CA VM:Director™ for z/VM

Administration Guide

Release 3.1, Second Edition

7/21/2014

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Mainframe VM Product Manager
- CA VM:Account™
- CA VM:Director™ for z/VM
- CA VM:Operator™
- CA VM:Secure™ for z/VM

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Documentation Changes

The following documentation updates have been made after the last release of this documentation.

Note: In PDF format, page references identify the first page of the topic in which a change is made. The actual change can appear on a later page.

- Added the Chapter 6: Managing CA VM:Director in a Single System Image Complex (see page 89) section.
- Added Chapter 24: Upgrade in Place Installation (see page 273) about upgrading z/VM 6.2.0 to z/VM 6.3.0.

Contents

Chapter 1: Introduction 15

Directory and Disk Space Management 15
General System Security 16
Interface with CA VM:Account 17
Hardware Requirements 17
Software Requirements 17
Starting and Stopping CA VM:Director 18
- Starting CA VM:Director Automatically 18
- Starting CA VM:Director Manually 18
- Changing Startup Options 19
- Copyright Screen Display 20
- Stopping CA VM:Director 20

Recommendations for Releasing to Users 22
Moving Through Screens and Menus 23
- Help with Screens and Menus 23

CA VM:Director Commands 24
- Help with CA VM:Director Commands 25

Format Conventions for Code Syntax 26
CA VM:Director Messages 31
- Help with CA VM:Director Messages 33

Changes in CA VM:Director 34

Chapter 2: Features and Concepts 35

CA VM:Director Directory Database 35
Configuration Files 36
VMDIRECT MANAGERS File 37
VMDIRECT GLOBALS File 37
VMDIRECT POSIX File 37
DASD Management 38
Directory Management 38
- Skeleton Files 38
- Directory Profiles 39

Authorizations 39
- System Administrator 40
- Directory Manager 40
- SFS Administrator 40

SFS Manager 41
Auditing in CA VM:Director 41
Servant Facility 41
VM Single System Image 42

Chapter 3: Directory Managers 43

Adding a Directory Manager 43
Example 45
Changing Directory Manager Authorizations 45
Removing a Directory Manager 46
Checking for Directory Manager Authorization and Reassigning User IDs 47
Checking for and Revoking SFS Manager Authorization 47
Checking for and Revoking SFS Administrator Authorization 47
Removing the User ID as a Directory Manager 47
Adding a System Administrator 48
Removing a System Administrator 48

Chapter 4: Configuration Files and Records 49

AUTHORIZ CONFIG File 51
Sample CA VM:Director AUTHORIZ CONFIG File 52
DASD CONFIG File 52
DASD CONFIG File Records 53
Sample DASD CONFIG File 54
Effects of Changing Records in the DASD CONFIG File 55
CA VM:Director PRODUCT CONFIG File 56
CA VM:Director PRODUCT CONFIG File Records 57
Sample CA VM:Director PRODUCT CONFIG File 58
CA VM:Director SECURITY CONFIG File 59
CA VM:Director SECURITY CONFIG File Records 59
Sample CA VM:Director SECURITY CONFIG File 59

Chapter 5: Managing CA VM:Director 61

Making DRCT Volume Changes 61
Changing the Size of the CA VM:Director DRCT and AUDT Minidisks 62
Enlarging the CP Object Directory (Moving It to a Different DASD Volume) 64
Step 1: Calculate the Number of Pages That CA VM:Director Needs to Initialize the Directory 65
Step 2: Find Enough Free Space on a CP–owned Volume for the Entire Directory 66
Step 3: Allocate a Placeholder Minidisk in This Space (Optional) 67
Step 4: Change the DASD CONFIG File So That It Lists the Correct Extents for the CP Object Directory 67
Step 5: Create an IGNORED Minidisk for CA VM:Director to Link to, If Necessary 68

Step 6: Back Up the Current CP Directory 68
Step 7: Format and Allocate the New Extent 69
Step 8: Shut Down CA VM:Director 70
Step 9: Restart CA VM:Director 71
Step 10: Change the Allocation Bit Map on the Old Extent and Re–IPL z/VM 71
Enlarging the CP Object Directory (Keeping It on the Same DASD Volume) 71
Step 1: Calculate the Number of Pages That CA VM:Director Needs to Initialize the Directory 72
Step 2: Find Enough Free Space on the Current CP–Owned Volume for the Entire Directory 73
Step 3: Update the Placeholder Minidisk in This Space, If Necessary 74
Step 4: Change the DASD CONFIG File So That It Lists the Correct Extents for the CP Object Directory 75
Step 5: Back Up the Current CP Directory 75
Step 6: Shut Down CA VM:Director 75
Step 7: Format and Allocate the New Extent 76
Step 8: Restart CA VM:Director 77
Backing Up CA VM:Director 77
CA VM:Director Minidisks and Backup Requirements 78
Backing Up the Source Directory (DRCT Minidisk) 78
Backing Up Other CA VM:Director Minidisks 80
Restoring from the BKUP Minidisk 82
Reconfiguring CA VM:Director 82
Servant Facility 84
Setting Up the Servant Facility 85
Disabling the Servant Facility 87
Enabling the Servant Facility 87
Querying the Status of the Servant Facility 87
Assigning Account Numbers to User IDs 88

Chapter 6: Managing CA VM:Director in a Single System Image Complex 89

Overview 90
Distributed Server Definition 91
Command Processing in an SSI Complex 91
Directory Entry Processing 91
IDENTITY Entry Processing 92
SUBCONFIG Entry Processing 92
Configuration Statements in an SSI Complex 93
ACCESS 94
RESID 94
DIRECT 94
SSINODE 95
Disk Space Management in an SSI Complex 96
Servant Facility in an SSI Complex 97

SSI Installation Considerations 97
Converting non-SSI to SSI 98
Installing on an SSI Complex 99
SSI Miscellaneous Considerations 99
Journaling Additions 100
Member Specific Command Processing 100
Recovery After an Outage 101

Chapter 7: Authorizations 105

Granting Authorization to a Command or Utility 106
Granting Authorization to Parts of a Command 107
Waiving Password Requirements for Commands 107
Granting Authorization to Create User IDs 108
Granting Authorizations to Use the MAINT MANAGE Command 108
Granting Authorizations to Use the GRANT AUTHORITY and REVOKE AUTHORITY Commands 110
Using Predefined Variable Lists 111
Specifying a File Space 111
Granting Authorizations to Use Commands on Only Some User IDs 113
Working with GRANT AUTHORITY and REVOKE AUTHORITY Authorizations 113
Restricting Authorizations 116
User ID Lists and Authority Lists 117
User ID Lists 118
Authority Lists 118
Using a List Name in Another List 119
Predefined Variable Lists 119
Creating Authorizations with User ID Lists and Authority Lists 120
Creating Authorizations to Use Commands on Only Some Groups of User IDs 121
Pattern Matching in Authorizations 122
Querying Authorizations 123
Special Authorization Queries 124
Querying Authorizations Given to Lists of User IDs or for Lists of Commands 124
Sample Authorizations 124
Authorization for a System Administrator 125
Authorizations for Directory Managers 125
Authorization for a SFS Administrator 128
Authorization for SFS Managers 128
Authorizations for General Users 128
Authorizations for Command Processing 129
Authorization to Use CA VM:Director Commands and Utilities 131

Chapter 8: Managing Directory Entries 143

Creating Directory Entries 143
Creating Directory Entries Through a Programming Application 145
Creating a User ID Based on Another User ID 145
Creating a Directory Entry by Answering Screen Prompts 146
Creating a User ID by Filling in the Blanks on a Screen 147
Creating Several Directory Entries at the Same Time 148
Allowing Directory Managers to Create Directory Entries 148
Checking User ID Password Expiration 149
Managing Inactive Directory Entries 150
Expiring a Password Automatically 150
Querying Inactive User IDs 151
Disabling User IDs 152
Re-enabling Disabled User IDs 152
Putting User IDs on Hold 153
Reactivating User IDs 154
Removing Directory Entries 155
Removing Directory Entries by an Interactive Command 156
Removing Directory Entries Through a Programming Application 156
Removing Directory Entries by Filling in the Blanks on a Screen 157
Removing Several Directory Entries at the Same Time 157

Chapter 9: Directory Profiles 159

Creating a Directory Profile 159
Using an Existing Directory Profile as a Prototype 160
Editing a Directory Profile 160
Deleting a Directory Profile 160
Including a Directory Profile in a Directory Entry 161
Editing a Directory Entry That Includes a Directory Profile 161
Replacing a Directory Profile in a Directory Entry 162
Removing a Directory Profile from a Directory Entry 163

Chapter 10: Skeleton Files 165

Working with Skeleton Files 165
Creating Skeleton Files 168
Changing a Skeleton File 168
Deleting a Skeleton File 169
Allowing Directory Managers to Use a Skeleton File 169
Preventing Directory Managers from Using a Skeleton File 170

Chapter 11: Managing Space 171

Creating Minidisks 171
Virtual Address Restrictions 172
Creating Minidisks Using Programming Applications 172
Creating Minidisks by Answering Screen Prompts 173
Creating Minidisks by Filling in the Blanks on a Screen 173
Duplicating a Minidisk 173
Analyzing Minidisks 174
Moving Minidisks or Changing Minidisk Information 176
Moving Minidisks or Changing Minidisk Information Using Programming Applications 177
Moving Minidisks or Changing Minidisk Information by Answering Screen Prompts 177
Moving Minidisks or Changing Minidisk Information by Filling in Blanks on a Screen 178
Transferring a Minidisk from One User ID to Another 178
Protecting Minidisks from Being Scratched 179
Allocating DASD 179
Subpool Categories 179
Adding DASD to the CA VM:Director Configuration 180
Mapping a DASD Volume 181
Defragmenting a DASD Volume 181
Deleting Minidisks 181

Chapter 12: SFS Overview 183

SFS Administrators 184
SFS Managers 184
Enrollment Limits 185
Default Enrollment Values 186
CA VM:Director Configuration for SFS User Administration 188
Automatic Configuration 188
Subsequent Configuration Changes 192
Resynchronizing the CA VM:Director Configuration of SFS 192
Allocation of Space in User Storage Groups 195
Interaction Between an SFS Manager's Allocation Limit and a User Storage Group's Allocation Limit 196

Chapter 13: Configuring CA VM:Director for SFS 199

Before You Begin 199
Describing SFS to CA VM:Director 200
Adding a File Pool and User Storage Groups to the CA VM:Director Configuration 200
Changing a User Storage Group in a Known File Pool 203
Deleting a User Storage Group from the CA VM:Director Configuration 204
Removing a File Pool from the CA VM:Director Configuration 205

Querying the Status of File Pools 207
Resynchronizing the CA VM:Director Configuration of SFS 207
Defining SFS Administrators and Managers 208
Adding a New SFS Administrator 208
Removing an SFS Administrator 208
Adding a New SFS Manager 209
Reviewing or Changing an SFS Manager's Authority 212
Deleting File Pools from a Manager's Authority 213
Changing a Manager's Authority for a File Pool 213
Deleting an SFS Manager 215
Granting and Revoking Access to Files Spaces 216
Preparing to Use GRANT AUTHORITY and REVOKE AUTHORITY Commands 216
Migrating Minidisk Files to SFS 217

Chapter 14: Improving CA VM:Director Performance 219

Service Virtual Machine Performance 219
Performance with SFS 220

Chapter 15: Macros 221

Tracing Macros 221

Chapter 16: Security Features 223

Verifying Passwords with Another Security System 223
Verifying Directory Link Statements 223
Verifying Minidisk Link Modes and Passwords 224
Verifying Changes to Logon Passwords 224

Chapter 17: VMDIRECT MANAGERS File Reference 225

VMDIRECT MANAGERS File Record 225
DEFSIZE Record 228
Definitions 228
Description 229
DEVTYPE Record 229
Definitions 229
Description 230
IDLIMIT Record 230
Definitions 230
Description 230
MANAGER Record 231

Definitions 231
Description 231
SKELETON Record 232
Definitions 232
Description 232

Chapter 18: Special Comment Reference 233

Summary of CA VM:Director Special Comments 233
*AC= [Account Number] 235
*ED= [Edit] 235
*END= [End] 236
*ENROLL= [Enroll in SFS] 236
*EX= [Expire] 237
*FP= [File Pool] 237
*PW= [Password Change Date] 238
*SP= [Sample Profile] 238

Chapter 19: VMDIRECT GLOBALS File Reference 241

VMDIRECT GLOBALS File 241
Changing the Default Machine Mode 242

Chapter 20: POSIX Support in CA VM:Director 243

Maintaining POSIX Group Names and Group IDs Using CA VM:Director 243
POSIX Information in a Directory Entry or Profile 244
Querying and Changing POSIX Information 244
VMDIRECT POSIX File 245

Chapter 21: Systems Management API Support 247

Configure the CA VM:Director Server Environment 247
Install CA VM:Director API Components in the SMAPI Server 249
Files Provided by CA VM:Director for the SMAPI Server 250
Understanding a Directory Manager Specific Return Code (596) from the Socket SMAPI Server 250
CA VM:Director Command Authorization 251

Chapter 22: ESM Support (CA VM:Director only) 253

ESM Command Authorization 253
Creating Command Authorization Rules 254
Granting Authorization to General Users 255
Granting Authorization to Directory Managers 256

RACROUTE Resource Names 257
Authorizations for Command Processing 265
ESM Configuration Statement 265
Security Impact 266
Setting Up User Exits for Co-Existence of CA VM:Director with an ESM 266
Suppressing CP Commands Using the PASSWORD User Exit 266
Synchronizing ESM Passwords with CP Directory Passwords Using the PASSCHNG User Exit 267
Verifying the Identity of the Command Originator Using the CHKPASS User Exit 267
Enabling CA VM:Director Access to ESM-Controlled Resources 267
Enabling End-User Access to ESM-Controlled CA VM:Director Commands 269

Chapter 23: Upgrade in Place Installation 273

Upgrading z/VM 6.2.0 to z/VM 6.3.0 273

Index 277

Chapter 1: Introduction

This document contains information for CA VM:Director for z/VM (CA VM:Director).

CA VM:Director is an efficient and comprehensive directory and DASD management system for z/VM. It does the following:

- Significantly reduces the time required to manage the resources defined in the z/VM directory
- Prevents costly errors in DASD mapping

With CA VM:Director, you can delegate aspects of directory management to your users without jeopardizing system control. The product manages the z/VM directory by distributing the burden of user ID and DASD management from a single system administrator to several directory managers. They also allow users to make some directory changes for their own virtual machines.

Note: Due to documentation reuse, some passages are marked "(CA VM:Secure only)" or "(CA VM:Director only)".

Directory and Disk Space Management

CA VM:Director provides the following features for managing minidisks and directories:

DASD Management

CA VM:Director offers several commands and user exits to help manage DASD. You can easily accomplish common DASD management tasks, such as mapping volumes, defragmenting volumes, migrating volumes, and protecting minidisks from being scratched accidentally.

Shared File Space Management

CA VM:Director offers several commands and special comments to help manage shared file space. You can easily identify to CA VM:Director the file pools and user storage groups you want it to manage; specify which directory managers can control allocations of shared file space; and then accomplish common SFS management tasks, such as enrolling user IDs in SFS, allocating space to user IDs, and migrating data from minidisks to SFS.

Auditing

CA VM:Director provides auditing for all its functions. Reports display all directory maintenance activity. You can tailor these reports to display information in a format that suits your needs or that includes only the kinds of information you require.

Configurable Authorizations

CA VM:Director lets you authorize user IDs to use commands or parts of commands, for instance, a command with only one or two of its many options. You can also group user IDs or commands into lists so that you can authorize many users or authorize user IDs to many commands with a single authorization record.

Disaster Recovery

CA VM:Director provides three utilities for disaster recovery of your directory from disk.

Directory Profiles

You can easily change directory profiles with one CA VM:Director command. You can also expand directory entries that contain directory profiles and edit the working directory entry.

General System Security

CA VM:Director provides basic security features to sites that are not running the CA VM:Secure Rules Facility. These features include:

Password verification with other security systems

CA VM:Director can check the source directory to verify logon passwords that users enter while executing CA VM:Director commands. If the password is managed by a separate External Security Manager product like RACF, then the command processor can call the CHKPASS User Exit, which could ask the ESM to verify the password.

Directory link statement verification

CA VM:Director can verify a directory link to a virtual machine. You can also prevent users from defining directory links to sensitive or privileged user IDs, such as MAINT or OPERATOR.

Minidisk link mode and password verification

CA VM:Director can secure sensitive minidisks from harm by verifying changes made to a minidisk's access mode and link passwords.

Logon password checking

CA VM:Director lets you define standards for how many characters a password can have and what those characters can be. The product can also force users to change their passwords on a regular schedule and prevent reuse of recently used passwords.

Additionally, CA VM:Secure provides the following:

Rules Facility (CA VM:Secure only)

The CA VM:Secure Rules Facility provides complete rule-based security control. For more information, see the *Rules Facility Guide*.

Interface with CA VM:Account

If you use the CA resource accounting product, CA VM:Account, CA VM:Director performs the following accounting functions:

- Passing all account number changes to CA VM:Account for validation. CA VM:Director prevents users from charging resources consumed to invalid account numbers.
- When user IDs exceed the CA VM:Account budget limits, CA VM:Director can change the logon password to NOLOG in the CP source and object directory.

Note: For more information about how CA VM:Director works with CA VM:Account to validate account number changes, see the *CA Mainframe VM Product Manager Product Interface Guide*. For more information about how account numbers work with CA VM:Director, see Assigning Account Numbers to User IDs *(see page 88).*

Hardware Requirements

To access CA VM:Director function screens, you must use a display or full-screen (3270-type) terminal with a minimum of 24 lines. If you have a line-mode terminal, you must use line-mode commands to access CA VM:Director.

CA VM:Director reserves some virtual addresses for devices and the CA VM:Director service virtual machine.

Note: For more information about the reserved virtual addresses, see Virtual Address Restrictions (see page 172).

Software Requirements

CA products support z/VM releases that are supported by IBM.

(CA VM:Secure only) If you are using the CA VM:Secure Rules Facility, you must regenerate your CP nucleus every time you upgrade the CP component of CA VM:Secure to the next release, or apply service to the component.

You normally change the allocation area on the online directory volume when CA VM:Director is not active. When the product server is running, you can use the CPFMTXA product command to change the allocation area for the online directory volume. See the CPFMTXA command description in the *Reference Guide*.

The CA VM:Director ASSIGN and CHANGE commands use the CMS FILEPOOL EXEC to rename an individual file space within a file pool. Ensure that CA VM:Director has access to the MAINT 193 minidisk on which the FILEPOOL EXEC resides.

Starting and Stopping CA VM:Director

You can start CA VM:Director automatically at each system initialization or manually after your system is running.

Starting CA VM:Director Automatically

To have CA VM:Director start automatically when you initialize your z/VM system, include an XAUTOLOG command for the CA VM:Director server virtual machine in the PROFILE EXEC of your AUTOLOG1 user ID.

Note: For more information about installing and starting CA VM:Director, see the *Installation Guide*.

Starting CA VM:Director Manually

Follow these steps to start CA VM:Director manually (after your system is IPLed and is already running):

1. Enter the following on the CA VM:Director server console:

   ```
   IPL CMS
   ```

   You will see a prompt asking if you want to start CA VM:Director.

2. Respond to the prompt by entering **yes**.

   When CA VM:Director completes initialization, with or without prompting of Steps 1 and 2, it displays the following message:

   ```
   CA VM:Director INITIALIZATION COMPLETE ON mm/dd/yy
   ```

3. Disconnect CA VM:Director:

   ```
   #cp disconn
   ```

Changing Startup Options

Startup options control how CA VM:Director operates. You can change the options at any time. The CA VM:Director service virtual machine implements the changes when it next gets autologged.

To change startup options, use the VMSERVER command to update the definition in the VMSERVER NAMES file, as follows:

1. Log on to VMANAGER.
2. Issue this command:

   VMSERVER *servername*

   servername

   Specifies the name of the product server virtual machine.
3. Add one or more options to the Server Startup Command tag, in any order. The options are listed in the following section.
4. File the changes with PF6. Press PF3 to exit the VMSERVER command.

Note: For more information about changing startup options, see the *CA Mainframe VM Product Manager Reference Guide*.

CA VM:Director Startup Options

AUDINV

Includes invalid passwords in the audit records. On CA VM:Secure, this option is not supported if you are using long password phrases.

ALTVOL *vaddr volume*

Writes the CP object directory on an alternate volume. The variables *vaddr* and *volume* override the values specified on the DIRECT record in the PRODUCT CONFIG file and cause the CP object directory to be built on the specified device. CA VM:Director terminate operation after initializing the directory on the alternate volume.

SAVEDIR

Causes CA VM:Director to do a source start and save the directory entries back to the product DRCT minidisk to reflect any changes to the entries. Use this option when a DEVTYPE record in the DASD CONFIG file has changed and you want to have the source directory entries reflect what is being placed in the object directory.

Note: For information about the DEVTYPE record, see DEVTYPE Record in the *Reference Guide*

SOURCE

Uses the source directory database to rewrite the CP object directory at startup, even if no changes are detected in the source directory database. If you do not add the SOURCE option to the Server Startup Command tag, you can still use the option by entering SOURCE when the prompt appears.

Copyright Screen Display

A copyright screen appears the first-time you use CA VM:Director in any one logon session. The screen remains on your terminal screen for 15 seconds or until you press Enter.

When you have seen and cleared the copyright screen, you can prevent it from displaying again by renaming the VMDIRECT COPYRGHT file located on the public minidisk.

Stopping CA VM:Director

There are three ways to stop CA VM:Director:

To stop CA VM:Director this way:	Use this command:
Abnormally	ABEND

To stop CA VM:Director this way:	Use this command:
After active processes complete	END
Immediately, without waiting for active processes to complete	END FORCE

- As with all CA VM:Director commands, you must prefix commands with VMDIRECT if you issue them from a user ID other than the CA VM:Director service virtual machine.

Checking CA VM:Director Activity Before Stopping

Before you stop CA VM:Director, check who is using the system by entering the QPCB command.

```
vmdirect qpcb
```

This command lists the currently active processes and the commands they are executing.

Occasionally a user ID has an active process that prevents either a graceful shutdown of CA VM:Director, or a shutdown within a reasonable amount of time. When CA VM:Director takes longer to stop than you think is reasonable after you enter the END command, check which user ID has a lock on an object. Object locks indicate that something is still processing.

You can check for activity by entering the QLOCK command.

```
vmdirect qlock
```

This command lists all commands that are holding locks, the kinds of locks they are holding, and the objects on which they are holding locks. To correlate these commands with the user IDs that are running them, enter the QPCB command (see above) and match the commands with the currently active processes they are executing.

