's Guide*, SC24-6245
- *CMS/TSO Pipelines: Author's Edition*, SL26-0018

Application Programming

- *z/VM: CMS Application Development Guide*, SC24-6162
- *z/VM: CMS Application Development Guide for Assembler*, SC24-6163
- *z/VM: CMS Application Multitasking*, SC24-6164
- *z/VM: CMS Callable Services Reference*, SC24-6165
- *z/VM: CMS Macros and Functions Reference*, SC24-6168
- *z/VM: CP Programming Services*, SC24-6179
- *z/VM: CPI Communications User's Guide*, SC24-6180

- *z/VM: Enterprise Systems Architecture/ Extended Configuration Principles of Operation*, SC24-6192
- *z/VM: Language Environment User's Guide*, SC24-6199
- *z/VM: OpenExtensions Advanced Application Programming Tools*, SC24-6202
- *z/VM: OpenExtensions Callable Services Reference*, SC24-6203
- *z/VM: OpenExtensions Commands Reference*, SC24-6204
- *z/VM: OpenExtensions POSIX Conformance Document*, GC24-6205
- *z/VM: OpenExtensions User's Guide*, SC24-6206
- *z/VM: Program Management Binder for CMS*, SC24-6211
- *z/VM: Reusable Server Kernel Programmer's Guide and Reference*, SC24-6220
- *z/VM: REXX/VM Reference*, SC24-6221
- *z/VM: REXX/VM User's Guide*, SC24-6222
- *z/VM: Systems Management Application Programming*, SC24-6234
- *z/VM: TCP/IP Programmer's Reference*, SC24-6239
- *Common Programming Interface Communications Reference*, SC26-4399
- *Common Programming Interface Resource Recovery Reference*, SC31-6821
- *z/OS: IBM Tivoli Directory Server Plug-in Reference for z/OS*, SA76-0148
- *z/OS: Language Environment Concepts Guide*, SA22-7567
- *z/OS: Language Environment Debugging Guide*, GA22-7560
- *z/OS: Language Environment Programming Guide*, SA22-7561
- *z/OS: Language Environment Programming Reference*, SA22-7562
- *z/OS: Language Environment Run-Time Messages*, SA22-7566
- *z/OS: Language Environment Writing Interlanguage Communication Applications*, SA22-7563
- *z/OS MVS Program Management: Advanced Facilities*, SA22-7644
- *z/OS MVS Program Management: User's Guide and Reference*, SA22-7643

Diagnosis

- *z/VM: CMS and REXX/VM Messages and Codes*, GC24-6161
- *z/VM: CP Messages and Codes*, GC24-6177
- *z/VM: Diagnosis Guide*, GC24-6187
- *z/VM: Dump Viewing Facility*, GC24-6191
- *z/VM: Other Components Messages and Codes*, GC24-6207
- *z/VM: TCP/IP Diagnosis Guide*, GC24-6235
- *z/VM: TCP/IP Messages and Codes*, GC24-6237
- *z/VM: VM Dump Tool*, GC24-6242
- *z/OS and z/VM: Hardware Configuration Definition Messages*, SC33-7986

z/VM Facilities and Features

Data Facility Storage Management Subsystem for VM

- *z/VM: DFSMS/VM Customization*, SC24-6181
- *z/VM: DFSMS/VM Diagnosis Guide*, GC24-6182
- *z/VM: DFSMS/VM Messages and Codes*, GC24-6183
- *z/VM: DFSMS/VM Planning Guide*, SC24-6184
- *z/VM: DFSMS/VM Removable Media Services*, SC24-6185
- *z/VM: DFSMS/VM Storage Administration*, SC24-6186

Directory Maintenance Facility for z/VM

- *z/VM: Directory Maintenance Facility Commands Reference*, SC24-6188
- *z/VM: Directory Maintenance Facility Messages*, GC24-6189
- *z/VM: Directory Maintenance Facility Tailoring and Administration Guide*, SC24-6190

Open Systems Adapter/Support Facility

- *zEnterprise System, System z10, System z9 and eServer zSeries: Open Systems Adapter-Express Customer's Guide and Reference*, SA22-7935
- *System z9 and eServer zSeries 890 and 990: Open Systems Adapter-Express Integrated Console Controller User's Guide*, SA22-7990

- *System z: Open Systems Adapter-Express Integrated Console Controller 3215 Support*, SA23-2247
- *System z10: Open Systems Adapter-Express3 Integrated Console Controller Dual-Port User's Guide*, SA23-2266

Performance Toolkit for VM

- *z/VM: Performance Toolkit Guide*, SC24-6209
- *z/VM: Performance Toolkit Reference*, SC24-6210

RACF Security Server for z/VM

- *z/VM: RACF Security Server Auditor's Guide*, SC24-6212
- *z/VM: RACF Security Server Command Language Reference*, SC24-6213
- *z/VM: RACF Security Server Diagnosis Guide*, GC24-6214
- *z/VM: RACF Security Server General User's Guide*, SC24-6215
- *z/VM: RACF Security Server Macros and Interfaces*, SC24-6216
- *z/VM: RACF Security Server Messages and Codes*, GC24-6217
- *z/VM: RACF Security Server Security Administrator's Guide*, SC24-6218
- *z/VM: RACF Security Server System Programmer's Guide*, SC24-6219
- *z/VM: Security Server RACROUTE Macro Reference*, SC24-6231

Remote Spooling Communications Subsystem Networking for z/VM

- *z/VM: RSCS Networking Diagnosis*, GC24-6223
- *z/VM: RSCS Networking Exit Customization*, SC24-6224
- *z/VM: RSCS Networking Messages and Codes*, GC24-6225
- *z/VM: RSCS Networking Operation and Use*, SC24-6226
- *z/VM: RSCS Networking Planning and Configuration*, SC24-6227
- *Network Job Entry: Formats and Protocols*, SA22-7539

Prerequisite Products

Device Support Facilities

- *Device Support Facilities: User's Guide and Reference*, GC35-0033

Environmental Record Editing and Printing Program

- *Environmental Record Editing and Printing Program (EREP): Reference*, GC35-0152
- *Environmental Record Editing and Printing Program (EREP): User's Guide*, GC35-0151

Index

Special characters

A

C

D

E

H

I

M

O

P

R

S

T

U

V

W

Product Number: 5741-A07

Printed in USA

SC24-6190-02