Effects of Forcing an Immediate Stop

When you force an immediate shutdown of CA VM:Director without waiting for current processes to complete, you may cause errors in CA VM:Director. For example, if a process that is modifying a minidisk definition is forced to terminate without completing, that minidisk may be incorrectly mapped. If this occurs, you can clear up these minidisks by reinitializing CA VM:Director with the SOURCE startup option.

Effects of Stopping CA VM:Director Abnormally

If you use the ABEND command to abnormally end CA VM:Director, it terminates with an ABN001 abend. Following the abend, control returns to CMS with a return code of 200. CA VM:Director produces a dump according to your specifications on the DUMP record in the PRODUCT CONFIG file.

CA VM:Director abends on its own if it encounters a severe operational error. Technical Support can often debug errors in the CA VM:Director system by analyzing a dump from your product system. In some cases, however, an error occurs that is not severe enough to trigger an automatic abnormal termination, but is severe enough that a dump may be necessary to analyze the problem. In these cases, Technical Support may ask you to create a dump by terminating the product abnormally using the ABEND command.

Do not use this ABEND command to terminate to create a dump unless a critically unusual situation occurs and you need a dump of the CA VM:Director system. For normal termination of CA VM:Director operation, use the END command.

Recommendations for Releasing to Users

As a system administrator, you are responsible for designating other system administrators (although this is not recommended) and directory managers and defining directory managers' and users' authorizations.

- Designate the user ID VMANAGER (the user ID under which you will be performing system administrator work) as the system administrator. You can do this by giving VMANAGER authorization for all commands, utilities, and user exits in the AUTHORIZ CONFIG file with this record:

  ```
  GRANT * OVER *ALL TO VMANAGER
  ```

  You will be the technical contact between your site and CA.

- Depending on the size and structure of your organization, designate one or more directory managers. CA VM:Director allows you to define a directory manager for each group of users on the system.

  Directory managers can perform routine functions such as defining minidisks, authorizing resource sharing, and managing disk space. They must be familiar with skeleton files and subpools and with the user IDs they manage. Inform the directory managers to contact you when problems arise.

  Provide directory managers with the *Reference Guide*, *Directory Manager Guide*, *Messages Reference Guide*, and parts of the *Administration Guide*.

- If you are using SFS, designate SFS managers. SFS managers must be directory managers whose user IDs will be using SFS. Any directory manager that manages user IDs that need SFS must also be an SFS manager.

  SFS managers need the same documentation as directory managers; they need no additional documentation.

- Provide general users with authorizations to the CA VM:Director commands they will use. Be sure they know the names and user IDs of their directory managers.

Notes:

- (CA VM:Secure only) If the CA VM:Secure Rules Facility is installed, provide Rules Facility users with the *Rules Facility Guide.*
- For more information about creating authorizations, see Authorizations (see page 105).

Moving Through Screens and Menus

The following commands bring up either screens in which you fill in fields or menus from which you select options:

- CONFIG
- MANAGE
- ULIST
- USER
- SUBCONFIG

All five commands require that you enter your logon password before you can use them. CA VM:Director prompts you for your password after you enter the command and before CA VM:Director displays its menus or entry screens. The logon password is required as another level of access control. It helps ensure that unauthorized people cannot use administration commands if they happen to gain access to your terminal.

To use these commands without having to enter your logon password, you must have NOPASS authorization in the AUTHORIZ CONFIG file.

Note: For information about the NOPASS authorization, see Waiving Password Requirements for Commands (see page 107). For more information about these commands and how they work, see the *Reference Guide*. Chapters throughout this guide also explain parts of these commands, because they cover the various tasks you can complete with CA VM:Director.

Help with Screens and Menus

You can get information about any currently displayed menu or screen by pressing PF1 (Help). This program function key displays a help screen that describes what you can do from the menu or screen.

When you are finished reading the help screen, press PF3 (End) to return to the original screen.

CA VM:Director Commands

You can enter CA VM:Director commands from two places:

- CMS, either directly or through CP SMSG
- CA VM:Director console

The format in which you must enter a CA VM:Director command depends on where you enter it:

- From CMS directly, you must precede CA VM:Director commands with the name of the service virtual machine:

  `VMDIRECT` *`command`* `[` *`parameters`*`]`

  All CA VM:Director commands can be entered from CMS.

- Through CP SMSG, you must precede CA VM:Director commands with #CP SMSG and the name of the CA VM:Director service virtual machine:

  `#CP SMSG VMDIRECT` *`command`* `[`*`parameters`*`]`

  Only some commands are enterable through CP SMSG, and you must have NOPASS authorization in the AUTHORIZ CONFIG file to issue these commands through CP SMSG. These commands are:

  - ABEND
  - CMS
  - END
  - EXPIRE
  - JOURNAL
  - MAP
  - NOLOG
  - QPCB
  - QUERY
  - RESET
  - TRACE

- From the CA VM:Director console**,** do not precede product commands with the name of the product service virtual machine:

  `command` `[`*`parameters`*`]`

Only some CA VM:Director commands can be entered at the product console, and only when the product console is running connected. These commands are:

- ABEND
- CMS
- END
- JOURNAL
- MAP
- QLOCK
- QPCB
- RESET

Important! Do not enter other commands from the CA VM:Director service virtual machine console. Further, never issue commands to other CA products from the CA VM:Director console. CA VM:Director operation will cease while that command executes, and this can cause your entire system to hang.

Help with CA VM:Director Commands

You can get help for selected CA VM:Director commands through the CMS HELP facility. To see an overview of CA VM:Director online help and a list of commands for which help is available, enter the following from CMS:

```
help vmdirect
```

To see the online help screen for one of the commands listed in the overview, move the cursor to that command and press PF1. When you are finished reading the help screen, press PF3 to return to the online help overview screen and PF3 again to exit the CMS HELP facility.

You can bypass the online help overview screen and get help for a command directly by entering **help vmdirect** and the name of the CA VM:Director command for which you need help. For example, to see the online help screen on the CA VM:Director QUERY command, enter the following from CMS:

```
help vmdirect query
```

Note: For more information about using the CMS HELP facility, see the IBM *CMS User's Guide* appropriate for your system.

Format Conventions for Code Syntax

This section describes the format and conventions used to document commands, utilities, and user exits. Each convention provides examples, describing how to use commands, how to use options, or how the system responds to user entries.

Note: The examples and instructions throughout this document use VMANAGER as the user ID for system administrator and maintenance functions. Also, this document uses VMDIRECT as the user ID for the service virtual machine (svm). These user IDs are the default values. If you use non-default user IDs for the system administrator or the svm, replace the default values throughout this document with the values that you use.

Command Abbreviations

When a command contains uppercase and lowercase letters, then the uppercase letters denote the shortest acceptable abbreviation that you can use to type the command. However, when a code item appears entirely in uppercase letters, you cannot abbreviate the item.

You can type the code item in uppercase letters, lowercase letters, or any combination.

Example:

```
CMDNAme
```

In this example, you can enter CMDNA, CMDNAM, or CMDNAME in any combination of uppercase and lowercase letters.

Continuation

The code syntax or code fragment definitions can continue from one line to the next line. The following examples describe code continuation:

Example 1:

```
A | B C | D
```

This code is equivalent to the following code:

```
A
| B C
| D
```

Example 2:

```
{choice1 | choice2 | choice99}
```

This code is equivalent to the following code:

```
 { choice1
 | choice2
 | choice99 }
```

Default Values

An underlined code item denotes the default value. The system uses the default value unless you override it. You can override the default value by coding an option from the available list.

Example:

[parm1 | *parm2* | *parm3*]

In this example, the code item *parm1* is the default value, and this is used by the system when you do not specify any of the options. However, you can code *parm1*, *parm2*, or *parm3.*

Keywords and Constants

A keyword name or constant always appears in uppercase letters. Code the keyword name or constant exactly as shown in the following example:

Example:

```
STOP {tracenumber | * | [USER] userid}
```

This example displays the USER keyword.

Optional Choices

Defines optional code items—denoted by square brackets around a code item.

Example:

```
CMDName [parm1]
```

In this example, you can choose *parm1* or no parameter at all. However, when two or more items are enclosed in square brackets and separated by vertical bar characters, all of them are optional.

Multiple Optional Choices

When two or more items are enclosed in square brackets and separated by vertical bar characters, all of them are optional.

Example:

In this example, you can choose *parm1, parm2, parm3,* or nothing at all.

```
[parm1 | parm2 | parm3]
```

Positional Parameters

Commands with positional parameters are identified by nested square brackets. Each positional parameter requires the specification of all previous positional parameters. The following example describes the positional parameter:

Example:

```
CMDName [posparm1 [posparm2 [posparm3]]]
```

In this example, *posparm3* can be specified only when *posparm1* and *posparm2* are also specified.

Repeatable Choices

A list of code items enclosed in square brackets and followed by an ellipsis means that you can select more than one item or, in some cases, repeat a single item.

Example 1:

```
[value1 | value2 | value3]...
```

In this example, you can choose a single value, more than one value, or none of the values.

Repetition

An ellipsis following a code item means that the code item can be repeated.

Example 2:

```
Repeat…
```

Example 3:

```
{Operand1 | Operand2}...
```

In this example, you must choose at least one operand and you can choose both operands. If one of the operands is a variable, you may also be able to repeat that operand with different values.

Required Choices

You must select one item from a list of items when they are enclosed in curly braces. The items are separated by a vertical bar character.

Example:

```
CMDNAme {A | B | C}
```

In this example, your choice results in CMDNAME A, CMDNAME B, or CMDNAME C.

Special Symbols

The following list describes the meaning of the special symbols used in codes:

- {} (encloses a list of operands, one of which is required).
- [] (enclose an optional operand or operands).
- " " (enclose the name of a syntax fragment)
- underline (identifies a default value)
- | (separates alternative operands)
- . . . (Indicates that the preceding item or group can be repeated).

Symbols

The following list displays symbols. These symbols should be coded exactly as they appear in the code syntax.

- * (Asterisk)
- : (Colon)
- , (Comma)
- = (Equal Sign)
- — (Hyphen)
- ( )(Parentheses)
- . (Period)

Syntax Fragments

Some codes use fragments, when the code syntax is too lengthy. The fragment name appears between double quotes in the code syntax.

The expanded fragment appears in the syntax after all other parameters or at the bottom of the code syntax. A heading with the fragment name identifies the expanded fragment.

Example1:

```
CMDName "Parms"

Parms:

[ParmA] [ParmB] [ParmC]
```

In this example, the fragment is named "Parms", and the expanded fragment appears at the bottom of the code syntax.

Example 2:

```
CMDName [( "Options"]


Options:

  OptionA
| OptionB
| OptionC
| OptionD [OptionD1]
```

In this example, the fragment is named Options. The list of options is too long to fit on one line, so they are listed vertically. Note that the outermost brackets and the "(" symbol are in the main line, not in the vertical list. Also note that one option has its own optional parameter.

System Response

Uppercase characters represent system responses or prompts.

Example:

```
ENTER YOUR LOGON PASSWORD:
```

This example displays a system response.

User–Entered Commands or Records

User-entered commands are shown in bolded lowercase letters even though you can enter commands in either upper or lower case.

Example:

`vmsecure addentry writers tcom (noformat nowait`

This example shows what a user-entered command looks like.

In this example "writers tcom" is the file name and file type of the directory entry you are adding.

However, if the entry is a record, it appears in uppercase letters.

Example:

```
ACCESS DRCT 1B0 U
```

This example shows a configuration record that is entered by a user.

Variables

Lowercase items in italics denote variables.

Example:

CMDNAme *varname*

In this example, *varname* represents a variable that you must specify when you code the command CMDNAME.

CA VM:Director Messages

While using CA VM:Director, you receive messages that do one of the following:

- Inform you of the status and progress of the function or task

- Prompt you for information about your system
- Indicate whether an error occurred
- Provide instructions for resolving errors

Messages displayed in line mode or in the command execution log have a defined format. This format is described using the following sample message:

```
VMXCOM1091E COMMAND LENGTH CANNOT EXCEED 256 CHARACTERS
```

The first token in the message is named the message identifier. Use this identifier to find the cause and required actions for a specific message using the CMS HELP facility or the product *Message Reference Guide*. The remainder of the message is the message text that provides the appropriate information. The message header components provide additional information. For this message:

VMX

This three letter code Is a unique product identifier for each product.

COM

This three letter code identifies the routine or module that issued the message.

1091

This message number identifies each separate message issued by a product.

E

This severity code is a code to identify the type of message.

You can tell the type of message by the last letter of the identifier, which is its *severity code*. The severity code can be one of the following:

A

Action message.

You need to perform an action, for example, move the cursor or enter information.

E

Error message.

The function or task can continue normally if you correct the error, but it may end.

I

Informational message.

CA VM:Director initialization has completed or a task that the product is executing has completed. When an informational message appears immediately after a severe error message (severity code S), it tells you how to correct the problem the severe error message identified.

R

Response message.

A request for information has been presented. You need to respond to a CA VM:Director message.

S

Severe error message.

An error preventing the function or task from continuing has occurred. You are returned to CMS and must restart the procedure after correcting the error.

T

Terminal error message.

A serious internal error has occurred, preventing the CA VM:Director service virtual machine from continuing execution.

W

Warning message.

An abnormal condition has occurred. The function or task continues, although the results may be affected by the abnormal condition.

Help with CA VM:Director Messages

Note: For more information or help for a CA VM:Director message you receive, see the *Message Reference Guide* or use the CMS HELP facility for that message.

The *Message Reference Guide* lists all messages that CA VM:Director can generate, explains the possible causes of each message, and tells you how to respond to that message.

To get help for a CA VM:Director message, from CMS, enter **help**, followed by the 3-character product identifier, the message number, and its severity code.

For example, to get help for the message VMXCOM1091E, enter the following from CMS:

```
help vmx1091e
```

When you are finished reading the message help, press PF3 to exit the CMS HELP facility.

Note: For more information about using the CMS help facility, see the IBM *CMS User's Guide* appropriate for your system.

Changes in CA VM:Director

For information about new or changed features, see the *Release Notes.*

Chapter 2: Features and Concepts

This chapter summarizes the features and components in CA VM:Director and some of the concepts you must understand before you decide how to use or customize these components for your site.

CA VM:Director Directory Database

The CA VM:Director directory database consists of a set of files that reside on the CMS–formatted DRCT minidisk. These files contain directory source statements that are compatible with the CP directory statements. However, unlike the single card–image directory file format used in CP, CA VM:Director directory entries are kept in individual files. One or more of these directory entry files are combined to form a *virtual machine definition* which specifies the configuration and characteristics for a virtual machine or VM user ID. There are two types of *virtual machine definition*:

Single configuration

This type is formed from a USER directory entry, optionally combined with an included PROFILE directory entry.

Multiconfiguration

This type is formed from an IDENTITY directory entry, optionally combined with an included PROFILE directory entry, and merged with all associated SUBCONFIG directory entries.

In the CA VM:Director directory database, each USER, PROFILE, IDENTITY, or SUBCONFIG entry is contained in a separate CMS file. The entries are combined to form the definition for each user ID on the system. For USER and IDENTITY entries, the file name of an entry is the name of the user ID it defines. PROFILE entry files are named by the identifier on an INCLUDE statement used to reference the entry. SUBCONFIG entry files are named by the identifier on a BUILD statement used to reference the entry. The file type of each file is the name of the directory manager for that directory entry.

Another directory database minidisk, the BKUP minidisk, is allocated on a separate volume, and receives copies of the directory entry files whenever they are modified. The files on this disk serve as a backup and enable you to recover the current status of the CA VM:Director directory database in the event the DRCT minidisk is lost.

A third directory database minidisk, the HOLD minidisk, holds directory entry files of user IDs that have been temporarily removed from the CP directory. In this way, the actual disk space allocated for a held user's minidisks is retained in the system while the directory entry for the user ID is not. You can easily restore the user ID to its original state.

Note: For information about how the directory database is created, see VMXGNR Utility in the chapter "Utility Reference" in the *Reference Guide*.

Configuration Files

CA VM:Director uses configuration files to control the processing of many product–related functions. Each configuration file can include only specific records. The configuration files are:

AUTHORIZ CONFIG

Controls who is authorized to issue CA VM:Director commands.

DASD CONFIG

Maintains information about DASD controlled by CA VM:Director.

PRODUCT CONFIG

Controls how CA VM:Director operates.

SECURITY CONFIG

Controls password processing. For CA VM:Secure, this file also controls how the Rules Facility operates.

Most configuration records in these files can be changed while CA VM:Director is running, and their new values become effective as soon as you save the changes in the configuration file. This is referred to as *dynamic* reconfiguration.

Several configuration file records cannot be changed while CA VM:Director is running.

Note: For more information about these records, see Configuration Files and Records (see page 49).

You can update these configuration files directly through an XEDIT interface controlled by CA VM:Director.

Note: For more information about each of the configuration files and their associated records, see the chapter "Configuration File Reference" in the *Reference Guide*.

If you use CA VM:Director to manage SFS allocations, information about your SFS is stored in a configuration file that you do not edit directly. This file, SFS CONFIG, is created when you configure CA VM:Director for SFS, and CA VM:Director alone maintains it, based on the descriptions and enrollments you add to CA VM:Director. If you edit this file directly, you will invalidate the CA VM:Director functions for SFS management.

Note: For more information about configuration for SFS, see Configuring CA VM:Director for SFS (see page 199).

VMDIRECT MANAGERS File

The VMDIRECT MANAGERS file defines existing VM user IDs as directory managers to whom you can delegate directory management tasks. Each directory manager can perform routine directory maintenance, minidisk management, and user ID management tasks for a group of user IDs. These tasks include creating and deactivating user IDs and creating and changing passwords for user IDs. Typically, as system administrator, you will manage several directory managers, and the directory managers will manage the users.

Note: For more information about VMDIRECT MANAGERS file and the records it uses, see VMDIRECT MANAGERS File Reference (see page 225).

VMDIRECT GLOBALS File

The VMDIRECT GLOBALS file, which resides on the CA VM:Director DRCT minidisk, contains the CA VM:Director version of the GLOBALDEFS section in the CP source directory. Use this file, instead of the USER DIRECT file, to specify default settings that can apply to all users on the system.

Note: For more information about the VMDIRECT GLOBALS file, see VMDIRECT GLOBALS File Reference (see page 241).

VMDIRECT POSIX File

The VMDIRECT POSIX file, which resides on the CA VM:Director DRCT minidisk, contains the POSIX group definitions. Use this file, instead of the USER DIRECT file, to define POSIX groups.

Note: For more information about the VMDIRECT POSIX file and other POSIX-related information, see POSIX Support in CA VM:Director (see page 243).

DASD Management

Disk space allocation is controlled through SUBPOOL records in the DASD CONFIG file. SUBPOOL records associate certain allocation parameters with a name that is referenced during allocation. Subpools do not directly define the space to be allocated, but contain DASD extent definitions that the subpool can use.

The actual disk storage definition begins with a VOLUME record that specifies the disk's volume label and its device type. Each contiguous block of disk space on a volume that is to be allocated for user minidisks must be defined with an EXTENT record that follows the corresponding VOLUME record. Exact definitions indicate the beginning and ending cylinder or block address and the names of the subpools from which allocations can be made. The extents cannot overlap. You can also specify extents to be used by CP.

At system startup, CA VM:Director uses the records in the DASD CONFIG file to construct a disk storage layout. Each user definition is examined, and a complete storage map, defined at the minidisk level, is created in virtual memory. No minidisk, except those specifically listed in the DASD CONFIG file to be ignored, is allowed to overlap an extent definition or another minidisk.

Directory Management

When you create a new user ID, one or more directory entries are added to the CA VM:Director directory database. CA VM:Director creates these entries based on statements contained in a skeleton file and a directory profile, or an input file on an accessed minidisk.

Skeleton files make it easier to define directory entries with the same characteristics.

Skeleton Files

Skeleton files are prototype CP directory entries. Control statements in skeleton files specify the entry name and components of each virtual machine, among other things. They can also contain INCLUDE statements that effectively include the statements from a directory profile in the skeleton file.

You can use skeleton files as the basis for directory entries for new user IDs. In contrast to directory profiles, however, skeleton files are not merely referenced by directory entries; rather, they are copied into the new directory entries for user IDs you create by using skeleton files.

Note: For more information about using skeleton files, see Skeleton Files (see page 165).

Directory Profiles

In a z/VM system, you define each user with *virtual machine definitions* in the CP online directory. Control statements in the directory entries specify the user ID, password, and components of each user's virtual machine.

Directory profiles contain sets of frequently used control statements. Instead of entering the same control statements repeatedly in many directory entries, you can create one directory profile that contains these commonly used control statements. You then add an INCLUDE statement to each directory entry that needs to use the control statements in that directory profile.

Directory profiles offer several advantages over repeating the same control statements in numerous directory entries:

- Easier to define user IDs with the same characteristics
- Saves space in the CP online directory and the CA VM:Director DRCT disk
- Easier to maintain or change numerous directory entries dynamically and at the same time

Note: For information about using directory profiles with CA VM:Director, see Directory Profiles (see page 159). For more information about directory profiles in general, see the appropriate IBM *Planning and Administration* documentation.

Authorizations

Most CA VM:Director command and utility functions should be restricted for use by only system administrator and directory manager user IDs. Restricting their use means that only these user IDs can perform powerful CA VM:Director functions, although all users and all processes on your VM system benefit from them.

Most user ID and minidisk management functions should be restricted to the user IDs you make *system administrator* and *directory managers*. If you are using SFS, the SFS–related functions should be restricted to the user IDs you make *SFS administrators* and *SFS managers*.

System Administrator

As system administrator, you are responsible for designating other system administrators (although this is not recommended) and directory managers, and defining directory managers' and users' authorizations.

You can designate any existing VM user ID as a CA VM:Director system administrator by adding a GRANT record to the AUTHORIZ CONFIG file.

Note: For information about user authorizations and granting user authorizations, see Authorizations (see page 105). For more information about the GRANT record and how to use it, see GRANT Record in the chapter "Configuration File Reference" in the *Reference Guide*.

Directory Manager

CA VM:Director lets you define existing VM user IDs as directory managers to whom you can delegate directory management tasks. Each directory manager can perform routine directory maintenance, minidisk management, and user ID management tasks for a group of user IDs. Typically, as system administrator, you will be managing several directory managers, and the directory managers will manage the users.

SFS Administrator

SFS administrators in CA VM:Director perform three primary functions:

- Identify to CA VM:Director the SFS file pools and user storage groups on your system that it will manage
- Designate existing directory managers as SFS managers, and specify which file pools, storage groups, and default allocations those managers can use when they allocate SFS space under CA VM:Director
- Enroll any user ID in any SFS file pools under CA VM:Director control, giving that user ID any amount of file space in any storage group

You can designate any existing VM user ID as an SFS administrator in CA VM:Director by adding a GRANT SFSADMIN record to the AUTHORIZ CONFIG file for that user ID.

Note: For information about user authorizations and granting user authorizations, see Authorizations (see page 105). For more about the GRANT record and how to use it, see GRANT Record in the chapter "Configuration File Reference" in the *Reference Guide*.

SFS Manager

SFS managers in CA VM:Director are any directory managers that an SFS administrator has designated to also be SFS managers. SFS Managers can add, or enroll, any user ID that they can manage in SFS.

Any VM user ID that is a directory manager can be made an SFS manager. To designate a user ID as an SFS manager, you must be an SFS administrator (authority SFSADMIN); your status as a CA VM:Director system administrator is of no consequence.

The SFS administrator defines SFS managers through the SFS Managers Configuration Menu.

Auditing in CA VM:Director

CA VM:Director records audit information for most CA VM:Director commands and directory changes. CA VM:Director writes audit records to the VMDIRECT AUDIT file on the AUDT minidisk each time a user issues an audited command. The AUDIT user exit is called for each record. You can use it to select which audit records you want to be written to the AUDT minidisk.

When the AUDT minidisk fills, use the AUDITEXT command to move the audit records to your 191 minidisk and reinitialize the AUDT minidisk. You can then use the information in the audit records as input to the VMXSRB report program. VMXSRB generates a report of all audited system actions. To exclude certain types of audit records from the reports, you can use the SECURITY REPORTS user exit.

Note: For more information about the AUDITEXT command, VMXSRB report program, and the SECURITY REPORTS user exit, see the *Reference Guide*.

Servant Facility

The Servant Facility is an optional feature of CA VM:Director. It is a set of special worker virtual machines that help CA VM:Director perform long-running minidisk and SFS management CMS tasks. With the Servant Facility, directory managers can perform minidisk management functions disconnected from their terminals. This means that they can submit minidisk management requests to CA VM:Director and use their user IDs for other work while CA VM:Director, through the Servant Facility worker virtual machines, processes these requests.

You must use the Servant Facility if you use CA VM:Director for SFS administration and want to use the DELETE and CHANGE commands. With the Servant Facility, potentially long–running commands can execute on a CA VM:Director servant instead of tying up the CA VM:Director service virtual machine.

The Servant Facility is either enabled or disabled on your system, and is used by CA VM:Director whenever CA VM:Director needs extra worker machines to complete long–running tasks in a timely fashion. No user intervention, either by you or by any operators, is required to use the Servant Facility if it is enabled.

Note: For more information about the Servant Facility, see Servant Facility (see page 84).

VM Single System Image

The VM Single System Image facility is a capability which enables multiple VM systems to be defined and managed as if they were a single VM system. CA VM:Director supports this facility by supplying the following environmental characteristics:

- **The CP Object Directory on each member system is identical**. A USER virtual machine definition is exactly the same, no matter which member system that user runs on.
- **Directory Management Interfaces on each member system are identical**. You enter product commands exactly the same way, no matter which member you run on. All authorizations to product commands and services are identical for an administrator, directory manager, or general user.

In a Single System Image complex, CA VM:Director operates as a set of servers, with one server virtual machine running on each member system. A *master* server runs on one member to perform all the function of a non-SSI product server, including:

- Processing commands
- Updating Configuration and Source Directory Entries
- Compiling the Object Directory

An *agent* server runs on each other member node to perform this subset of the functions of the *master* server:

- Compiling the Object Directory

The *agent* also implements additional new function:

- Responds to synchronization requests from the *master* product server
- Converts to replace the *master* server if an outage situation occurs

Chapter 3: Directory Managers

Directory managers manage, create, change allocations for, and delete their assigned user IDs. They can also use a limited number of commands against their own user IDs.

Assuming they have been given the appropriate command authorizations, directory managers can use the ADDMDISK, CHGMDISK, DELMDISK, DUPENTRY, DUPMDISK, EDIT, GETENTRY, DELENTRY, and REPENTRY commands on the user IDs they manage and on their own user IDs. They can also use the MAINT and MANAGE commands on the user IDs they manage, but cannot use the MAINT and MANAGE commands to manage their own user IDs.

This section contains the following topics:

Adding a Directory Manager (see page 43)
Changing Directory Manager Authorizations (see page 45)
Removing a Directory Manager (see page 46)
Adding a System Administrator (see page 48)
Removing a System Administrator (see page 48)

Adding a Directory Manager

You can define any existing user ID, except the following, as a directory manager:

- The CA VM:Director service virtual machine user ID
- GLOBALS
- POOL
- POSIX
- PROFILE

If you try to make the CA VM:Director service virtual machine user ID a directory manager, CA VM:Director prevents the update or will not initialize. If you try to make the other user IDs directory managers, these names will conflict with similarly named objects in the CP object directory and the CA VM:Director directory database. This will also prevent CA VM:Director initialization or yield errors from the ADMIN MANAGERS command.

You must have system administrator authority to add a directory manager.

Adding a directory manager puts you into an XEDIT session on the VMDIRECT MANAGERS file. In this file, you must add one MANAGER record and one SKELETON record for each user ID you want to make a manager. You then add any of the other records that further define the scope of a directory manager's authority.

To add a directory manager

1. Enter the following:

   ```
   vmdirect admin managers
   ```

   This opens the VMDIRECT MANAGERS file for editing.

2. Add one of each of the following records for each user ID you want to make a directory manager.

   Enter the records in the following order:

Record	Include on this record
MANAGER	Directory manager user ID, the amount of disk space the manager can allocate, and the subpools from which the manager can allocate disk space
SKELETON	Skeleton files this directory manager can use to create user IDs

3. Optionally add one of each of the following records for each directory manager. Enter them in any order, but following the MANAGER record for that directory manager:

Record	Include on this record
DEFSIZE	Minidisk size that the directory manager assigns by default when defining minidisks
DEVTYPE	Device type that the directory manager assigns by default when allocating minidisks
IDLIMIT	Number of user IDs that the directory manager can manage

4. Save this XEDIT session and exit the VMDIRECT MANAGERS file.
5. Grant the new directory manager authorization to all of the commands necessary to perform the duties you assign. For commands that are typically used by directory managers, see the *Directory Manager's Guide*.

You can do this by adding GRANT records to the AUTHORIZ CONFIG file.

Note: For more information about tailoring user authorizations, see Authorizations (see page 105). For information about using the GRANT record and authorizing user IDs to use specific commands, see GRANT Record in the chapter "Configuration File Reference" in the *Reference Guide*.

Example

Add directory manager CARLAT and let her use only the skeleton file GENERAL when she creates user IDs. Allow CARLAT to allocate as much space as she wants from the subpool MAIN, and to assign the 3380 device type by default and 1M of disk space by default for user IDs she creates. (One megabyte of space on a 3380 device requires two cylinders; this will not be a problem because we are allowing CARLAT to allocate as much space as she wants, as long as it is available.) Finally, allow CARLAT to manage no more than 10 user IDs.

1. Enter the following:

   `vmdirect admin managers`

2. Add these five records to the file:

```
MANAGER CARLAT * MAIN
SKELETON CARLAT GENERAL
DEVTYPE CARLAT 3380
DEFSIZE CARLAT 1M
IDLIMIT CARLAT 10
```

3. Save this XEDIT session and exit the VMDIRECT MANAGERS file.

Changing Directory Manager Authorizations

You can change the scope of authority, such as which skeleton files a directory manager can use or how much space a directory manager can allocate, at any time. These changes are effective the next time the manager creates a user ID or changes any aspect of the user configuration that the VMDIRECT MANAGERS file controls; however, they have no effect on user IDs the directory manager has already created or is currently managing.

Before you change a directory manager's scope of authority, you may want to check the status of the user IDs that directory manager manages. For instance, before you reduce the amount of space that a directory manager can allocate, verify that directory manager actually has some unused space that you can take back.

Most of the information you will want to review about a directory manager is available through the QUERY command. For example:

Directory manager's specification	Command that lets you review this specification
Current space allocations and space left to allocate	QUERY ALLOC MANAGER *mgrid*
Current space allocations and space left to allocate for all directory managers	QUERY MANAGERS

Directory manager's specification	Command that lets you review this specification
Skeleton files available for creating user IDs	QUERY SKELETON MANAGER *mgrid*
Subpools available for creating user IDs	QUERY SUBPOOLS MANAGER *mgrid*

Note: For more information about these commands or about other information you can find for directory managers, see QUERY Command in the chapter "Command Reference" in the *Reference Guide*.

To change a directory manager's scope of authority

1. Enter the following:

   ```
   vmdirect admin managers
   ```

2. Find the MANAGER record for the manager you want to change. The other records for this manager must be immediately following.
3. Change any setting on any of these records, except for the name of the directory manager's user ID.
4. Save this XEDIT session and exit the VMDIRECT MANAGERS file.

Removing a Directory Manager

You can remove directory manager authority for a user ID if that manager user ID:

- Does not manage any user IDs (is not a directory manager)
- Does not have SFS manager authorization
- Does not have SFS administrator authorization

If a manager user ID has these authorizations, you must remove them in the order shown above.

Note: For more information about how to remove directory manager authority, see the following:

- Checking for Directory Manager Authorization and Reassigning User IDs (see page 47)
- Checking for and Revoking SFS Manager Authorization (see page 47)
- Checking for and Revoking SFS Administrator Authorization (see page 47)
- Removing the User ID as a Directory Manager (see page 47)

Checking for Directory Manager Authorization and Reassigning User IDs

To determine whether the user ID is managing user IDs, use the QUERY (MANAGER *mgrid* command. Replace *mgrid* with the manager's user ID.

If the user ID is managing user IDs, use the ASSIGN command to reassign those user IDs to other directory managers.

Checking for and Revoking SFS Manager Authorization

To determine whether the user ID is an SFS manager, use the QUERY MANAGER command. If N/A displays in the FILEPOOLS column for the manager user ID, this user ID is not an SFS Manager.

If the user ID is an SFS manager, use the VMDIRECT ADMIN SFSMGRS command to revoke the authorization. Enter selection **3** and the manager user ID. When prompted 'Do you wish to delete SFS Manager *userid* ', respond **Yes**.

Checking for and Revoking SFS Administrator Authorization

To determine whether the user ID is an SFS administrator, use the CONFIG AUTHORIZ command. Search for a record such as GRANT SFSADMIN TO *mgrid*, where *mgrid* is the manager's user ID. Delete this record.

While you are in the AUTHORIZ CONFIG file, remove this user ID's authorizations to all management commands this user ID no longer needs. Check specifically for authorizations to commands you authorized this user ID to use. You can remove the unnecessary authorizations by removing the appropriate GRANT records for this user ID.

Removing the User ID as a Directory Manager

When the user ID no longer has directory manager, SFS manager, or SFS administrator authorization, you can remove it as a directory manager.

To remove a user ID as a directory manager

1. Enter the following:

   ```
   vmdirect admin managers
   ```

2. Find the MANAGER record for the manager you want to remove. The other records for this manager must be immediately following.
3. Remove all the records for this directory manager.
4. Save this XEDIT session and exit the VMDIRECT MANAGERS file.
5. (CA VM:Secure only) Optional: Consider modifying the SECURITY CONFIG file to remove the manager user ID from any GROUP records on which it is specified. This action will revoke authorization for this user ID to create group rules and group log messages.

Adding a System Administrator

The system administrator is a directory manager that is authorized to use all commands. Make a system administrator by adding the following GRANT record to the AUTHORIZ CONFIG file, replacing *userid* with the name of the user ID to make a system administrator:

```
GRANT * TO userid
```

This record grants all authorities to the specified user ID. If you want to use the system administrator user ID created with CA VM:Director, VMANAGER, as your system administrator, grant all authority to VMANAGER.

Decide how you want to manage the system administrator user ID. To protect your system's security, you may want to create a special user ID that does nothing but manage the system administrator user ID. Consider the following methods to manage it:

- Create a directory manager user ID that will only manage the system administrator user ID. Allow this directory manager to manage only this one user ID.
- Give your MAINT user ID authorization to manage the system administrator user ID.

Be sure to authorize the user ID that needs to manage the system administrator user ID to use the appropriate directory manager commands. For commands that are typically used by directory managers, see the *Directory Manager Guid*e.

Removing a System Administrator

The procedure to remove a user ID as system administrator is to remove the GRANT * record for that user ID from the AUTHORIZ CONFIG file. Be sure that you give that user ID whatever authorizations it still needs, if any, to perform other CA VM:Director functions. You can do this with GRANT records, as you did for all other user IDs that need to use CA VM:Director commands or utilities.

Chapter 4: Configuration Files and Records

CA VM:Director uses configuration files to control the processing of most product–related functions. Each configuration file contains records that can only appear in that file. To edit the configuration files, use the CONFIG command. This chapter describes the CONFIG command and explains each configuration file and the records that can appear in them.

The majority of the records can be changed dynamically, and their new values take effect immediately after the file is saved. This is referred to as dynamic reconfiguration. To update certain configuration file records you must take down the CA VM:Director product server. In an SSI complex, the servers on every member must be taken down before changing certain records. The records are:

- ACCESS
- ALTERNATE
- AUTOPASS
- DIRECT
- RESID
- SSINODE

Changes to those records take effect when you restart the product.

You can set up or change the CA VM:Director configuration by entering this command:

```
vmdirect config
```

The CA VM:Director System Configuration Menu appears.

```
CONFIG                  System Configuration Menu                  VM:Director
--------------------------------------------------------------------------------
                    + - - - - - - - - - - - - - - - - +
                    |          Selection:             |
                    + - - - - - - - - - - - - - - - - +

 1 DASD Configuration         Maintain VM:Director controlled DASD definitions
                               (EXTENT, IGNORE, SUBPOOL, etc.)

 2 Product Configuration      Control how VM:Director should operate
                               (SERVANT, USEREXIT, etc.)

 3 Authorizations             Authorize who can issue VM:Director commands
                               (GRANT, LIST, WITHHOLD)

 4 Security Configuration     Control security aspects of VM:Director
                               (AUTOEXP, JOURNAL, PWSUPRES)

 5 SFS Configuration          Change File Pools and Storage Groups

                  Copyright (c) yyyy CA. All rights reserved.
--------------------------------------------------------------------------------
PF1:Help              PF3:Quit              PF10:Print               PF12:End
```

Options 1 through 4 start XEDIT on the appropriate configuration file for you to edit.

Option 5 displays the SFS Configuration Menu, from which you can add or delete SFS file pools and update user storage group information. CA VM:Director reflects the changes in the SFS CONFIG file.

Note: For more information about setting up or changing the CA VM:Director configuration for SFS, see Configuring CA VM:Director for SFS (see page 199).

If you know which configuration features you want to work with, you can enter the CONFIG command with different parameters to access the appropriate configuration file or configuration submenu directly.

- To work with the AUTHORIZ CONFIG file, enter the following:

  vmdirect config authoriz

- To work with the DASD CONFIG file, enter the following:

  vmdirect config dasd

- To work with the PRODUCT CONFIG file, enter the following:

  ```
  vmdirect config product
  ```

Note: While CA VM:Director is running, you cannot change any ACCESS, DIRECT, or (for CA VM:Director only) ENCRYPT records in this file. If you need to change, add, or remove any of these records, you must stop CA VM:Director first, then edit this configuration file.

- To work with the SECURITY CONFIG file, enter the following:

  ```
  vmdirect config security
  ```

  Note: You cannot change the AUTOPASS record in this file while CA VM:Director is running. If you need to change, add, or remove this record, you must stop CA VM:Director first, and then edit this configuration file.

- To work with the CA VM:Director configuration for the SFS CONFIG file, enter:

  ```
  vmdirect config sfs
  ```

  This displays the SFS Configuration Menu.

Note: For more information about the CONFIG command, see CONFIG Command in the chapter "Command Reference" in the *Reference Guide*.

This section contains the following topics:

AUTHORIZ CONFIG File (see page 51)
DASD CONFIG File (see page 52)
CA VM:Director PRODUCT CONFIG File (see page 56)
CA VM:Director SECURITY CONFIG File (see page 59)

AUTHORIZ CONFIG File

The AUTHORIZ CONFIG file can contain the following three types of records that are used to identify and set up user authorizations for CA VM:Director commands:

- GRANT
- LIST
- WITHHOLD

As part of installation, the AUTHORIZ CONFIG file is created for you and resides on the CA VM:Director A–disk. None of the records are required by CA VM:Director; however, it is recommended that you use them to tailor authorizations to your requirements.

To make changes to any of the records in this file, use the CONFIG command with the AUTHORIZ parameter. All changes take effect immediately. You cannot use these records in any other configuration file.

Sample CA VM:Director AUTHORIZ CONFIG File

The following is a sample CA VM:Director AUTHORIZ CONFIG file:

```
LIST *TESTUSR TONY JOEY MAINT
*
GRANT * TO OPER1 JOEY VMANAGER
GRANT * TO TOMMY OPER2
GRANT MAINT ACCOUNT TO *ALL
GRANT MAINT LINK TO *ALL
GRANT MAINT MDPW TO *ALL
GRANT MAINT MGRID TO *ALL
GRANT MAINT PASSWORD TO *ALL
GRANT MAINT REVIEW TO *ALL
GRANT MAINT RLINK TO *ALL
GRANT MAINT STORAGE TO *ALL
GRANT MAY *SELF TO *ALL
GRANT QUERY ACCOUNT *SELF TO *ALL
GRANT TRANSFER *SELF TO *ALL
GRANT USESEL01 *SELF TO *ALL
GRANT USESEL02 *SELF TO *ALL
GRANT USESEL03 *SELF TO *ALL
GRANT USESEL04 *SELF TO *ALL
GRANT USESEL05 *SELF TO *ALL
GRANT USESEL06 *SELF TO *ALL
GRANT USESEL07 *SELF TO *ALL
GRANT USESEL08 *SELF TO *ALL
GRANT USESEL09 *SELF TO *ALL
GRANT USESEL10 *SELF TO *ALL
GRANT USESEL11 *SELF TO *ALL
```

DASD CONFIG File

The DASD CONFIG file can contain the following six types of records that are used to define the configuration the CA VM:Director will manage:

- DEVTYPE
- EXTENT
- IGNORE
- NOLINK
- SUBPOOL
- VOLUME

As part of installation, the DASD CONFIG file is created for you and resides on the CA VM:Director A–disk.

To make changes to any of the records in this file, use the CONFIG command with the DASD parameter. All changes take effect immediately. You cannot use these records in any other configuration file.

DASD CONFIG File Records

The following table shows the sequence requirements of the records that appear in the DASD CONFIG file. For a sample DASD CONFIG file, see Sample DASD CONFIG (see page 54).

Record	Sequence Requirements	Quantity Allowed
SUBPOOL	Must appear before the specified subpool is referenced by an EXTENT record	Multiple
DEVTYPE	Must appear before the specified device is referenced by a VOLUME record	Multiple
VOLUME	Does not require that an EXTENT record follow it. When one does not, all space on the entire volume is treated as not controlled by CA VM:Director; the volume itself is controlled	Multiple
EXTENT	Must appear immediately following the VOLUME record to which it applies, or following other EXTENT records for the same volume	Multiple
IGNORE	Can appear anywhere in the file	Multiple
NOLINK	Can appear anywhere in the file	Multiple

You can manage DASD through CA VM:Director if you identify the DASD to CA VM:Director in its DASD CONFIG file.

Sample DASD CONFIG File

The following sample DASD CONFIG illustrates the statement use.

```
*
SUBPOOL MAIN ROTATING LOWEND *
*
 VOLUME VMPK01 9336
 EXTENT 0 15 * ALLOCATION
 EXTENT 16 122199 MAIN
 EXTENT 122200 126015 * SPOOL SPACE
 EXTENT 126016 557999 MAIN
 VOLUME VMPK02 3390
 EXTENT 1 3338 MAIN
*
IGNORE $ALLOC$
*
NOLINK MAINT 190
NOLINK MAINT 19E
```

Record	Purpose
SUBPOOL MAIN ROTATING LOWEND *	Defines one subpool, MAIN, with ROTATING LOWEND as the allocation procedure for MAIN, and no maximum size for allocation requests (designated by the asterisk). CA VM:Director uses the volumes in the reverse order from the way they are defined in the DASD CONFIG file. It allocates the first minidisk on VMPK02 and begins at the low end of an available extent. It allocates the second minidisk on VMPK01 and begins at the low end of an available extent.
VOLUME VMPK01 9336	Defines a 9336 DASD with a volume label of VMPK01.
EXTENT 0 15 * ALLOCATION EXTENT 16 122199 MAIN EXTENT 122200 126015 * SPOOL SPACE EXTENT 126016 557999 MAIN	Define four extents for volume VMPK01. The first and third extents are fixed CP extents (indicated by the asterisk), and are used for allocation and spool space, respectively (indicated by the comments following the asterisks). The second and fourth extents are available for minidisk allocations and are assigned to the allocation definitions set up for subpool MAIN.
VOLUME VMPK02 3390	Defines a 3390 DASD with a volume label of VMPK02.

Record	Purpose
EXTENT 1 3338 MAIN	Defines one extent for volume VMPK02. This extent is available for minidisk allocations and is assigned to the allocation definitions set up for subpool MAIN.
IGNORE $ALLOC$	Specifies minidisks to ignore during overlay checking.
NOLINK MAINT 190 NOLINK MAINT 19E	Identifies directory links not to be mapped.

Effects of Changing Records in the DASD CONFIG File

Although you can dynamically change all values on all DASD CONFIG file records, there are certain restrictions and expectations that are established for each record type.

DEVTYPE Records

You can add, change, and delete DEVTYPE records without restrictions. Any new values are used in all subsequent references.

EXTENT Records

You can add EXTENT records without restrictions.

You can delete EXTENT records, but all minidisks defined within those extents will no longer be controlled by CA VM:Director.

When changing the cylinder or block ranges, existing minidisks cannot cross the boundary between two extents and must be either completely within a single extent or completely outside all extents. If you do not follow these guidelines, the extent information will not be updated. Additionally, any minidisks that were previously contained within a defined extent and are not within any extent after the update will no longer be controlled by CA VM:Director.

You can change the subpools listed on an EXTENT record as long as the subpools are still defined. If you change an allocatable extent with subpools to a fixed CP extent, you must use IGNORE records to make all minidisks in that extent ignored to avoid overlay conflicts that will prevent the update from completing.

IGNORE Records

You can add, change, and delete IGNORE records without restrictions, unless removing an IGNORE statement will cause an overlap.

NOLINK Records

You can add, change, and delete NOLINK records without restrictions. However, if you change or delete a NOLINK record so that fewer minidisks are removed from LINK mapping, then CA VM:Director must scan the entire object directory to recreate the LINK mapping structure.

VOLUME Records

You can add VOLUME records without restrictions.

You can delete VOLUME records, but all minidisks defined on those volumes will no longer be controlled by CA VM:Director.

You can change the device type name on a VOLUME record as long as the true underlying device type does not change. For example, assume you have a DEVTYPE record that defines a custom device type, 3390NMDC, based on a 3390 prototype, and a VOLUME record that specifies the 3390NMDC name. You can change the device type name to 3390 or any other name that also resolves to a true 3390 device type.

SUBPOOL Records

SUBPOOL records can be added without restrictions.

You can delete SUBPOOL records as long as the subpools being deleted are not referred to on any EXTENT records.

You can change SUBPOOL records without restrictions. The new values will be used for all subsequent allocations from that subpool.

CA VM:Director PRODUCT CONFIG File

For CA VM:Director, the PRODUCT CONFIG file can contain the following records:

- ACCESS (cannot be changed while CA VM:Director is running)
- ALTERNAT (cannot be changed while CA VM:Director is running and SERVANTS are ON)
- DIRECT (cannot be changed while CA VM:Director is running)
- DUMP
- ESM
- IPLDISK
- MACLOAD
- MAXOLDPW
- MESSAGE
- MSGCASE
- PRODUCT
- RESID (cannot be changed while CA VM:Director is running)
- SERVANT
- SSINODE (cannot be changed while CA VM:Director is running)

- SYSOPER
- USEREXIT

As part of installation, the PRODUCT CONFIG file is created for you and resides on the CA VM:Director A–disk. You cannot use the records in this file in any other configuration file.

To make any changes to the ACCESS, DIRECT, RESID, or SSINODE records, or if you are running with the Servant Facility and want to add an ALTERNAT FORMAT record, you must take down CA VM:Director and make the changes manually in the file. The changes will take effect when CA VM:Director is brought back up. For the remaining records, you can change them dynamically, and the changes take effect immediately.

When you are operating in a Single System Image complex, you must bring down the CA VM:Director server on every member node in order to make a manual change to the PRODUCT CONFIG file. After the change is made, restart each server by issuing an IPL CMS command so that the configuration files are re-accessed properly.

CA VM:Director PRODUCT CONFIG File Records

The following table indicates whether records that can appear in the PRODUCT CONFIG file are required, recommended, or optional. For a sample file, see Sample CA VM:Director PRODUCT CONFIG File in this chapter.

Record	Use	Quantity Allowed
ACCESS	Required for DRCT, BKUP, LOG, and HOLD minidisks. Recommended for the AUDT minidisk. ACCESS COMMON is required if you are using the Servant Facility. An ACCESS LOG record is required if running in a Single System Image complex.	Multiple
ALTERNAT	Optional	Multiple
AUDRECFM	Optional	1
DIRECT	Required	1
DUMP	Recommended	1
ESM	Required if your site is enabling interfacing with an external security manager	1
IPLDISK	Recommended	1
MAXOLDPW	Optional	1
MESSAGE	Optional	Multiple

Record	Use	Quantity Allowed
MSGCASE	Optional; recommend using the MESSAGE record instead	1
PRODUCT	Optional	Multiple
RESID	Required only if your site has implemented the IBM Systems Management API support or Single System Image support	1
SERVANT	Required if using SFS administration; otherwise recommended; requires use of ACCESS COMMON record	1
SSINODE	Optional	Multiple
SYSOPER	Optional	Multiple
USEREXIT	Optional	Multiple

Sample CA VM:Director PRODUCT CONFIG File

The following is a sample CA VM:Director PRODUCT CONFIG File:

```
*
*SERVANT ON VMX$ 120 1 6
*
DUMP VMDIRECT NOCLEAR
*
SYSOPER OPER1
SYSOPER OPER2
RESID VMDIRECT
*
***
*
ACCESS AUDT 1D0 Q
ACCESS BKUP 1B1 P
ACCESS HOLD 1B2 T
ACCESS DRCT 1B0 U
ACCESS RULE 1B4 V
ACCESS COMMON 194 N
*
MESSAGE MSGNOH MIXED
*
DIRECT 1A0 VST023
```

CA VM:Director SECURITY CONFIG File

For CA VM:Director, the SECURITY CONFIG file can contain the following records that provide security features:

- AUTOPASS (cannot be changed while CA VM:Director is running)
- JOURNAL
- PWSUPRES

As part of installation, the SECURITY CONFIG file is created for you and resides on the CA VM:Director A–disk. You cannot use these records in any other configuration file.

CA VM:Director SECURITY CONFIG File Records

The following table indicates whether the records that can appear in the SECURITY CONFIG file are required, recommended, or optional.

Record	Use	Quantity Allowed
AUTOPASS	Optional	1
JOURNAL	Optional	1
PWSUPRES	Optional but recommended	1

Sample CA VM:Director SECURITY CONFIG File

```
*
JOURNAL 1 2
```

Chapter 5: Managing CA VM:Director

CA VM:Director must be properly managed and maintained to provide its fullest benefits for your system management and security requirements.

Proper maintenance of CA VM:Director includes the following:

- Changing the size of the CA VM:Director DRCT and AUDT minidisks
- Verifying that the CP object directory has enough space, and increasing the amount of space available for the CP object directory when necessary
- Altering privilege classes to suit changing processing requirements
- Backing it up regularly and frequently enough to keep any losses to an acceptable level
- Updating the CA VM:Director configuration to accurately reflect the status of your z/VM system or the way users use your z/VM system

This chapter also includes information about how to set up the Servant Facility.

This section contains the following topics:

Making DRCT Volume Changes (see page 61)
Changing the Size of the CA VM:Director DRCT and AUDT Minidisks (see page 62)
Enlarging the CP Object Directory (Moving It to a Different DASD Volume) (see page 64)
Enlarging the CP Object Directory (Keeping It on the Same DASD Volume) (see page 71)
Backing Up CA VM:Director (see page 77)
Reconfiguring CA VM:Director (see page 82)
Servant Facility (see page 84)
Assigning Account Numbers to User IDs (see page 88)

Making DRCT Volume Changes

CA VM:Director rewrites the volume allocation map on the object directory volume each time a directory is updated, caching information from that area.

To change anything affecting cylinder 0 of the volume, such as reallocating the volume using the CPFMTXA utility or relabeling the volume, perform the following steps:

1. Bring CA VM:Director down.
2. Perform the activity to change the allocation map or label of the volume.

3. To have CA VM:Director pick up the new allocation information, reinitialize CA VM:Director using the SOURCE operand.

Note: If you do not perform the tasks in this order, the new volume allocation or label information will be overlaid during the next CA VM:Director directory update.

If changing a volume label, change the DIRECT configuration file record in the PRODUCT CONFIG file and make appropriate DASD CONFIG changes. This may also require changes in other product configurations to indicate the new volume where the online directory resides.

Note: For more information about making changes to an online directory volume, see the following sections in this chapter:

- Changing the Size of the CA VM:Director DRCT and AUDT Minidisks (see page 62)
- Enlarging the CP Object Directory (Moving It to a Different DASD Volume) (see page 64)
- Enlarging the CP Object Directory (Keeping It on the Same DASD Volume) (see page 71)

Changing the Size of the CA VM:Director DRCT and AUDT Minidisks

When the number of user IDs increases on your z/VM system, you may need to increase the size of the DRCT (1B0) minidisk. If the amount of audit information increases, you will need to increase the size of the AUDT (1D0) minidisk.

(CA VM:Secure only) If you install the Rules Facility, you will need to increase the size of the AUDT (1D0) minidisk.

To change the size of the DRCT minidisk or AUDT minidisk

1. From CMS, enter the following:

   `VMDIRECT MANAGE server 6 vaddr`

   server

   Substitute the user ID of your CA VM:Director server virtual machine (usually VMDIRECT). If you are changing the size of an AUDT disk in a Single System Image cluster, then there is an AUDT minidisk defined for each member of the cluster. In that case, substitute the name of the SUBCONFIG directory entry for the member containing the particular AUDT disk you are modifying.

   vaddr

   One of the following:

   - 1B0 for the DRCT minidisk
   - 1D0 for the AUDT minidisk

   The Move a Minidisk screen appears.

2. To change the size of the minidisk, specify the new size in the Size field and pool name in the Pool field if desired.
3. Press Enter to submit your changes to CA VM:Director.

   If there are problems with any of the values you entered, CA VM:Director returns you to that screen where you can correct the problems.

You can also change the size of the minidisks by answering screen prompts, using the MANAGE subfunction of the MAINT command:

`VMDIRECT MAINT MANAGE server DISKMOVE vaddr`

server

Substitute the user ID of your CA VM:Director server virtual machine (usually VMDIRECT). If you are changing the size of an AUDT disk in a Single System Image cluster, then there is an AUDT minidisk defined for each member of the cluster. In that case, substitute the name of the SUBCONFIG directory entry for the member containing the particular AUDT disk you are modifying.

vaddr

One of the following:

- 1B0 for the DRCT minidisk
- 1D0 for the AUDT minidisk

Specify the new minidisk size when prompted. When you have finished, enter **end** in response to the prompt for the next MANAGE subfunction.

You can also change the size of the minidisks by using the CHGMDISK command, specifying all information directly on the command line:

```
VMDIRECT CHGMDISK server vaddr * size devtype vaddr *
```

server

Substitute the user ID of your CA VM:Director server virtual machine (usually VMDIRECT). If you are changing the size of an AUDT disk in a Single System Image cluster, then there is an AUDT minidisk defined for each member of the cluster. In that case, substitute the name of the SUBCONFIG directory entry for the member containing the particular AUDT disk you are modifying.

vaddr *

One of the following:

- 1B0 for the DRCT minidisk
- 1D0 for the AUDT minidisk

Note: For more information about the *size* and *devtype* parameters, see CHGMDISK Command in the chapter "Command Reference" in the *Reference Guide*.

Enlarging the CP Object Directory (Moving It to a Different DASD Volume)

This section describes how to enlarge the CP object directory by moving it to a different DASD volume. For instructions on enlarging the CP object directory without moving it to a different DASD volume, see Enlarging the CP Object Directory (Keeping It on the Same DASD Volume) (see page 71).

Note: Although this section describes *enlarging* the CP object directory, you can use the instructions to *move* the directory without enlarging it. For example, you can move the directory because of hardware failures on the DASD volume on which the directory resides. To simply move the directory, start with Step 2.

Monitor the CP object directory so that you can enlarge its directory extents when it runs out of control block space. This happens during normal directory use.

To check your CP object directory, look for the following messages at CA VM:Director startup time. The first message indicates how many objects (users, profiles, and pool members) you have in the directory and how many pages of space the directory needs to hold the control blocks:

```
065I DIRECTORY CONTAINS number USERS number PROFILES AND
     number POOL MEMBERS ON number PAGES
```

The second message indicates the total directory size and how much is available:

```
294I THE DIRECTORY CONTAINS number TOTAL PAGES. number PAGES
     ARE AVAILABLE FOR USE
```

When the messages indicate that you need to enlarge the CP object directory allocation (there are few pages available compared to the total directory size), take the following steps to calculate the required size, and then to change it to at least that size.

1. Calculate the number of pages that CA VM:Director needs to initialize the directory.
2. Find enough free space on a CP–owned volume for the entire directory.
3. Optionally, allocate a placeholder minidisk in this space.
4. Change the appropriate EXTENT record in the DASD CONFIG file so that it accurately reflects the new location of the CP object directory.
5. Create an ignored minidisk for CA VM:Director to link to, if necessary.
6. Back up the current CP directory.
7. Format and allocate the new extent.
8. Stop CA VM:Director and update the configuration.
9. Restart CA VM:Director.
10. Change the allocation bit map on the OLD DRCT area and re–IPL z/VM.
11. Update links to the CP object directory.

Step 1: Calculate the Number of Pages That CA VM:Director Needs to Initialize the Directory

IBM CP planning guides give space requirements for the CP object directory area. See the appropriate IBM planning and administration documentation for your space requirement, and then use the information below in the IBM formulas for calculating the total control block space (TCBS):

1. Change MAX (1, USERs/8 or 9) to MAX (1, USERs/7).
2. Treat each directory profile on the system as a user ID: add 1 to USERs for each profile and add 1 to DEVs for each device in the profile (this is in addition to the tallying that the IBM formula does for directory profiles).
3. Treat users on hold as active users.
4. Change (TOTDSGRGIDs X 4) to (TOTDSGRGIDs X 8).

The TCBS calculation with these numbers gives the approximate number of pages that CA VM:Director needs to initialize the online directory. It is only approximate because CA VM:Director interactively updates the online directory. Therefore, we recommend that you add about 25 percent to the TCBS that you calculated above for work space.

After you change the directory allocation size, you can determine how close your estimate was by tracking the number of times you see message 065I on the CA VM:Director console while CA VM:Director is running. While this message is generated at CA VM:Director startup to tell you the number of online directory pages currently in use, it is also generated when CA VM:Director runs out of work space and then reclaims it. The reclaiming of work space is called *dynamic rebuild*.

If you see many occurrences of this message during the day, consider increasing the size of the online directory allocation.

Step 2: Find Enough Free Space on a CP–owned Volume for the Entire Directory

As a rule, *enough space* is about double the amount of space you calculated in Step 1. This free space must be on one volume, but it can contain multiple extents and does not need to be contiguous. However, you can improve performance by allocating one contiguous area.

1. To find free space, enter the following:

   `vmdirect map * free`

   This command shows free space on all DASD volumes controlled by CA VM:Director.

   To find where you have directory space allocated, enter the CP QUERY ALLOC DRCT command. Only one directory area will be listed as active. If more than one DRCT area is allocated, we recommend you delete the inactive areas before continuing. To delete the space, mark it as PERM, using the procedure in Step 10.

   The following example shows a portion of the VMDIRECT MAP command response. It displays the availability on 3390–type DASD volume ESA240:

   ```
   *VOLUME: ESA240     TYPE: 3390  UTILIZATION: 83%
   ALLOCATION  SIZE    STARTING    ENDING      USAGE INFORMATION
   *FREE*      75      1           75          *FREE*
   *FREE*      54      190         243         *FREE*
   *FREE*      23      650         672         *FREE*
   ```

2. If the output of this command indicates that there is no free space or not enough free space available, move a minidisk of the correct size to another volume to make the space available.

   Check how much free space is available after you move a minidisk. Use the MAP command to do so. If there is still not enough space, move as many minidisks as necessary to free up the space you need.

3. Calculate the range of the new free space. These numbers are required when you are ready to format and allocate the new extent:

   - For a CKD device, the starting and ending cylinder numbers define the range of this new free space.
   - For an FBA device, the starting and ending page numbers define the range of this new free space

To calculate the starting page number, use this formula:

$$\frac{startingblocknumber + 7}{8}$$

To calculate the ending page number, use this formula:

$$\frac{endingblocknumber - 7}{8}$$

Step 3: Allocate a Placeholder Minidisk in This Space (Optional)

You may want to allocate a minidisk as a placeholder for the CP DRCT area. With this minidisk defined, the CP DRCT area will show up in the output when you use IBM's DISKMAP command against the USER DIRECT file. In addition, if you change to another directory management product, this area will continue to be protected even though it is no longer protected by an entry in the CA VM:Director DASD CONFIG file. (You will create the entry in the DASD CONFIG file in a later step.)

If you want to create a placeholder minidisk, allocate that minidisk now. The recommended owner of this minidisk is $DIRECT$ (your site may have a different naming convention). CA VM:Director will ignore this minidisk in the CA VM:Director DASD CONFIG file. To create a placeholder minidisk (for example, $DIRECT$ 124) to cover the new CP DRCT area, follow these steps:

1. Enter **VMDIRECT CONFIG DASD** and add the following record:

   ```
   IGNORE $DIRECT$ 124
   ```

   Save the change.

2. Enter **VMDIRECT EDIT $DIRECT$** and add an MDISK statement from the beginning of the CP object directory to the end.

   For example, if the new directory is located on volume ESA240, a 3390 device, from cylinder 1 to cylinder 50, add:

   ```
   MDISK 124 3390 1 50 ESA240 RR
   ```

   File the change.

Step 4: Change the DASD CONFIG File So That It Lists the Correct Extents for the CP Object Directory

You must change the DASD CONFIG file to reflect the new location of the CP object directory so that you or other directory managers do not inadvertently allocate a new minidisk over the area you just allocated. Simply covering the area with a placeholder minidisk is not sufficient because subsequently scratching or moving that minidisk can cause CP system failures.

1. Enter the following:

   vmdirect config dasd

2. Find the EXTENT record containing the new allocated area and change it so that it correctly represents the new CP area. Verify that a fixed CP extent covers the new area.

   If the area you just created is in the middle of one extent, split the extent into three to isolate the CP area. For example:

   ```
   EXTENT beg end
   EXTENT beg *      <=== new CP area
   EXTENT beg end
   ```

   Save the changes and close the DASD CONFIG file.

Step 5: Create an IGNORED Minidisk for CA VM:Director to Link to, If Necessary

CA VM:Director requires a read/write minidisk from cylinder 0 to the end of the DRCT area that you intend to allocate in Step 7. This IGNORED minidisk normally belongs to $DASD$ and is usually a full–pack minidisk. If you do not have a full–pack minidisk for the DASD volume that will contain the new CP object directory, do the following:

1. Enter **VMDIRECT CONFIG DASD** and add the following record:

   ```
   IGNORE $DASD$ vaddr
   ```

   vaddr

   The address of the full–pack minidisk you will create in the next step.

   Save this change.

2. Enter **VMDIRECT EDIT $DASD$** and add an MDISK statement from 0 to END on the pack where the new DRCT area will be located. For example:

   ```
   MDISK 124 3390 0 END ESA240 RR
   ```

   Save this change.

Step 6: Back Up the Current CP Directory

As a precaution, back up your current directory before continuing. Use the VMXBKP01 utility to make a backup. VMXBKP01 writes a USER DIRECT file to disk. Send a copy of the USER DIRECT file to OPERATOR's reader. We also recommend you write a copy of this file to tape.

As an extra precaution, you may also want to take a full DDR dump of the volume containing your object directory.

Step 7: Format and Allocate the New Extent

Link to the minidisk described in Step 5 in read/write mode. Format the new extent and allocate it, using the CPFMTXA command. The command responds with a series of prompts to first format and then allocate the directory space.

When you finish with the format/allocate program, press Enter to return to CMS.

For example, the volume that will contain the new object directory is a 3390, linked at virtual address 100. It has a label of ESA240. The new object directory will be located on cylinders 1 through 50.

Important! Do not re-IPL until you complete Step 10: Change the allocation bit map on the old extent and re–IPL z/VM.

The series of command prompts and entries required to format this volume looks like the following, where the command prompts are in light type and the user entries are in bold type:

```
ENTER FORMAT OR ALLOCATE:
format
FORMAT FUNCTION SELECTED:
ENTER DEVICE ADDRESS (CUU):
0100
ENTER DEVICE TYPE:
3390
ENTER START CYLINDER (xxx) OR LABEL:
0001
ENTER END CYLINDER (xxx):
0050
ENTER DEVICE LABEL:
esa240
FORMAT STARTED
FORMAT DONE
000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
ENTER FORMAT OR ALLOCATE:
```

The series of command prompts and user entries to allocate the new directory space looks like the following:

```
ENTER FORMAT OR ALLOCATE:
allocate
ALLOCATE FUNCTION
ENTER DEVICE ADDRESS (CUU):
100
ENTER DEVICE TYPE:
3390
ENTER DEVICE LABEL:
esa240
ENTER ALLOCATION DATA FOR VOLUME ESA240
TYPE CYL CYL
drct 1 50
end
ALLOCATION RESULTS
PERM 0000 0000
DRCT 0001 0050
PERM 0052 3339
DEVICE 100 VOLUME ESA240 ALLOCATION ENDED
ENTER FORMAT OR ALLOCATE:
```

Step 8: Shut Down CA VM:Director

1. Log on to the user ID of your CA VM:Director service virtual machine and enter the END command to shut down CA VM:Director. Do not log off, because you will need to link to the volume containing the new object directory.
2. Go to your CA VM:Director console. Verify that the 191 minidisk is accessed as file mode A.
3. XEDIT file PRODUCT CONFIG A and update the DIRECT record to reflect the volser of the new DASD volume that contains the CP DRCT area.
4. XEDIT file DIRECT STATEMNT A and verify that the new pack is correctly referenced.
5. Access the 1B0 minidisk at file mode U.

   Carefully XEDIT the CA VM:Director source directory entry. The file name of the correct file will be VMDIRECT. Change the definition of the 1A0 minidisk to be a multiple write link to the ignored minidisk described in Step 5. For example:

   ```
   LINK $DASD$ 124 1A0 MW
   ```

   Save this change.
6. Detach the 1A0 minidisk.
7. Use the CP LINK command to link to the new ONLINE directory (the minidisk described in Step 5) as 1A0 with link mode MW.

Step 9: Restart CA VM:Director

Start CA VM:Director with the VMXSYS SOURCE ALTVOL 1A0 volser command and add any other options that you normally use. To check for these options, log on to VMANAGER and issue VMSERVER VMDIRECT. Check the Server Startup Command tag.. CA VM:Director writes the CP object directory to the new directory volume and shuts down with return code 4.

Step 10: Change the Allocation Bit Map on the Old Extent and Re–IPL z/VM

When your z/VM system is IPLed, CP searches for the DRCT area on all online SYSTEM volumes, in slot order, as defined in the CP SYSTEM CONFIG file. CP uses the first DRCT area it finds.

If the pack on which the old DRCT area resides is earlier in the search order than the pack on which the new DRCT area resides, CP uses the old DRCT area when a system IPL is performed. To force CP to use the new DRCT area, change the allocation bit map on the old DRCT area. Do this by executing CPFMTXA. Use the ALLOCATE function and change the allocation to PERM.

For backout purposes, we do not recommend that you reformat the old space at this time.

Bring down your z/VM system normally by using the SHUTDOWN command. Then, IPL your system and verify that CP and CA VM:Director are using the new DRCT area by entering:

```
cp query alloc drct
```

When you are sure that CP and CA VM:Director are using the new DRCT area, you can reformat the old directory space.

Enlarging the CP Object Directory (Keeping It on the Same DASD Volume)

This section describes how to enlarge the CP object directory without moving it to a different DASD volume. For instructions on enlarging the CP object directory by moving it to another DASD volume, see Enlarging the CP Object Directory (Moving It to a Different DASD Volume) (see page 64).

Monitor the CP object directory so that you can enlarge its directory extents when it runs out of control block space. This happens during normal directory use.

To check your CP object directory, look for the following messages at CA VM:Director startup time. The first message indicates how many objects (users, profiles, and pool members) you have in the directory and how many pages of space the directory needs to hold the control blocks:

```
065I DIRECTORY CONTAINS number USERS number PROFILES AND
     number POOL MEMBERS ON number PAGES
```

The second message indicates the total directory size and how much is available:

```
294I THE DIRECTORY CONTAINS number TOTAL PAGES. number PAGES
     ARE AVAILABLE FOR USE
```

When the messages indicate that you need to enlarge the CP object directory allocation (there are few pages available compared to the total directory size), take the following steps to calculate the recommended size and then to change it to at least that size. (The next several pages provide details for completing each of these steps.)

To enlarge the CP object directory

1. Calculate the number of pages that CA VM:Director needs to initialize the directory.
2. Find enough free space on the current CP–owned volume for the entire directory.
3. Update a placeholder minidisk in this space, if necessary.
4. Change the appropriate EXTENT record in the DASD CONFIG file so that it accurately reflects the new location of the CP object directory.
5. Back up the current CP directory.
6. Stop CA VM:Director.
7. Format and allocate the new extent.
8. Restart CA VM:Director.

Step 1: Calculate the Number of Pages That CA VM:Director Needs to Initialize the Directory

IBM CP planning guides give space requirements for the CP object directory area. For more information about your space requirement, see the appropriate IBM planning and administration documentation, and then use the following information in the IBM formulas for calculating the total control block space (tcbs).

Note: Some versions of CP call the TCBS number of records NR, the number of USERs NU, and the number of DEVs ND.

1. Change MAX (1, USERs /8) to MAX (1, USERs /10).
2. Treat each directory profile on the system as a user ID: add 1 to USERs for each profile and add 1 to DEVs for each device in the profile (this is in addition to the tallying that the IBM formula does for directory profiles).
3. Treat users on hold as active users.
4. Change (TOTDSGRGIDs X 4) to (TOTDSGRGIDs X 8).

The TCBS calculation with these numbers gives the approximate number of pages that CA VM:Director needs to initialize the online directory. It is only approximate because CA VM:Director interactively updates the online directory. Therefore, we recommend that you add about 25 percent to the TCBS that you calculated above for work space.

After you change the directory allocation size, you can determine how close your estimate was by tracking the number of times you see message 065I on the CA VM:Director console while CA VM:Director is running. While this message is generated at CA VM:Director startup to tell you the number of online directory pages currently in use, it is also generated when CA VM:Director runs out of work space and then reclaims it. The reclaiming of work space is called *dynamic rebuild*.

If you see many occurrences of this message during the day, consider increasing the size of the online directory allocation.

Step 2: Find Enough Free Space on the Current CP–Owned Volume for the Entire Directory

As a rule, *enough space* is about double the amount of space you have just calculated in Step 1. This free space must be on one volume, but it can contain multiple extents and does not need to be contiguous. However, you can improve performance by allocating one contiguous area.

1. To find free space on the volume that currently contains the CP object directory, enter the MAP command with the volser of the volume that contains the CP object directory (the volume listed in the CP QUERY ALLOC DRCT output) and the FREE parameter. For example, the following command displays the free space on DASD volume ESA240:

   vmdirect map esa240 free

   The command response in the following example displays the availability on 3390–type volume ESA240:

```
*VOLUME: ESA240     TYPE: 3390          UTILIZATION: 83%

ALLOCATION   SIZE   STARTING    ENDING  USAGE INFORMATION
*FREE*       75     1           75      *FREE*
*FREE*       54     190         243     *FREE*
*FREE*       23     650         672     *FREE*
```

2. If the output of this command indicates that there is no free space or not enough free space available, move a minidisk of the correct size to another volume to make the space available.

   Check how much free space is available after you move a minidisk. Use the MAP command to do so. If there is still not enough space, move as many minidisks as necessary to free up the space you need.

3. Calculate the range of the new free space. These numbers are required when you are ready to format and allocate the new extent:

   - For a CKD device, the starting and ending cylinder numbers of this new free space define the range
   - For an FBA device, by the starting and ending page numbers of this new free space define the range

   To calculate the starting page number, use this formula:

   $$\frac{startingblocknumber + 7}{8}$$

   To calculate the ending page number, use this formula:

   $$\frac{endingblocknumber - 7}{8}$$

Step 3: Update the Placeholder Minidisk in This Space, If Necessary

If your site uses a minidisk as a placeholder for the CP areas such as the CP DRCT area, you need to update this user ID's MDISK statement to represent the new location of the CP DRCT area. The benefit of this minidisk is that, when you use the IBM DIRMAP utility against the USER DIRECT file, the location of the CP DRCT area shows up in the output. In addition, if you change to another directory management product, this area will continue to be protected even though the area is no longer protected by an entry in the CA VM:Director DASD CONFIG file. (You will update this entry in the DASD CONFIG file in a later step.)

If you have a placeholder minidisk that represents the current CP object directory, you can update that minidisk now to reflect the new CP DRCT area. For example, if $DIRECT$ 123 is a placeholder for the old CP object directory and the new directory will be located on volume ESA240, a 3390 device, from cylinder 1 to cylinder 50, use the **VMDIRECT EDIT $DIRECT$** command to change the location of $DIRECT$'s 123 minidisk to the following:

```
MDISK 123 3390 1 50 ESA240 RR
```

Step 4: Change the DASD CONFIG File So That It Lists the Correct Extents for the CP Object Directory

You must change the DASD CONFIG file to reflect the new location of the CP object directory so that you or other directory managers do not inadvertently allocate a new minidisk over the area you just allocated. Simply covering the area with a placeholder minidisk is not sufficient because subsequently scratching or moving that minidisk can cause CP system failures.

1. Enter the following:

   vmdirect config dasd

2. Find the EXTENT record containing the new allocated area and change it to so that it correctly represents the new CP area. Verify a fixed CP extent covers the new area.

   If the area you just created is in the middle of one extent, split the extent into three to isolate the CP area. For example:

   ```
   EXTENT beg end
   EXTENT beg *      <=== new CP area
   EXTENT beg end
   ```

   If the newly allocated area begins or ends the current extent, split the extent in two so that the new CP area is isolated. For example:

   ```
   EXTENT beg end
   EXTENT beg *      <=== new CP area
   ```

3. Save the changes and close the DASD CONFIG file.

Step 5: Back Up the Current CP Directory

As a precaution, back up your current object directory before continuing. Use the VMXBKP01 utility to make a backup. VMXBKP01 writes a USER DIRECT file to disk. Send a copy of the USER DIRECT file to OPERATOR's reader. We also recommend that you write a copy of this file to tape.

As an extra precaution, you may also want to take a full DDR dump of the volume containing your object directory.

Step 6: Shut Down CA VM:Director

Log on to the user ID of your CA VM:Director service virtual machine and enter the END command to shut down CA VM:Director. If you do not shut down CA VM:Director, unpredictable results may occur in CA VM:Director or in the outcome of CPFMTXA. Do not log off.

Step 7: Format and Allocate the New Extent

Link to the minidisk that covers cylinder 0 through the end of the new CP DRCT area in read/write mode. Format the new extent and allocate it, using the CPFMTXA command. The command responds with a series of prompts to first format and then allocate the directory space.

When you finish with the format/allocate program, press Enter to return to CMS.

For example, the volume that will contain the new object directory is a 3390, linked at virtual address 100. It has a label of ESA240. The new object directory will be located on cylinders 1 through 50.

Important! Do not re-IPL until you complete Step 10: Change the allocation bit map on the old extent and re–IPL z/VM, in Enlarging the CP Object Directory (Moving It to a Different DASD Volume) in this chapter.

The series of command prompts and entries required to format this volume looks like the following, where the command prompts are in light type and the user entries are in bold type:

```
ENTER FORMAT OR ALLOCATE:
format
FORMAT FUNCTION SELECTED:
ENTER DEVICE ADDRESS (CUU):
0100
ENTER DEVICE TYPE:
3390
ENTER START CYLINDER (xxx) OR LABEL:
0001
ENTER END CYLINDER (xxx):
0050
ENTER DEVICE LABEL:
esa240
FORMAT STARTED
FORMAT DONE
000 NO. PAGE RECORDS WITH READ-CHECK ERRORS
ENTER FORMAT OR ALLOCATE:
```

The series of command prompts and user entries to allocate the new directory space looks like the following:

```
ENTER FORMAT OR ALLOCATE:
allocate
ALLOCATE FUNCTION
ENTER DEVICE ADDRESS (CUU):
100
ENTER DEVICE TYPE:
3390
ENTER DEVICE LABEL:
esa240
ENTER ALLOCATION DATA FOR VOLUME ESA240
TYPE CYL CYL
drct 1 50
end
ALLOCATION RESULTS
PERM 0000 0000
DRCT 0001 0050
PERM 0052 3339
DEVICE 100 VOLUME ESA240 ALLOCATION ENDED
ENTER FORMAT OR ALLOCATE:
```

Step 8: Restart CA VM:Director

Start CA VM:Director with the VMXSYS SOURCE command and add any options that you normally use. Log on to VMANAGER and issue VMSERVER VMDIRECT and check the Server Startup Command tag.. CA VM:Director writes the CP object directory to the new directory area.

Backing Up CA VM:Director

You must consider several of the CA VM:Director minidisks when you make your disaster recovery plan.

Those that must be backed up can be backed up with any of several CA VM:Director utilities, with CA VM:Backup, or with your normal CMS backup procedure.

Note: For more information about any of the CA VM:Director backup utilities, VMXBKP01, VMXBKP02, and VMXBKP03, see the chapter "Utility Reference" in the *Reference Guide*.

CA VM:Director Minidisks and Backup Requirements

The following table lists the labels, default addresses, and backup requirements for the minidisks that comprise the CA VM:Director service virtual machine. Recommended backup procedures for these minidisks appear under Backing Up Other CA VM:Director Minidisks.

Label	Default Address	Contents	Back Up?
AUDT†	1D0	Audit records that are written to the VMDIRECT AUDIT file each time an audited command executes	Yes, but indirectly
BKUP	1B1	Changes that were made to files on the DRCT minidisk	Yes
COMMON†	194	Programs used by the CA VM:Director servants	No
DRCT	1B0	CA VM:Director source directory files	Yes
HOLD	1B2	User IDs that have been placed on hold (not in the CP source directory)	Yes
IPLDISK†	1B3	Program to expire passwords	Optional
LOCAL	191	CA VM:Director configuration files, modified code, and the VMDIRECT MANAGERS file, and all skeleton files	Yes
LOG	1B6	Member synchronization files	No
RUNTIM	192	Executable code for the CA VM:Director service virtual machine	Yes

† These minidisks are optional

Backing Up the Source Directory (DRCT Minidisk)

The DRCT minidisk holds the source directory. The minidisk contains source directory entry files, which combine to form a virtual machine definition for each user ID on the system. CA VM:Director provides two defenses against disasters on this minidisk:

- Writes all source directory updates to the BKUP minidisk
- Allows you to use the following utilities to back up the DRCT minidisk:

- VMXBKP01 utility to create a card image backup
- VMXBKP02 utility to create a mirror image backup
- VMXBKP03 utility to back up to a minidisk

These utilities back up the DRCT minidisk in different ways; the different backups are useful for restoring the source directory under different conditions. In all backups, these utilities erase the contents of the BKUP minidisk as soon as the backup of DRCT is complete, so that the BKUP minidisk always stores only non–backed–up data. This helps ensure that the contents of the BKUP minidisk do not duplicate or overlap the backed up data.

Note: For more information about these utilities, see the chapter "Utility Reference" in the *Reference Guide*.

Creating Card Image Backups

You create a card image backup of the DRCT source directory minidisk on disk by using the VMXBKP01 utility. A card image backup is a single file that contains all of the source records in the form that is acceptable to the DIRECTXA command.

Note: Record the physical location of the minidisk containing the disk backup, and have it available and incorporated into your site's disaster recovery plans.

Creating Mirror Image Backups

You create a mirror image of the DRCT minidisk on another minidisk by using the VMXBKP02 utility. This utility uses the IBM DDR utility. A mirror image backup is a duplicate copy of the DRCT minidisk that can be used directly by CA VM:Director in the event the original DRCT minidisk is corrupted.

We recommend that you perform your daily backups of the CA VM:Director DRCT minidisk with either this VMXBKP02 utility or the VMXBKP03 utility (see below). If a hardware failure corrupts the DRCT minidisk, you can restore the source directory and put it back in service immediately from either of these backups.

Creating Backups to Minidisks

You back up the DRCT minidisk by copying it to another minidisk with the VMXBKP03 utility. This utility uses the file–copy utility that is specified on the ALTERNATE record in the PRODUCT CONFIG file, or, if that record is absent from the file, the CMS COPYFILE command.

VMXBKP03 copies all the files from the DRCT source directory minidisk to a minidisk that is owned by the user ID that uses the VMXBKP03 utility.

We recommend that you perform your daily backups of the CA VM:Director DRCT minidisk with either this VMXBKP03 utility or the VMXBKP02 utility (see above). If a hardware failure corrupts the DRCT minidisk, you can restore the source directory and put it back in service immediately from either of these backups.

Backing Up Other CA VM:Director Minidisks

The following sections summarize the contents of each minidisk and recommends backup procedures for each.

AUDT Minidisk

Contents

Audit records that CA VM:Director accumulates since the last execution of the AUDITEXT command.

Recommended backup procedure

Do not back up the AUDT minidisk directly. Instead, on a regular basis, use the AUDITEXT command to extract the audit data from this minidisk and place it on your auditor's minidisk. Back up the audit data file from its new location using your normal backup procedures for user minidisks.

COMMON Minidisk

Contents

All modules and EXECs that the Servant Facility uses to move and copy minidisks.

Recommended backup procedure

CA VM:Director copies the necessary files to the COMMON minidisk during CA VM:Director initialization; therefore, no special backup procedure is necessary. Use CA VM:Backup or your normal backup procedure.

If the minidisk ever becomes corrupted, dynamically rebuild the product configuration information by issuing the CONFIG PRODUCT command. This command displays the PRODUCT CONFIG file in XEDIT. Simply file the PRODUCT CONFIG file without making changes to rebuild the COMMON minidisk. During dynamic reconfiguration of the PRODUCT CONFIG file, CA VM:Director will dynamically allocate and initialize a new COMMON minidisk. If the permanent COMMON minidisk was corrupted, you must reformat it with the CMS FORMAT command.

HOLD Minidisk

Contents

Directory entry files of user IDs on hold.

Recommended backup procedure

Use CA VM:Backup or your normal CMS backup procedure for daily backups. At least once a week, run the VMXBKP01 utility with the HOLD option. This option includes user IDs on the HOLD minidisk in its source directory backup.

IPLDSK Minidisk

Contents

An IPLable program that forces users to change their password.

Recommended backup procedure

None. CA VM:Director can recreate the IPLDSK minidisk at any time.

If you choose to back it up anyway, use CA VM:Backup or your normal non–CMS backup procedure. Treat the IPLDSK as a non–CMS minidisk for CA VM:Backup backups.

LOCAL Minidisk

Contents

CA VM:Director configuration files, VMDIRECT MANAGERS file, skeleton files.

Recommended backup procedure

CA VM:Backup or your normal CMS backup procedure.

RUNTIM Minidisk

Contents

All CA VM:Director program material that was deployed from the product user ID BUILDSRV minidisk.

Recommended backup procedure

CA VM:Backup or your normal CMS backup procedure.

Restoring from the BKUP Minidisk

CA VM:Director writes updates to the DRCT source directory minidisk and the BKUP minidisk. If you lose your DRCT source directory minidisk, you can restore your source directory from the last backup you made, and then copy over any new files from the BKUP minidisk. To protect against data loss in the event of hardware problems, put the DRCT and BKUP minidisks on different DASD volumes.

Reconfiguring CA VM:Director

You need to reconfigure CA VM:Director when you change your z/VM system or change the users that use your z/VM system. For instance, adding new directory managers, granting users authority to use new commands, or adding new DASD require that you reconfigure CA VM:Director.

You can reconfigure most aspects of CA VM:Director at almost any time by using the ADMIN command or the CONFIG command.

The ADMIN command changes directory manager information, directory pool definitions, directory profiles, skeleton files, and SFS managers; the CONFIG command changes users' authorizations to CA VM:Director commands, descriptions of DASD that CA VM:Director is responsible for managing, CA VM:Director processing behavior, and descriptions of SFS if you use it.

Make these configuration changes by editing (or adding or removing) records in various configuration files or by filling in screen fields. The configuration feature you change determines which configuration file you change or which screens you fill in.

Configuration feature	Command that lets you change this feature
Directory manager	ADMIN MANAGERS

Configuration feature	Command that lets you change this feature
Global definitions used during directory processing (for example, default machine mode for any user directory entry that does not contain a MACHINE statement)	ADMIN GLOBALS
POSIX group information	ADMIN POSIX
SFS manager	ADMIN SFSMGRS
Skeleton file	ADMIN SKELETON *skelname*
Directory pool's master definition	ADMIN POOL *poolname*
Directory profile	ADMIN PROFILE *profname*
User authorizations to use commands	CONFIG AUTHORIZ
Descriptions of DASD that CA VM:Director manages†	CONFIG DASD
Product processing behavior	CONFIG PRODUCT
Security	CONFIG SECURITY
Servant Facility	CONFIG PRODUCT
SFS file pools or user storage groups	CONFIG SFS

† Although allocating minidisks and serializing these allocation requests are configuration changes, they must be done only through the MANAGE command and other CA VM:Director features that are designed to automate DASD management.

You can make most of these configuration changes while CA VM:Director is running. They are applied to CA VM:Director dynamically and take effect as soon as you save them. Others require that you first terminate CA VM:Director. These changes are applied to CA VM:Director when you restart it after making those changes.

Note: For more information about which records can be added with the CONFIG command and which records have to be added manually after CA VM:Director is terminated, see Configuration Files and Records (see page 49).

Servant Facility

The Servant Facility is a set of special worker virtual machines that help CA VM:Director perform long–running CMS tasks such as formatting and copying minidisks, and deleting and renaming SFS file spaces.

These extra worker virtual machines, or servant machines, let directory managers start minidisk management functions and continue to work while the minidisk management functions are processed. They save time and increase productivity.

You can use the Servant Facility whenever your staff has many minidisk management functions to perform; you can turn the Servant Facility off and on at any time.

Note: You must use the Servant Facility if you are using CA VM:Director for SFS administration and want to use the DELETE and CHANGE commands. With the Servant Facility, potentially long–running commands can execute on an CA VM:Director servant instead of tying up the CA VM:Director service virtual machine.

When the Servant Facility is turned on, CA VM:Director dynamically creates and autologs servant user IDs when it needs them. It creates as many servant virtual machines as it requires, up to the limit you choose or up to 9999, whichever is smaller.

Servant user IDs are created in the range VMX$0001 to VMX$9999 by default (you can change the first four characters of their user IDs). CA VM:Director uses its own directory entry as a prototype for the directory entries for these servant user IDs, and, as for all directory entries created from prototypes, it strips all minidisk references from each new directory entry and changes the user ID, password, and other statements as necessary.

Note: The servant user IDs are created with a maximum virtual storage (vstor) size of 24M. Additionally, because the servant user IDs are modeled after the VMDIRECT directory entry, you cannot have an IPL 190 statement in the CA VM:Director service virtual machine’s directory entry.

When CA VM:Director requires a servant for a process, it places the process in a wait queue and searches for an available servant. If it finds an available servant, it attaches that servant to the process and places the process in the ready queue. If it cannot find an available servant, it creates a directory entry for a new servant and adds the entry to the object directory. It then autologs the new servant, attaches the servant to the process, and places the process in the ready queue.

When a process finishes with a servant, it releases the servant. CA VM:Director then removes any links that were made during the processing and verifies the number of free servant user IDs against the number that must remain standing by. If there are more free servant user IDs than required, CA VM:Director logs off the extra ones.

You define the characteristics of the Servant Facility, including the number of free servants that can remain waiting and the maximum number of servants that can exist, through the SERVANT record. This record exists in the PRODUCT CONFIG file.

Note: For more information about defining Servant Facility characteristics, see SERVANT Record in the chapter "Configuration File Reference" in the *Reference Guide*.

Setting Up the Servant Facility

To set up the Servant Facility

1. Examine the PRODUCT CONFIG file to see if there is an ACCESS COMMON record. (Presence of this record means the COMMON minidisk is already defined.) Enter the following:

   ```
   vmdirect config product
   ```

   This opens the PRODUCT CONFIG file for editing in XEDIT.

   - If the ACCESS COMMON record is already in the file, and it is defined at address 194, continue to Step 2.
   - If it is not defined at address 194, you will only have to change the current address on the ACCESS COMMON record to 194; you do not have to create a new COMMON minidisk.

     a. Stop CA VM:Director with the END command.

     b. Change the current address on the ACCESS COMMON record to 194. The new ACCESS COMMON record will be:

        ```
        ACCESS COMMON 1B4 C
        ```

     c. Continue to Step 2.

- If there is no ACCESS COMMON record, you will need to add one:

a. Exit the PRODUCT CONFIG file.

b. Create a minidisk for the CA VM:Director service virtual machine at address 194 with a label of COMMON. You can use any method to create the minidisk. 540 4K blocks 3390 cylinders or an equivalent number of FBA blocks will be sufficient.

c. Stop CA VM:Director with the END command.

d. Add the following record to the PRODUCT CONFIG file to identify the COMMON minidisk:

```
ACCESS COMMON 194 C
```

Note: For more information, see ACCESS record in the chapter "Configuration File Reference" in the *Reference Guide*.

e. Continue to Step 2.

2. Verify that the SERVANT record is set to ON. If it is not, change it to ON.

   Note: For more information, see SERVANT record in the chapter "Configuration File Reference" in the *Reference Guide*.

3. If you made any changes, save this XEDIT session and exit the PRODUCT CONFIG file.

   If CA VM:Director is still running, the Servant Facility is enabled as soon as you exit the PRODUCT CONFIG file.

4. If you had to stop CA VM:Director, restart it with the IPL CMS command.

You can disable and enable the Servant Facility dynamically by editing the SERVANT record in the PRODUCT CONFIG file. The change takes effect immediately.

Using the Servant Facility with a Monitoring Program

If you install a system-monitoring program that forces off idle users, have it ignore the CA VM:Director servant machines. If the monitor does not ignore the servant machines, it may force the servants off when they are idle.

Using the Servant Facility with an External Security Manager

If you install a security package, define the servant user IDs to the security package.

In addition to defining the servant user IDs to the security package, create privileges to allow the servants to link to the following minidisks, as necessary:

- CA VM:Director 194 minidisk
- CA VM:Director 600-6FF minidisks

Disabling the Servant Facility

To disable the servant facility

1. Enter the following:

   vmdirect config product

   This opens the PRODUCT CONFIG file for editing in XEDIT.

2. On the SERVANT record, change ON to OFF:

   SERVANT ON *prefix timeout numfree numactive*
   SERVANT OFF *prefix timeout numfree numactive*

   Save this XEDIT session and exit the PRODUCT CONFIG file.

The Servant Facility is disabled as soon as you exit the PRODUCT CONFIG file. CA VM:Director deletes all servant machines that are not attached to a process. Those that are attached to a process are deleted when the process ends.

Enabling the Servant Facility

To enable the Servant Facility that has been previously disabled

1. Enter the following:

   vmdirect config product

2. On the SERVANT record, change OFF to ON:

   SERVANT OFF *prefix timeout numfree numactive*
   SERVANT ON *prefix timeout numfree numactive*

   Save this XEDIT session and exit the PRODUCT CONFIG file.

The Servant Facility is enabled as soon as you exit the PRODUCT CONFIG file.

Querying the Status of the Servant Facility

To see the current status of the Servant Facility, issue the following CA VM:Director QUERY command:

vmdirect query servant

If the Servant Facility is disabled, CA VM:Director responds that it is off.

If the Servant Facility is enabled, CA VM:Director responds that it is on and displays the maximum number of servants allowed, the number currently free, and the maximum number that can remain idle. This information comes from the SERVANT record you added to the PRODUCT CONFIG file.

Assigning Account Numbers to User IDs

CA VM:Director enables you to control access to system resources by verifying account numbers when you assign them to a user ID.

Use the *AC= special comment to list account numbers that a user ID is allowed to use. You must have previously defined the account numbers using the ACCOUNT directory statement.

When you create a user ID or change a user ID account number (using the MAINT command), the change can apply immediately or at next logon or can be a temporary change. CA VM:Director follows the procedure below to verify whether the new account number is valid for that user ID.

First, it scans the user ID directory entry file for any *AC= special comments for account numbers the user can use:

- If CA VM:Director finds the new account number listed on a *AC= special comment in the user ID directory entry, CA VM:Director verifies the new account number and updates the new account number as you request, either at the next logon, temporarily, or immediately.
- If CA VM:Director does not find the new account number listed on a *AC= special comment in the user ID directory entry, CA VM:Director calls the ACCOUNT user exit to verify the change. If you have not installed the ACCOUNT user exit or if the ACCOUNT user exit rejects the change, CA VM:Director does not verify the new account number and generates an error.

If the account number is valid, CA VM:Director next determines how to change the account number based on whether you use the CA VM:Account interface:

- If you use the CA VM:Account interface, all account number changes take effect immediately and are permanent. CA VM:Account issues DIAGNOSE X’84’ to update the VMDBK, and CA VM:Director updates the CP directory for the next logon.
- If you do not use the CA VM:Account interface, and you have selected either an immediate change or one at the next logon, CA VM:Director updates the new account number in the CP directory. If you have selected either an immediate or temporary change, CA VM:Director also updates the VMDBK using DIAGNOSE X’84’.

Note: For information, see the ACCOUNT user exit in the chapter "User Exit Reference" in the *Reference Guide*. For more information about the CA VM:Account interface, see the *CA Mainframe VM Product Manager Product Interface Guide*.

Chapter 6: Managing CA VM:Director in a Single System Image Complex

A VM Single System Image (SSI) complex is a multisystem environment. An SSI enables you to manage separate VM systems as if they were a single computing resource. CA VM:Director maintains a consistent view of system administration definitions across all members of an SSI complex.

This section contains the following topics:

Overview (see page 90)
Distributed Server Definition (see page 91)
Command Processing in an SSI Complex (see page 91)
Directory Entry Processing (see page 91)
Configuration Statements in an SSI Complex (see page 93)
Disk Space Management in an SSI Complex (see page 96)
Servant Facility in an SSI Complex (see page 97)
SSI Installation Considerations (see page 97)
SSI Miscellaneous Considerations (see page 99)

Overview

The presentation of a consistent view of system administration across an SSI complex requires that CA VM:Director support the following environment characteristics:

- The CP Object Directory is identical on every member system. A USER directory entry defines a user ID that can log in any member system (one at a time). The virtual machine definitions are identical. An IDENTITY directory entry defines a user ID that can log in multiple members simultaneously. The member-specific definitions in a SUBCONFIG directory entry can tailor each logon instance.
- CA VM:Director provides identical access to the directory management or resource access control administration interfaces from all members of the complex. You can enter product commands in the same way from any member system.

Change virtual machine definitions and configuration file statements simultaneously in every member system. This synchronization preserves the single system image. To accomplish this synchronization in real time, CA VM:Director operates as a set of distributed servers, one on each member system. These servers communicate with each other. Each server runs in one of the following two modes:

A *master server* runs on one member node to perform the following functions of a non-SSI CA VM:Director server:

  - Processing commands
  - Updating and compiling Configuration files
  - Updating and compiling CP Directory Entries
- An *agent server* runs on every other member node to perform the following subset of the functions of the master CA VM:Director server:
  - Compiling Configuration files
  - Compiling CP Directory Entries
  - An agent server implements the following new functions:
    - Responding to synchronization requests from the master server
    - Converting itself to replace a master when the master server has an outage.

Distributed Server Definition

CA VM:Director implements the product server virtual machine definition as an IDENTITY entry in an SSI complex. This implementation enables the server (usually VMDIRECT) to log in to all cluster member systems at the same time. Additional changes in the product server virtual machine definition are as follows:

- Audit minidisks are member-specific and are defined in a SUBCONFIG entry specific to each member.
- The minidisks that are defined for the CP Object Directory area are member-specific and are defined in a SUBCONFIG entry specific to each member.
- The distributed product server virtual machines communicate using APPC protocol over a GLOBAL APPC resource definition. You create the definition by adding an IUCV *IDENT statement to the IDENTITY directory entry.
- A two-phase commit process is used to synchronize changes across the complex. A new LOG minidisk is defined to the product server to record the status of a change while it is in flight.

Command Processing in an SSI Complex

The MODULE command program issues product commands on any member system in the complex. The commands always execute on the Master CA VM:Director server. If a command changes one of the following entities, the Master server instructs each Agent to make the same changes in its environment.

- A directory entry
- (CA VM:Secure only) An access control rule
- The server configuration

Certain diagnostic or maintenance commands can be directed to run on a particular server. That direction requires more command parameters. For more information, see Member Specific Command Processing (see page 100).

Directory Entry Processing

Product commands have been extended to support the new IDENTITY and SUBCONFIG directory entry types. These new entry types force restrictions on how product commands process them. Those processing restrictions are documented in this section and in the detailed command descriptions in the *Reference Guide*.

IDENTITY Entry Processing

Commands operate on IDENTITY entries as if they are USER entries. The following special processing rules are in force for IDENTITY entries:

- An IDENTITY statement can be present in a SKELETON entry.
- A USER entry can be changed into an IDENTITY entry, or an IDENTITY entry (containing no BUILD statements) can be changed into a USER entry. The CHGENTRY command accomplishes this change as follows:

  ```
  CHGENTRY entry [USER|IDENTITY]
  ```

- BUILD statements are added to an IDENTITY entry whenever the ADDENTRY command creates a SUBCONFIG entry.
- The removal of a SUBCONFIG entry by the DELENTRY command removes BUILD statements from an IDENTITY entry.
- The EDIT, EDX, or REPENTRY commands cannot add or delete BUILD statements.
- A BUILD statement within an IDENTITY entry can name a SUBCONFIG entry which does not exist.
- A SUBCONFIG entry can exist without a BUILD statement naming it.

SUBCONFIG Entry Processing

Only a subset of product commands operates on SUBCONFIG entries, because only a subset of statement types is enabled in these entries. The creation of a SUBCONFIG entry creates a BUILD statement in a specified IDENTITY entry. The deletion of a SUBCONFIG entry by the DELENTRY command removes the BUILD statement in the appropriate IDENTITY entry.

SUBCONFIG Creation

The ADDENTRY command supports the IN and ON options for SUBCONFIG creation as follows:

ADDENTRY *entry* (IN *identity* ON *membername*

entry

Names the new SUBCONFIG entry.

identity

Names the IDENTITY entry where a BUILD statement, referencing the new SUBCONFIG, is added.

membername

Names the member where the SUBCONFIG is active. For an SSI Ready Object Directory, the member name must be an asterisk. For an SSI Enabled Object Directory, the member name must be previously defined on an SSINODE configuration statement in the PRODUCT CONFIG file. For more information about configuration, see Configuration Statements in an SSI Complex (see page 93).

When creating a SUBCONFIG entry, ADDENTRY requires one of the following conditions:

- A SKELETON file starting with a SUBCONFIG statement
- A file containing the desired SUBCONFIG entry statements

As part of the SUBCONFIG creation process, a BUILD statement naming the new SUBCONFIG is added to the named IDENTITY entry.

SUBCONFIG Removal

The DELENTRY command supports SUBCONFIG deletion as follows:

DELENTRY *entry*

entry

Names the SUBCONFIG entry to remove.

Part of the SUBCONFIG deletion process is the removal of a BUILD statement naming the SUBCONFIG from the owning IDENTITY entry.

Configuration Statements in an SSI Complex

This section documents the configuration statements that are required when operating in an SSI complex.

ACCESS

The ACCESS statement specifies the LOG minidisk address. Create the minidisk in the product server IDENTITY entry if it does not exist, and add the following statement to the PRODUCT CONFIG file:

```
ACCESS 01B6 LOG
```

RESID

The RESID statement specifies the establishment of a connection for APPC protocol resource-based communications. The following statement in the product server IDENTITY directory entry defines the resource:

```
IUCV *IDENT resource GLOBAL
```

The server PRODUCT CONFIG file defines the connection to the global APPC resource by including the following statement:

```
RESID resource GLOBAL
```

The resource parameter value is a one-to-eight character resource name which is unique within your VM system ISFC collection.

Note: For more information, see “RESID Record" in the *Reference Guide*.

DIRECT

The DIRECT statement specifies the device address and volume serial for a full-volume minidisk on the volume containing the CP Object Directory cylinders. The product server or another ID owns this minidisk. A LINK allows access to the minidisk. Each member in an SSI complex has its own Object Directory volume. Place the MDISK or LINK for the volume of each member in the server SUBCONFIG entry. The virtual address must be the same. Because each volume serial is different, use the following form of the DIRECT statement in the PRODUCT CONFIG file:

```
DIRECT vdevno m01res m02res m03res m04res
```

The vdevno parameter specifies the virtual address. From one to four volume serials are specified, one for each member.

SSINODE

You define an SSINODE statement to name each member. You can place up to 32 SSINODE names in the PRODUCT CONFIG file, but SSI currently uses only four. Use the same member names that you specified in the SYSTEM CONFIG file. Use the following format of the SSINODE statement:

```
SSINODE name
```

When SSINODE is present, then the source directory is compiled into an SSI Enabled format Object Directory. This format requires that valid node names be present on any BUILD statements in an IDENTITY directory entry. Use this format of the Object Directory only on a member of an SSI complex. Unpredictable results occur if an SSI Enabled Object Directory is used on a non-SSI system. Refer to the IBM CP Planning and Administration publication for the possible problems running this combination.

When SSINODE is not present, then the source directory is compiled into an SSI Ready format Object Directory. This format requires an asterisk to represent the node name on a BUILD statement in an IDENTITY directory entry. This format of the Object Directory is allowed on the following entities:

- A non-SSI VM system
- An SSI system that is configured with only one member node

The latter configuration enables you to more easily convert a non-SSI system into a single member node SSI system.

For a description of this conversion process, see SSI Installation Considerations (see page 97).

Disk Space Management in an SSI Complex

Disk space in an SSI complex is shared and accessible from every member system. However, you can configure certain volumes to be accessible from only one member. Any minidisk defined in a SUBCONFIG entry can be accessed only from within the member system where that SUBCONFIG is active.

We recommend that you separate shared volumes and unshared volumes for minidisk allocation. One method to accomplish the separation is to use separate allocation POOLs. For example, you could perform the following configurations:

- One or more pools for shared volumes
- One pool per member system for the volumes which are dedicated for use by that member

We recommend that you configure the product Servant Facility feature to simplify SSI operation. The product server is able to launch a servant virtual machine in any member node. The product server can then format or copy the minidisks that are defined within SUBCONFIG entries. For more information, see “Servant Facility in an SSI Complex.”

If the servant facility is not in use issue the following kinds of commands from the member node that has access to the minidisk:

- Commands that include long running operations, such as FORMAT, COPYFILE, or DDR processes
- Commands that operate on nonshared minidisks

The SSI environment does not allow any operation which involves the parallel use of two minidisks defined in SUBCONFIG directory entries for different member systems. For example, you cannot use a command like DUPMDISK to duplicate a minidisk in one SUBCONFIG to another SUBCONFIG on a different member. To accommodate this restriction, you can perform the following procedure:

1. Duplicate a nonshared minidisk to a shared minidisk.
2. Duplicate the shared minidisk to a new unshared minidisk.
3. Remove the intermediate shared minidisk.

Servant Facility in an SSI Complex

We strongly recommend that you configure the Servant Facility in an SSI environment. This configuration simplifies the execution of product commands which execute long running minidisk operations such as formatting and copying. The product server launches servant virtual machines on any member system, as they are needed. Without the servant facility, the format and copy operations execute in the virtual machine that issues the command. The issuer must then run on a specific member system when the minidisk involved is reachable only from that member. If the servant facility is used, the command issuer is not tied to a particular member but can run on any member. This arrangement simplifies the operation of the SSI complex.

SSI Installation Considerations

CA VM:Director is set up to manage a VM Single System Image complex by one of the following two methods:

1. An existing non-SSI VM system, already running CA VM:Director can be converted to an SSI VM system.
2. CA VM:Director can be installed on an SSI complex where it is not currently running.

Converting non-SSI to SSI

This method starts with a non-SSI VM system, where you have already installed CA VM:Director. Follow the IBM procedure “Converting a z/VM System to a Single-Member z/VM SSI Cluster” from the *CP Planning and Administration* document. The required CA VM:Director steps are as follows:

1. For any directory entry that you want to run concurrently on multiple members of the cluster, perform the following actions. During the “Update the User Directory” step, perform the following actions:

   - Change USER directory entries to IDENTITY directory entries by using CA VM:Director commands.
   - Restructure these virtual machine definitions by moving MDISK and other statements from the IDENTITY to a SUBCONFIG entry.

   a. Change the type of a USER directory entry to an IDENTITY, by using the following format of the CHGENTRY command:

      `CHGENTRY entry IDENTITY`

   b. Create the needed SUBCONFIG entry by using the following format of the ADDENTRY command:

      `ADDENTRY NAME-1 ( IN NAME ON *`

   c. Move needed MDISK definitions from the IDENTITY to the SUBCONFIG by using the following format of the TRANSFER command:

      `TRANSFER NAME 191 NAME-1 191`

   d. Move other required statements from an IDENTITY to a SUBCONFIG using the EDIT command.

   e. Convert the VMDIRECT directory entry into an IDENTITY with SUBCONFIG format.

   For more information about the requirements for the product server, see Distributed Server Definition (see page 91).

2. Instead of performing the “Change the User Directory to SSI-Enabled” step by running DIRECTXA, use product facilities to accomplish this conversion as follows:
   a. LOGON to the product server and issue the END command if it was running.
   b. ACCESS the product source directory minidisk (usually 01B0).
   c. Execute the SSIENAB utility program. This utility performs the following actions:
      - Retrieve the SSI member name from the running system.
      - Convert all BUILD statements on the source directory minidisk from the form “BUILD ON *” to the form “BUILD ON membername”.

        These steps convert the source to SSI Enabled format.

      Note: For more information, see the SSIENAB utility in the *Reference Guide*.

      Use the following form of the command:

      `SSIENAB ON filemode`

      filemode

      Specifies the CMS file mode letter where the source directory minidisk is accessed.
   d. Add an SSINODE statement to the PRODUCT CONFIG file.

      Note: For more information about the statement, see the *Reference Guide*.

      Use the following form of the command:

      `SSINODE membername`
   e. IPL CMS to bring up the product server. This action performs a SOURCE start to rewrite the CP Object Directory in SSI Enabled format.

Installing on an SSI Complex

This method is a new installation of the product on a running VM system. The differences between installing in a non-SSI complex or in an SSI complex are minor, and are documented in the *Installation Guide*.

SSI Miscellaneous Considerations

New capabilities in CA VM:Director interfaces support the unique environmental characteristics of an SSI complex.

Journaling Additions

The journal facilities lock out users or terminals that make excessive access attempts with incorrect passwords. In an SSI complex, journaling is extended to be an attribute of the complex, rather than an attribute of one member. Any lockout is applied to reject logon attempts from any member after the incorrect password limit is exceeded on the same or a different member. The incorrect attempt counts are known and accumulated across all members in the complex.

The unique device address of the terminal identifies journal entries for terminals on a non-SSI system. On an SSI complex, the same terminal address can be used on different member systems to identify different terminals. Journal entries for the SSI environment add the member system node name to the journal entry as the unique identifier of a specific terminal.

Member Specific Command Processing

You can operate CA VM:Director as a set of distributed servers in an SSI complex. In this case, product commands are passed to the *Master* server for execution. A limited number of product commands are appropriately applied only on a specific member. One of these commands is the AUDITEXT command which extracts member-specific audit data from a product server. For this situation, a new LOCAL command prefix is defined. Using the command prefix directs a given command to execute on the server running in the same member node as the command issuer.

Example

This example shows how to extract the audit file from a specific member product server named VMDIRECT.

Issue the following command from a user ID running on the same member system as the VMDIRECT server:

```
vmdirect local auditext
```

Recovery After an Outage

If one of the CA VM:Director product servers in an SSI complex experiences an outage, it is important to recover quickly. An outage can occur because of an error in the product server or in the member VM system where it is running. The best solution is to restart the failing SSI member or failing CA VM:Director product server virtual machine or both. If you are able to do that, normal operations resume with your complex fully functional. If a member or product server cannot be made operational, you have one of the following situations:

- The SSI member running the master product server is down.
- The SSI member running an agent product server is down.
- The master product server is down.
- An agent product server is down.

Master Server Member Outage

If the member running the Master CA VM:Director server was not shutdown in an orderly manner, the remaining members cannot share the disk space. The member that causes the problem must be manually removed from the complex. To remove the member, issue the following command:

```
SET SSI MEMBER membername DOWN
```

Normal CA VM:Director operation resumes when you issue the TAKEOVER command on one of the SSI member nodes which is still operating. The TAKEOVER command converts one of the Agent product servers into a Master product server. Any of the Agent servers can be converted to become the new Master server. From an authorized user ID on the selected node, issue the following command:

```
vmdirect local takeover
```

Agent Server Member Outage

When a member running an Agent server is down, the CA VM:Director server continues to process commands and to update the Object Directory. These changes are resynchronized when the downed Agent server comes back up.

Problems can arise when you make directory changes to user IDs that are XAUTOLOGged during the system IPL before the CA VM:Director server comes up. These user IDs may include OPERATOR, AUTOLOG1, DISKACNT, EREP, OPERSYMP, and VMDIRECT. These problems arise because the Object Directory still has the contents as they existed before the Agent server member was taken down. To recover successfully, perform the following steps before you perform the IPL of the downed member:

1. Obtain a copy of the USER DIRECT file, which contains the latest set of changes. To do so, execute the VMXBKP01 utility on one of the operational member systems.
2. Attach the volume containing the Object Directory for the member that is down. To rewrite the Object Directory on the downed member, execute the DIRECTXA utility.
3. Bring up the member and start CA VM:Director normally.

Master Product Server Outage

If a Master product server is down but the member VM system is still operational, no CA VM:Director product commands are processed. Therefore the Object Directory is not updated on any member system. Normal recovery can be as simple as restarting the server. If you must restore volumes containing the Object Directory or any CA VM:Director minidisks, perform the following steps:

1. Bring down all the Agent servers on other nodes.
2. Restart the Master server.
3. Restart the Agent servers.

The Object Directory can change during the recovery procedures. If this change occurs, resynchronize the Object directory on all members of the complex before restarting the CA VM:Director servers. See the Agent Server Member Outage (see page 102) section on how to resynchronize these Object Directory areas by running VMXBKP01 and DIRECTXA.

Agent Product Server Outage

If an Agent server is down but the SSI member it runs on is operational, the following problem can occur. Any Object Directory changes on the other member systems could make the Object Directory on the member where CA VM:Director is down become back-leveled.

References to unsynchronized Object Directories to minidisks which have been moved can result in security exposures or errors in programs referencing these minidisks.

The safest approach is to shut down the member where the failing CA VM:Director server was running, until it can be repaired.

Chapter 7: Authorizations

You can divide responsibilities among users by carefully granting user IDs the authority to use different commands. (By default, authorization to use any CA VM:Director command is withheld from all users.) You can authorize a user ID to issue commands on behalf of itself and other user IDs. To help you create and customize user authorizations, CA VM:Director also provides predefined lists of user IDs and special processing authorizations.

You need to determine the level of authorization you want to grant each user. Before you set up authorizations for your system, consider the following guidelines:

- Grant the CA VM:Director system administrator authorization to use all commands and menu selections.
- Grant directory managers authorization to use a particular command, group of commands, or menu selection.

  By carefully planning these authorizations, you can delegate many of the daily directory and disk space management tasks to the directory managers. Plan these authorizations carefully to cover all aspects of your site's VM installation.
- Grant general users authorization to use those commands and menu selections that enable them to manage their own virtual machine. Users can then perform tasks such as maintaining their own system password and controlling access to their minidisks by others.

  For example, for users in the Technical Support group, you may want to authorize them to use all selections on the User Selection Menu. For the users in the Purchasing group, you can prevent them from using certain selections from the User Selection Menu by withholding that authorization.

You give user IDs authorization to use commands by adding GRANT records to the AUTHORIZ CONFIG file. In its simplest form, a GRANT record uses the following format:

```
GRANT authority TO users
```

The *authority* can be as simple as authorization to use an entire command, a command and some of its parameters, or a list of commands. The variable, *users*, is a user ID or list of user IDs to be granted authorization.

Note: For complete format information for the GRANT record, see GRANT Record in the chapter "Configuration File Reference" in the *Reference Guide*.

You deny users authorization to use commands by adding WITHHOLD records to the AUTHORIZ CONFIG file. In its simplest form, a WITHHOLD record uses the following format:

```
WITHHOLD authority FROM users
```

The *authority* can be an authorization to deny use of an entire command, a command and some of its parameters, or a list of commands. The variable, *users*, is a user ID or list of user IDs from which to withhold authorization.

Note: For complete format information for the WITHHOLD record, see WITHHOLD Record in the chapter "Configuration File Reference" in the *Reference Guide*.

Define user ID lists and authority lists that let you authorize many people for many commands in the list by adding LIST records to the AUTHORIZ CONFIG file. A LIST record uses the following format:

```
LIST *listname listitem ...listitem
```

The variable **listname* specifies the name of the list. The asterisk is required. Each *listitem* can be either a user ID or a command. For example, you can create a list of user IDs in the Technical Support department and then create another list that specifies the commands that the Technical Support department is authorized to use.

Note: For complete format information for the LIST record, see LIST Record in the chapter "Configuration File Reference" in the *Reference Guide*.

Granting Authorization to a Command or Utility

Before you grant authority to users to use a command or utility, you need to know what authorizations CA VM:Director requires to perform different functions.

The table Authorization to Use CA VM:Director Commands and Utilities lists the authorizations a user must have to issue each CA VM:Director command, menu selection, and utility. When you find the command or utility to which you want to grant authority, specify the appropriate authorization on a GRANT record.

Examples:

You want to let CARLAT use the ADMIN command and all of its parameters. Do this by specifying the ADMIN command's authorization, ADMIN, on a GRANT record in the AUTHORIZ CONFIG file:

```
GRANT ADMIN TO CARLAT
```

To let OPERATOR use the ABEND command, use this GRANT record:

```
GRANT ABEND TO OPERATOR
```

Granting Authorization to Parts of a Command

You can authorize users so that they can use a command with only selected parameters or options, without giving the users authorization to use all of them.

Examples:

To authorize CARLAT to use only the MANAGERS parameter of the ADMIN command, add the following GRANT record to your AUTHORIZ CONFIG file:

```
GRANT ADMIN MANAGERS TO CARLAT
```

You can authorize both CARLAT and WOODYB to use only the MANAGERS parameter of the ADMIN command on a single GRANT record:

```
GRANT ADMIN MANAGERS TO CARLAT WOODYB
```

Waiving Password Requirements for Commands

Many CA VM:Director commands require users to enter their logon passwords when they issue the command.

Note: For more information about the specific commands, see "Logon Password Prompts" in the chapter "Command Reference" in the *Reference Guide*.

To authorize user IDs to use a command without having to enter their logon passwords, give the user IDs NOPASS authorization.

Note: You must give NOPASS authorization separately from authorization to the command.

Examples:

To authorize WOODYB to use the ASSIGN command without entering his logon password each time he issues the command, add the following two GRANT records to your AUTHORIZ CONFIG file:

```
GRANT ASSIGN TO WOODYB
GRANT NOPASS ASSIGN TO WOODYB
```

To let WOODYB enter any command without entering his password, use the following GRANT record, after first adding GRANT records for the commands:

```
GRANT NOPASS TO WOODYB
```

To authorize WOODYB to use the CA VM:Director GRANT AUTHORITY command for all file spaces without entering his logon password each time, add the following GRANT records to your AUTHORIZ CONFIG file:

```
GRANT GRANT AUTHORITY *.* TO WOODYB
GRANT NOPASS GRANT TO WOODYB
```

To authorize WOODYB to use the CA VM:Director REVOKE AUTHORITY command for all file spaces without entering his logon password each time, add the following GRANT records to your AUTHORIZ CONFIG file:

```
GRANT REVOKE AUTHORITY *.* TO WOODYB
GRANT NOPASS REVOKE TO WOODYB
```

Granting Authorization to Create User IDs

Use the *NEWUSRS predefined list on a GRANT record to allow the issuing user to create new user IDs. We recommend that you give this authorization only to your site's directory managers.

Examples:

To allow CARLAT to use the MANAGE command to create user IDs, add the following GRANT record to the AUTHORIZ CONFIG file:

```
GRANT MANAGE *NEWUSRS TO CARLAT
```

The GRANT record above only lets CARLAT create users. For CARLAT to manage all users on the system, add the following GRANT record:

```
GRANT MANAGE TO CARLAT
```

Note: For more information, see Granting Authorization to a Command or Utility (see page 106).

Granting Authorizations to Use the MAINT MANAGE Command

When you authorize user IDs to use MANAGE subfunctions of the MAINT command, you treat the MAINT MANAGE command as a one–word command (MAINTMAN) and the subfunction as the parameter, using this general format:

```
GRANT MAINTMAN userid TO userids
```

userid

Authority (*what*).

Userids

User IDs (*whom*).

Examples:

To give directory manager CARLAT authority to use the ACTIVATE subfunction on user ID FRASIERC, add this GRANT record to the AUTHORIZ CONFIG file:

```
GRANT MAINTMAN FRASIERC ACTIVATE TO CARLAT
```

As with other authorizations, you can use the predefined variable lists (shown in the Predefined Variable Lists table) in both the authority and the user ID parts of an authorization.

To authorize directory manager CARLAT to use the ACTIVATE subfunction of the MAINT MANAGE command on only the user IDs she manages, add this GRANT record to the AUTHORIZ CONFIG file:

```
GRANT MAINTMAN *DIRUSRS ACTIVATE TO userids
```

***DIRUSRS**

A predefined variable list.

Userids

User IDs (*whom*).

To authorize all directory managers in the Sales group (in the *SALEMGR list) to use the ACTIVATE subfunction of the MAINT MANAGE command on all the user IDs they manage, add this GRANT record to the AUTHORIZ CONFIG file:

```
GRANT MAINTMAN *DIRUSRS ACTIVATE TO *SALEMGR
```

The table in this chapter lists the authorities required to authorize user IDs to use MAINT MANAGE subfunctions.

Granting Authorizations to Use the GRANT AUTHORITY and REVOKE AUTHORITY Commands

When you authorize user IDs to use the CA VM:Director GRANT AUTHORITY and REVOKE AUTHORITY commands, specify the two–word command authorization (GRANT AUTHORITY or REVOKE AUTHORITY) followed by the file space for which users can issue the command. You also specify the user IDs of the users who will be issuing the commands.

Note: The authorization is at the file space level. This means that the user is authorized to grant or revoke authority for all files and directories in the specified file space.

You cannot use the GRANT AUTHORITY or REVOKE AUTHORITY authorizations in LIST records.

The GRANT record for the GRANT AUTHORITY and REVOKE AUTHORITY authorizations uses the following structure:

```
GRANT {GRANT | REVOKE} AUTHORITY filespace TO "Issuer"
```

Issuer:

```
{userid | userid_list}
```

{GRANT | REVOKE} AUTHORITY *filespace*

Authority and file space.

{*userids* | *userid_list*}

Command issuer.

The WITHHOLD record for the GRANT AUTHORITY and REVOKE AUTHORITY authorizations uses the following structure:

```
WITHHOLD {GRANT | REVOKE} AUTHORITY filespace TO "Issuer"
```

Issuer:

```
{userid | userid_list}
```

{GRANT | REVOKE} AUTHORITY *filespace*

Authority and file space.

{*userids* | *userid_list*}

Command issuer.

By default, the command issuer can grant (or revoke) authority for all users. For example, this GRANT record allows directory manager CARLAT to use the CA VM:Director GRANT AUTHORITY command to grant access to the ENG:PROJECT file space for everyone:

```
GRANT GRANT AUTHORITY ENG:PROJECT TO CARLAT
```

Now that CARLAT is authorized, she can give anyone (FRAISERC is used in this example) read access to the ENG:PROJECT.PROJECT1 directory. To do so, CARLAT enters the following command:

```
vmdirect grant authority eng:project.project1 to fraiserc (read
```

Using Predefined Variable Lists

As with other authorizations, you can use the Predefined Variable Lists in the TO or FROM part of the authorization. For example, the *DIRMGRS list represents all of your site's directory managers. To prevent all directory managers from using the CA VM:Director GRANT AUTHORITY command to access data in the ENG:COST file space for all user IDs, add this record to the AUTHORIZ CONFIG file:

```
WITHHOLD GRANT AUTHORITY ENG:COST FROM *DIRMGRS
```

*DIRMGRS is the predefined variable list.

Note: For more information about using predefined variable lists, see Predefined Variable Lists.

Specifying a File Space

When you specify a file space on the authorization, *filespace* uses the following form:

filepool:*userid*

The *filepool* and *userid* can each be one of the following:

- Character string

- Pattern with a trailing *
- Predefined variable list
- Site–defined list represented by a LIST record

Note: The filespace parameter is not a directory ID; a period after the user ID, which usually designates the top-level directory, is not allowed.

Examples:

To allow WOODYB to use the CA VM:Director GRANT AUTHORITY command to grant access to files and directories in the file spaces of the users he manages in file pools ENG, TEST, and FIN, add the following LIST and GRANT records to the AUTHORIZ CONFIG file:

```
LIST *POOLS ENG TEST FIN
GRANT GRANT AUTHORITY *POOLS:*DIRUSRS OF *SELF TO WOODYB
```

LIST *POOLS ENG TEST FIN

Defines a list of file pools.

***POOLS**

Site-defined file pool list.

***DIRUSRS**

Predefined variable list.

***SELF**

Predefined variable list.

To allow CARLAT to grant and revoke authorities in all file pools except the human resources (HR) file pool, add the following records:

```
GRANT GRANT AUTHORITY *:* TO CARLAT
GRANT REVOKE AUTHORITY *:* TO CARLAT
WITHHOLD GRANT AUTHORITY HR:* FROM CARLAT
WITHHOLD REVOKE AUTHORITY HR:* FROM CARLAT
```

Note: For more information, see LIST records in the chapter "Configuration File Reference" in the *Reference Guide*.

Granting Authorizations to Use Commands on Only Some User IDs

You can narrow the scope of an authorization so that it is valid only when the user ID to whom you are granting the authority uses the command on certain user IDs. For example, this GRANT record allows WOODYB to use the EDIT command to change everyone's directory entries:

```
GRANT EDIT TO WOODYB
```

You want to authorize WOODYB to edit only NORMP's directory entry. You can do that by giving WOODYB a narrow authorization for the EDIT command, specifying NORMP as the only user ID on which WOODYB can use the command:

```
GRANT EDIT NORMP TO WOODYB
```

You can include the word OVER to describe these narrow authorizations if it helps you or whoever reads the AUTHORIZ CONFIG file to understand the intent of these records. The word fits between the command and the user ID or user IDs over which this authority is valid.

Example:

Both of the following GRANT records authorize WOODYB to use the EDIT command on NORMP's directory entry:

```
GRANT EDIT NORMP TO WOODYB
GRANT EDIT OVER NORMP TO WOODYB
```

OVER does not change the meaning of the record for CA VM:Director. You can use OVER only on those commands that pertain to a user ID. OVER has no meaning if used in an authorization for a command that pertains to the system or parts of the system. For example, you cannot use OVER when granting authorization to the MAP command.

Working with GRANT AUTHORITY and REVOKE AUTHORITY Authorizations

You can use the optional word OVER to narrow the scope of the authorization so that it is valid only when the user ID to whom you are granting authority uses the command on certain user ID. If you do not specify OVER, CA VM:Director treats the authorization as if you specified OVER *ALL, meaning the target of the command authorization can be anyone.

Examples:

This GRANT record allows WOODYB to grant access to the ENG:MISC file space for all users:

```
GRANT GRANT AUTHORITY ENG:MISC TO WOODYB
```

However, you want to authorize WOODYB to grant access for only FRAISERC. You can do that by giving WOODYB a narrow authorization for the GRANT AUTHORITY command, specifying the word OVER and FRAISERC as the only user ID over which WOODYB can use the command:

```
GRANT GRANT AUTHORITY ENG:MISC OVER FRAISERC TO WOODYB
```

Using the above authorization, WOODYB can give FRAISERC write access to the ENG:MISC.GENERAL directory. WOODYB enters the following command:

```
vmdirect grant authority eng:misc.general to fraiserc (write
```

Specifying User IDs

When specifying the word OVER on the GRANT AUTHORITY and REVOKE AUTHORITY authorizations, use the following forms of *userid* to indicate the user or group of users over which the command issuer can issue the command:

OVER *userid*

For *userid*, you can specify the following values:

- Single user ID
- Nickname
- Predefined variable list
- Site-defined list represented by a LIST record
- Keyword user IDs

Specifying Keyword User IDs

You can also use the PUBLIC and ALL keyword user IDs after the OVER parameter. These keywords can be used only on the GRANT AUTHORITY and REVOKE AUTHORITY command authorizations.

PUBLIC Keyword User ID

You can use the PUBLIC keyword user ID for both the GRANT AUTHORITY and REVOKE AUTHORITY command authorizations. The PUBLIC keyword user ID indicates all user IDs that can connect to the file pool. PUBLIC does not imply a specific user ID.

ALL Keyword User ID

You can use the ALL keyword user ID for only the REVOKE AUTHORITY command authorization. The ALL keyword authorization indicates all users for a file or directory.

Example:1

Authorize WOODYB to use the GRANT AUTHORITY command for the files and directories in the QA:FORMS file space for all users that can connect to that file pool. Add this GRANT record to the CA VM:Director AUTHORIZ CONFIG file:

```
GRANT GRANT AUTHORITY QA:FORMS OVER PUBLIC TO WOODYB
```

Example:2

Authorize WOODYB to use the REVOKE AUTHORITY command to revoke authority from all users for the directories in the QA:FORMS file space. Add this GRANT record to the CA VM:Director AUTHORIZ CONFIG file:

```
GRANT REVOKE AUTHORITY QA:FORMS OVER ALL TO WOODYB
```

Example:3

Allow CARLAT to grant authority to all users who can connect to the TCOM:SPEC and TCOM:DESIGN file spaces. Also, authorize CARLAT to use the REVOKE AUTHORITY command to remove authority from all users for the TCOM:PERSONAL file space. Add the following records to the CA VM:Director AUTHORIZ CONFIG file:

```
GRANT GRANT AUTHORITY TCOM:SPEC OVER PUBLIC TO CARLAT
GRANT GRANT AUTHORITY TCOM:DESIGN OVER PUBLIC TO CARLAT
GRANT REVOKE AUTHORITY TCOM:PERSONAL OVER ALL TO CARLAT
```

Example:4

Allow all directory managers (represented by the *DIRMGRS predefined variable list) to grant access to any users they manage to any of the managed user's file spaces in SYSUSE and HR file pools. Add the following records to the CA VM:Director AUTHORIZ CONFIG file:

```
LIST *FPOOLS SYSUSE HR
GRANT GRANT AUTHORITY *FPOOLS:*DIRUSRS OF *SELF OVER  *DIRUSRS OF
*SELF TO *DIRMGRS
```

Restricting Authorizations

You can restrict authority to use commands by using a combination of GRANT and WITHHOLD records in the AUTHORIZ CONFIG file. In its simplest form, a WITHHOLD record uses the following format:

```
WITHHOLD authority FROM users
```

By default, user IDs are not allowed to use CA VM:Director commands. You may want to grant a user ID the authority to use most parameters on a command but not all of them. For commands with many parameters, it may be easier to grant authorization to the whole command and then restrict authorization to a few of its parameters.

Example:1

The ADMIN command has five parameters: MANAGERS, POOL, PROFILE, SKELETON, and SFSMGRS. You want your directory manager WOODYB to perform all configuration functions except SFS administration. You can give him the following authorizations:

```
GRANT ADMIN MANAGERS TO WOODYB
GRANT ADMIN POOL TO WOODYB
GRANT ADMIN PROFILE TO WOODYB
GRANT ADMIN SKELETON TO WOODYB
```

Or, you can give him authorization to the ADMIN command and then restrict his use of the SFSMGRS parameters:

```
GRANT ADMIN TO WOODYB
WITHHOLD ADMIN SFSMGRS FROM WOODYB
```

In addition to restricting authorization to parts of commands, you can use a WITHHOLD record to restrict authorization to one command, even though it appears in an authority list and you already granted a user ID the authorization to that list. For more information, see User ID Lists and Authority Lists.

You can also use a WITHHOLD record to restrict one or more user IDs from using a command even if you granted authorization to a group of user IDs and the user IDs you want to restrict belong to that group.

Example:2

The following GRANT record authorizes all user IDs to use the USER command (*SELF and *ALL are explained in the table Predefined Variable Lists in this chapter):

```
GRANT USER *SELF TO *ALL
```

Example:3

This next record, in combination with the previous GRANT record, prevents user ID TRAINEE from using the USER command, even though TRAINEE is included in the group *ALL:

```
WITHHOLD USER *SELF FROM TRAINEE
```

Restrictions take precedence over grant authorization (WITHHOLD records take precedence over GRANT records) regardless of the order in which those records appear in the AUTHORIZ CONFIG file. For example, you include a GRANT record to allow user ID WOODYB to use the ADMIN command, and you also include a WITHHOLD record that prevents WOODYB from using the ADMIN command with the SFSMGRS parameter. The authorization that CA VM:Director ultimately recognizes is that WOODYB can use the ADMIN command and all of its parameters except the SFSMGRS parameter. This is true if the records appear in the AUTHORIZ CONFIG file either of these two ways:

```
WITHHOLD ADMIN SFSMGRS FROM WOODYB
.
.
GRANT ADMIN TO WOODYB
```

or

```
GRANT ADMIN TO WOODYB
.
.
WITHHOLD ADMIN SFSMGRS FROM WOODYB
```

User ID Lists and Authority Lists

You can group authorities or user IDs into lists that you can use in GRANT and WITHHOLD records. These lists allow you to create a single GRANT or WITHHOLD record to authorize or restrict groups of users to the same group of commands. A LIST record, which you use to create lists, uses the following format:

```
LIST *listname listitem listitem listitem ...
```

You can create lists of authorities and lists of user IDs for these purposes. A list can contain only one type of element—it cannot include both authorizations and user ID. However, you can create many lists, some for user IDs and some for authorities.

Note: The list name must begin with an asterisk; the rest of the list name can be no longer than seven characters.

Use a comma as a continuation character to continue list items to the next physical line. You can continue a LIST record up to 4,095 characters, including blanks. For example:

```
LIST *SALES1 ABBIE DEBBIE GLORIA SILAS ANDREA,
ARTHUR HARRY JOHN JOSEPH
```

User ID Lists

User ID lists let you group user IDs that will have the same command authorizations. LIST records that group user IDs have the following format:

```
LIST *listname userid userid userid
```

listname

The name of the list and *userid userid userid* are user IDs you want on the list.

Example:

To put BUDDY, BETH, BRUCE, and CARLAT, your publications managers, on one list named *PUBS, add this record to your AUTHORIZ CONFIG file:

```
LIST *PUBS BUDDY BETH BRUCE CARLAT
```

You can then use this list name, *PUBS, in any GRANT or WITHHOLD record where you grant or withhold these four user IDs the authority to certain commands.

You can include a user ID on more than one user ID list if the responsibilities of that user fall into different functional areas.

Authority Lists

Authority lists let you group commands that fit together functionally at your site and that you want to grant or restrict authorization to as a group of activities for several user IDs. LIST records that group commands use the following format:

```
LIST *listname authority authority authority ...
```

listname

The name of the command authority list.

Authority authority authority

Each authority for a CA VM:Director command that you want to be on the list.

Note: You cannot include any command parameters or options in an authority list. Each *authority* represents all versions of a command, and can only be a single word.

Example:

Your publications managers must have the authority to use the ADMIN, EDIT, and QUERY commands. You can group these commands in a single list, named *PUBCMDS, by including this record in your AUTHORIZ CONFIG file:

```
LIST *PUBCMDS ADMIN EDIT QUERY
```

Using a List Name in Another List

You can use a list name as an item in another list. This is true for both user ID lists and authority lists.

Example:

The following LIST record defines a list of 12 user IDs—the 9 that the *SALES1 list names plus JEREMY, LUCAS, and PAUL:

```
LIST *NONTECH JEREMY LUCAS *SALES1 PAUL
```

Predefined Variable Lists

CA VM:Director provides lists that represent dynamic groups of user IDs. The user IDs in these lists vary, depending on your configuration. For example, *DIRMGRS represents all of your site's directory managers. If only CARLAT is a directory manager, that is the only user ID represented by *DIRMGRS. If you later add WOODYB as a directory manager, the variable list *DIRMGRS now represents CARLAT and WOODYB.

You can use these predefined variable lists, shown in the table that follows, in part of a GRANT or a WITHHOLD record and as part of the authorization or as part of the object of the authorization. When they appear in part of the authorization, they narrow the scope of the authorization.

You can use a predefined variable list in any authorization that requires a user ID. Uses are explained in the next section.

Variable List	Entity Specified	Alternate Forms
*	All user IDs	- - -
*ALL	All user IDs	- - -
*ANY	All user IDs	- - -
*DIRMGRS	All user IDs defined as directory managers in the VMDIRECT MANAGERS file	- - -
*DIRUSRS *dirmgr*	All user IDs that *dirmgr* manages (same as *MANAGEE)	*DIRUSRS OF *dirmgr* *dirmgr*'S *DIRUSRS
*MANAGEE *dirmgr*	All user IDs that *dirmgr* manages (same as *DIRUSRS)	*MANAGEE OF *dirmgr* *dirmgr*'S *MANAGEE
*NEWUSRS	All user IDs that do not yet exist (allows them to be created)	- - -

Variable List	Entity Specified	Alternate Forms
*	All user IDs	- - -
*SELF	User ID to whom the authorization on the GRANT record is being given	- - -

The variable lists that specify a variable group of user IDs *that belong to a named user* ID *or group* have alternate forms.

These alternate forms, as shown in the Predefined Variable Lists table, are different only to help show the ownership of the variable group to anyone who reads the GRANT record.

Example:1

The following *user ID phrases* are identical. They all mean "all user IDs that directory manager CARLAT manages:"

```
*DIRUSRS CARLAT
*DIRUSRS OF CARLAT
CARLAT'S *DIRUSRS
```

Example:2

You can include these user ID phrases on GRANT records to authorize all user IDs that directory manager CARLAT manages to use a command or a list of commands.

```
GRANT EDIT TO *DIRUSRS CARLAT
GRANT EDIT TO *DIRUSRS OF CARLAT
GRANT EDIT TO CARLAT'S *DIRUSRS
```

Creating Authorizations with User ID Lists and Authority Lists

When you have created the lists you need, you can use them on GRANT and WITHHOLD records. For example, instead of creating 12 GRANT records—three command authorizations (ADMIN, EDIT, QUERY) for each of the four publications manager user IDs listed in User ID Lists—you create only one. This one record authorizes all user IDs on the *PUBS list to use all commands on the *PUBCMDS list:

```
GRANT *PUBCMDS TO *PUBS
```

You can include a user ID list on a GRANT record authorizing those user IDs to use a single command rather than an authority list. For example, you can authorize all user IDs included in the list *NONTECH (as shown in Using a List Name in Another List) to use the EDIT command:

```
GRANT EDIT TO *NONTECH
```

You can authorize a single user ID to use all the commands in an authority list. For example, you can authorize REBECCAH to use all commands in the *PUBCMDS list, even though she is not a publications manager:

```
GRANT *PUBCMDS TO REBECCAH
```

Creating Authorizations to Use Commands on Only Some Groups of User IDs

You can create authorizations with authority phrases that include predefined variable lists. These authority phrases narrow the scope of the authorization to include particular user IDs. These particular user IDs are *not* the ones granted authority, but are part of the authorization itself.

Example:1

This GRANT record allows the user ID WOODYB to use the ULIST command:

```
GRANT ULIST TO WOODYB
```

ULIST

The authority.

WOODYB

The user IDs for the authority.

In contrast, this GRANT record allows the user ID WOODYB to use the ULIST command, but only on user IDs that directory manager CARLAT manages:

```
GRANT ULIST *DIRUSRS CARLAT TO WOODYB
```

ULIST *DIRUSRS CARLAT

The authority.

WOODYB

The user IDs for the authority.

Example:2

This GRANT record does the same as the previous one, except the word OF was added to help you or whoever reads the AUTHORIZ CONFIG file to understand the content of these records:

```
GRANT ULIST *DIRUSRS OF CARLAT TO WOODYB
```

ULIST *DIRUSRS OF CARLAT

The authority.

WOODYB

The user IDs for the authority.

Example:3

As with authorizing user IDs to use commands on only some user IDs, you can include the words OF and OVER to help describe the intent of these records. These words fit between the command and the user IDs over which this authorization is valid.

Both of the next GRANT records authorize WOODYB to use the ULIST command on user IDs that directory manager CARLAT manages:

```
GRANT ULIST *DIRUSRS CARLAT TO WOODYB
GRANT ULIST OVER *DIRUSRS OF CARLAT TO WOODYB
```

Pattern Matching in Authorizations

Use a trailing asterisk to indicate all words that start with the specified characters. Use this capability with great care because you can inadvertently grant powerful authorizations to many user IDs.

Example:

The following record gives ULIST authorization over the user IDs that user ID DATAMGR manages to all user IDs that start with the character string DEV:

```
GRANT ULIST OVER *DIRUSRS OF DATAMGR TO DEV*
```

Querying Authorizations

CA VM:Director provides two types of information about authorizations:

- All records pertaining to an authorization about which you want information

  You can display every GRANT or WITHHOLD record that has a bearing on an authorization by using the LISTAUTH command. The LISTAUTH command displays the GRANT or WITHHOLD record that takes precedence, followed by all GRANT and WITHHOLD records that match but do not matter in the authorization.

- A user ID's ability to use a command

  You can find out if a user ID can use a command, or a command with a particular parameter, using the MAY command. The MAY command displays the one GRANT or WITHHOLD record that ultimately determines the authorization. The output of the MAY command is a message displayed on the screen and a return code. When used in an EXEC, the return code is not displayed but is used to indicate the result of the authorization query.

Example:

DEBBIE is in the sales directory managers group (list *SALES). The *SALES group is denied the use of the CHGMDISK command through a WITHHOLD record. However, DEBBIE is specifically authorized to use the CHGMDISK command on user IDs in the *MYLIST list by a GRANT record. *MYLIST includes the user ID JIM. Enter the following LISTAUTH command to find all authorizations that affect DEBBIE's ability to use the CHGMDISK command on user ID JIM:

```
vmdirect listauth debbie chgmdisk jim
```

CA VM:Director responds with the following, indicating that two authorizations affect the user IDs and commands you asked about:

```
REJECTED BY: WITHHOLD CHGMDISK FROM *SALES
ACCEPTED BY: GRANT CHGMDISK *MYLIST TO DEBBIE
```

The response shows the authorizations in the AUTHORIZ CONFIG file that mention DEBBIE, the CHGMDISK command, and user ID JIM, with the affected one listed first.

Next, enter the following MAY command to find whether DEBBIE can use the CHGMDISK command on user ID JIM:

```
vmdirect may debbie chgmdisk jim
```

CA VM:Director responds with the following, indicating that DEBBIE cannot use the CHGMDISK command on user ID JIM:

```
REJECTED BY: WITHHOLD CHGMDISK FROM *SALES
```

Special Authorization Queries

To check whether you can use a particular command or what authorizations affect you, use the LISTAUTH and the MAY commands. You also use these commands to check any other user's ability to use a command or all users' ability to use a command.

The LISTAUTH and the MAY commands provide special words to replace the user ID and the authority you are querying:

- Use the words I, ANYUSR (or SOMEUSR), or EVERYUSR to specify the user ID you are querying
- Use the words ANYCMD (or SOMECMD), EVERYCMD, ANYWRD (or SOMEWRD), or EVERYWRD to specify the authority you are querying

Note: For more information about using the special authorization queries available, see LISTAUTH Command and MAY Command in the chapter "Command Reference" in the *Reference Guide*.

Querying Authorizations Given to Lists of User IDs or for Lists of Commands

You can query the authorizations of all user IDs in a list of user IDs and you can query the authorizations user IDs have to all commands in a list of commands.

Example:

Your AUTHORIZ CONFIG file contains the following record, which defines the list *SALES to include user IDs GEORGE, SILAS, and GLORIA:

```
LIST *sales george silas gloria
```

The following command finds all authorizations that allow any user to NOLOG user IDs GEORGE, SILAS, and GLORIA (*every* member of *SALES):

```
vmdirect listauth anyusr nolog *sales
```

Sample Authorizations

Several authorizations are common to every system. The sample AUTHORIZ CONFIG file on the software distribution tape contains many of these common authorizations. Using the information contained in this chapter, you can customize the authorizations in the sample file for your site's needs. The following sections explain many of the common authorizations for the system administrator, directory manager, and general users.

Authorization for a System Administrator

Give the system administrator (VMANAGER) authorization to use all CA VM:Director commands for all users on the system:

```
GRANT * OVER *ALL TO VMANAGER
```

Authorizations for Directory Managers

Directory manager authorizations are usually the most complicated authorizations to give. Often you want to give different kinds of authorization to different directory managers.

We recommend that authorizations for directory managers enable them to use all selections from the Manager Selection Menu on only the user IDs they manage. This authorization is given by adding the following record to the AUTHORIZ CONFIG file:

```
GRANT MANAGE OVER *DIRUSRS OF *SELF TO *DIRMGRS
```

Because selection 3 on the Manager Selection Menu accesses the User Selection Menu, you also need to add the following GRANT record to allow directory managers to use it:

```
GRANT USER OVER *DIRUSRS OF *SELF TO *DIRMGRS
```

CA VM:Director provides authorizations for the Manager Selection Menu and for each of its selections separately. All Manager Selection Menu selections are authorized with the MANAGE authorization. Authorizations to individual selections are listed in the table, Authorization to Use CA VM:Director Commands and Utilities, in this chapter.

The following extended examples will help you create the authorizations you want.

Example:1

Authorizing your directory managers for Sales and for Shipping to use different commands on the user IDs they manage:

1. Create two lists of directory managers for your site.
   - List 1, *SALEMGR, includes directory managers in Sales
   - List 2, *SHIPMGR, includes directory managers in Shipping

     ```
     LIST *SALEMGR SILAS DAN GLORIA ALICIA
     LIST *SHIPMGR SCOTT LONNIE WENDY GARY
     ```

2. Create two lists of command authorities, one for each set of directory managers. The lists include the authorities for commands that each group of directory managers will use most often: *SALECMD for directory managers in Sales, and *SHIPCMD for directory managers in Shipping:

   ```
   LIST *SALECMD CHANGE MANAGE USER ULIST
   LIST *SHIPCMD CHGMDISK MANAGE USER CLASS EXPIRE
   ```

3. Give the directory managers authorization to use the commands over only the users they manage:

   ```
   GRANT *SALECMD OVER *DIRUSRS OF *SELF TO *SALEMGR
   GRANT *SHIPCMD OVER *DIRUSRS OF *SELF TO *SHIPMGR
   ```

   Alternatively, give the directory managers authorization to use the commands for all users:

   ```
   GRANT *SALECMD TO *SALEMGR
   GRANT *SHIPCMD TO *SHIPMGR
   ```

4. Allow the directory managers to create new users:

   ```
   GRANT *NEWUSRS TO *SALEMGR
   GRANT *NEWUSRS TO *SHIPMGR
   ```

5. Prevent DAN, who is authorized by the *SALECMD list, from using Manager Selection Menu selection 6 (moving a minidisk for the user IDs he manages):

   ```
   WITHHOLD MANSEL06 FROM DAN
   ```

6. Let SILAS, one of the directory managers in Sales, use the PASSWORD command over users on the *SHIPMGR list:

   ```
   GRANT PASSWORD OVER *DIRUSRS OF *SHIPMGR TO SILAS
   ```

7. Let SILAS, one of the directory managers in Sales, use the MANAGE command on the user IDs that LONNIE manages:

   ```
   GRANT MANAGE OVER LONNIE'S *MANAGEE TO SILAS
   ```

   or

   ```
   GRANT MANAGE OVER *DIRUSRS OF LONNIE TO SILAS
   ```

8. Allow all directory managers to create new user IDs:

   ```
   GRANT MANAGE OVER *NEWUSRS TO *DIRMGRS
   ```

9. Allow all directory managers to manipulate user IDs they manage:

   ```
   GRANT MANAGE OVER *DIRUSRS OF *SELF TO *DIRMGRS
   ```

Example:2

Giving full authorization to a menu and then withholding individual selections.

You want to authorize each directory manager to perform all management tasks except defining or reactivating a user ID (selection 1; MANSEL01) for any user ID. You want to give MANSEL01 authorization only to MARY:

1. Create a list, *DMAN, of all directory managers, except for MARY, at your site:

   ```
   LIST *DMAN JOHN MARK JANET MIKE PETE ALICE AMY
   ```

2. Give the directory managers and MARY authority to use all selections on the Manager Selection Menu:

   ```
   GRANT MANAGE TO *DMAN MARY
   ```

3. Allow MARY to create new users:

   ```
   GRANT MANAGE *NEWUSRS TO MARY
   ```

4. Withhold selection 1 on the Manager Selection Menu from directory managers *except* for MARY:

   ```
   WITHHOLD MANSEL01 FROM *DMAN
   ```

Example:3

Authorizing each program manager to move minidisks for any user IDs on the system:

1. Put all program managers in the *PRGMRS list:

   ```
   LIST *PRGMRS MAC DAVE JIM
   ```

2. Authorize your program managers to use only selection 6 from the Manager Selection Menu for any user ID:

   ```
   GRANT MANSEL06 OVER *ALL TO *PRGMRS
   ```

Example:4

Authorizing your program managers to use the User Selection Menu and to edit directory comments and review directory entries for any user IDs on the system:

1. Put all commands for program managers (the commands for Manager Selection Menu selections 3, 7, and 8) in one list, *PRGCMD:

   ```
   LIST *PRGCMD MANSEL03 MANSEL07 MANSEL08 USER
   ```

2. Authorize your program managers to use the commands in list *PRGCMD for any user ID:

   ```
   GRANT *PRGCMD OVER *ANY TO *PRGMRS
   ```

   or

   ```
   GRANT *PRGCMD TO *PRGMRS
   ```

Authorization for a SFS Administrator

To designate a user ID as the SFS administrator:

```
GRANT SFSADMIN TO sfsadmin_userid
```

This authorization allows the SFS administrator to perform all functions described in Configuring CA VM:Director for SFS (see page 199).

Authorization for SFS Managers

SFS manager authorization can be a complicated authorization to give because you may want to give different authority to different SFS managers.

In addition to the authorization they already have as directory managers, we recommend that you also give the SFS managers authority to use the DELETE, ENROLL, MODIFY, and MOVE2SFS CA VM:Director SFS commands on only the user IDs they manage.

If you want SFS managers to perform SFS user administration functions using the Manager Selection Menu, you must additionally grant those SFS managers MANAGE, MANAGE 10, or MANSEL10 authorization.

Note: For more information and instructions about designating user IDs as SFS administrators and SFS managers, see Configuring CA VM:Director for SFS (see page 199).

Authorizations for General Users

We recommend that authorizations for general users enable them to use the User Selection Menu to change the configuration of their own virtual machines. You can authorize them to use either all of the menu selections or only some of them.

CA VM:Director provides authorizations for the User Selection Menu and for each of its selections separately. All User Selection Menu selections are authorized with only the USER authorization and let users use those selections on their own user IDs. Authorizations to individual selections for the USER command authorization are listed in the table Authorization to Use CA VM:Director Commands and Utilities.

Example:1

To let CORY and LEE use all selections on the User Selection Menu, add this record to your AUTHORIZ CONFIG file:

```
GRANT USER TO CORY LEE
```

Example:2

To deny LEE the use of menu selection 4, add this record to the AUTHORIZ CONFIG file:

```
WITHHOLD USESEL04 FROM LEE
```

Example:3

To authorize all users to use all selections from the User Selection Menu on themselves, add this record to your AUTHORIZ CONFIG file:

```
GRANT USER TO *ALL
```

Example:4

To prevent all users from changing their distribution codes and account numbers, restrict access to menu selection 9 with this WITHHOLD record:

```
WITHHOLD USESEL09 FROM *ALL
```

Authorizations for Command Processing

Authorizations for command processing are those authorizations that do not relate to any one command. They refer instead to the way processing takes place for different commands.

*

User can use all commands.

***ALL**

User can use all commands.

***ANY**

User can use all commands.

ANYPOOL

User can allocate minidisks in any existing subpool.

ANYSKEL

User can create a user ID using any existing skeleton file.

CACHED

User can change the caching attribute of a minidisk: create or scratch a minidisk on a cached device or change an existing minidisk to or from a cached device.

MOVERO

User can move a minidisk that has read–only links.

Note: When you move a minidisk that a user has linked read–only, CA VM:Director holds the space the minidisk occupies until you use the RECLAIM command against the minidisk. CA VM:Director maintains the minidisk by redefining it to the CA VM:Director directory entry with a virtual address in the range 600 to 6FF. For more information, see the RECLAIM Command in the chapter "Command Reference" in the *Reference Guide*.

NOCOPY

User can move a minidisk without actually moving the data on that disk.

NOFORMAT ADD

User can add a minidisk without formatting it.

NOFORMAT DELETE

User can delete a minidisk without formatting it.

NOFORMAT MOVE

User can move a minidisk without formatting it.

NOFORMAT

Combination of NOFORMAT ADD, NOFORMAT DELETE, and NOFORMAT MOVE authorizations

NOPASS [*command*]

Skips the check of the logon password when a user issues *command*. The user must be separately authorized for the command.

Note: If you use the NOPASS special authorization and do not specify *command*, that user ID has NOPASS authorization for all the commands you authorize that user ID to use.

OVERRIDE NEWUSER

User can override rejection by the NEWUSER exit when creating a user ID, or to override the NEWUSER exit's choice of manager name.

SFSADMIN

User can configure CA VM:Director for SFS.

The following examples show how to use a few of the command processing authorizations.

Example:1

Authorize users in the SYSPROGS list to issue any command:

```
LIST *SYSPROGS MAINT VMANAGER JOE
GRANT *ALL TO SYSPROGS
```

Example:2

Let the directory managers enter any CA VM:Director command without being prompted for their logon password:

```
GRANT NOPASS TO *DIRMGRS
```

Example:3

Allow MAINT to allocate minidisks in any existing subpool:

```
GRANT ANYPOOL TO MAINT
```

Authorization to Use CA VM:Director Commands and Utilities

The following table describes the authorization a user needs to use CA VM:Director commands and utilities.

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
ABEND	Terminate CA VM:Director operation abnormally	ABEND	
ADDENTRY*	Create a directory entry for a user or profile from an input file or a skeleton file	ADDENTRY	[*entry*]
	Create a user ID or profile from an input file	ADDENTRY *entry* NOSKEL**	
	If the input file creates a minidisk	ADDMDISK	[*entry*]
	Create a user ID or profile with a skeleton file	ADDENTRY *entry* SKELETON**	
	If the skeleton file creates a minidisk	ADDMDISK	[*entry*]
ADDMDISK*	Add a minidisk for a user ID	ADDMDISK	[*entry*]

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
ADMIN	Use all parameters on the ADMIN command	ADMIN	
	Edit the VMDIRECT MANAGERS file	ADMIN MANAGERS**	
	Edit the VMDIRECT GLOBALS file	ADMIN GLOBALS**	
	Edit a subpool entry	ADMIN POOL**	[*poolid*]
	Edit the VMDIRECT POSIX file	ADMIN POSIX**	
	Edit a directory profile	ADMIN PROFILE**	[*profile*]
	Edit a skeleton file	ADMIN SKELETON**	[*skeleton*]
	Define or change SFS managers' enrollment defaults or enrollment limits	ADMIN SFSMGRS**	[*userid*]
ASSIGN	Assign a user ID to a different manager	ASSIGN	[*entry* [*mgrid*]]
AUDITEXT	Extract current audit information	AUDITEXT	
CHANGE	Change a user ID's name	CHANGE	[*entry*]
CHGENTRY	Change a USER to an IDENTITY or change an IDENTITY to a USER	CHGENTRY	[*entry*]
CHGMDISK*	Move or change a minidisk	CHGMDISK	[*entry*]
	With the NOCOPY option	NOCOPY	
CHGVOLNM	Change all references to the volser of any DASD volume controlled by CA VM:Director	CHGVOLNM	[*oldvolser*]
CLASS	Assign a CP privilege class	CLASS	[*class*]
CMD	Use the CMD command to route another command to an Agent product server in a Single System Image environment	CMD	

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
CMS	Execute a CMS or CP command on the CA VM:Director service virtual machine	CMS	[*word1 ...word15*]
COMPRESS	Defragment disk storage	COMPRESS	[*volser*]
CONFIG	Edit the CA VM:Director configuration files	CONFIG	
	Edit the AUTHORIZ CONFIG file	CONFIG AUTHORIZ**	
	Edit the DASD CONFIG file	CONFIG DASD**	
	Edit the PRODUCT CONFIG file	CONFIG PRODUCT**	
	Edit the SECURITY CONFIG file	CONFIG SECURITY**	
	Edit the CA VM:Director SFS configuration	CONFIG SFS**	
CPFMTXA	Use CPFMTXA command to change allocation on the object directory volume	CPFMTXA	
DELENTRY*	Delete an existing user ID or profile	DELENTRY	[*entry*]
	Delete a minidisk for a deleted user ID	DELMDISK	[*entry*]
DELETE	Delete file space for an active user ID	DELETE	[*userid*]
DELMDISK*	Delete a user ID's minidisk	DELMDISK	[*entry*]
DISPLINK	Display links to a user's minidisks	DISPLINK	[*userid*]
DUPENTRY	Create a new user ID based off an existing user ID	DUPENTRY	[existing [new]]
	For the template user ID's minidisks.	DUPMDISK	[*entry*]
	For the new user ID's minidisks.	ADDMDISK	[*entry*]
	No formatting of minidisks.	NOFORMAT	[*entry*[*mgrid*]]

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
	To use the MANAGER option.	MANAGER	
DUPMDISK*	Create an exact duplicate of an existing minidisk	DUPMDISK	[*entry*]
	For the user ID that owns the source minidisk (*sourceuser*)	DUPMDISK	[*entry*]
	For the user ID that owns the target minidisk (*targetuser*)	ADDMDISK	[*entry*]
EDIT	Edit a user ID's directory entry	EDIT	[*entry*]
EDX	Edit a user ID's directory entry, expanding any INCLUDE statement	EDIT	[*entry*]
END	Terminate CA CA VM:Director immediately or after current processes complete	END	
	Terminate CA VM:Director immediately only	END FORCE**	
	Terminate CA VM:Director operation only after current processes complete	END NOFORCE**	
ENROLL	Enroll a user ID into an SFS file pool	ENROLL	
ENTRY	Update or query directory entry contents	ENTRY	[*entry* [*subcommand*]]
EXPIRE	Expire a user ID's logon password	EXPIRE	[*userid*]
EXTRACT	Extract directory information	EXTRACT	
GENINCL	Add an INCLUDE statement to a userid's directory entry	GENINCL	[*userid* [*profile*]]
GETENTRY*	Retrieve current copy of a user ID's directory entry or a directory profile	GETENTRY	[*entry*]

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
GETPWEXP	Display user ID password expiration information	GETPWEXP	[*userid*]
GRANT AUTHORITY	Allow a user to grant access to a file space for other users	GRANT AUTHORITY *filespace*	[*userid*]
JOURNAL	Display password violations and reset password violation count	JOURNAL	
	Display password violations	JOURNAL LIST**	[*word1 …word4*]
	Reset a password violation count to zero	JOURNAL RESET**	[*word1 …word4*]
LISTAUTH	Query the authorizations specified in the AUTHORIZ CONFIG file	LISTAUTH	[*userid* [*authwrds*]]
LOCK*	Prevent updates to any object	LOCK	
	Prevent updates to a CMS file	LOCK FILE**	[*fname* [*ftype* [*fmode*]]
	Prevent updates to a user ID	LOCK USER**	[*userid*]
	Prevent updates to a profile	LOCK PROFILE**	[*profid*]
MACLOAD	Load a macro to the CA VM:Director service virtual machine	MACLOAD	
MAINT	Perform line-mode user functions	MAINT	[*subfunction*]
	Perform line-mode management functions	MAINTMAN	[*entry* [*subfunction*]]
	Perform line-mode user functions for another user ID	MAINTMAN *entry* USER**	[*subfunction*]
MANAGE	Use all selections, 1 through 10, on the Manager Selection Menu (selections withheld are shown as ***not available***.)	MANAGE	[*entry*]
	Create user IDs (part of selection 1)	MANAGE *NEWUSRS	

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
	Use menu selection 1	MANSEL01	[*entry*]
	Create user IDs (part of selection 1)	MANSEL01 *NEWUSRS	
	Use menu selection 2	MANSEL02	[*entry*]
	Use menu selection 3	MANSEL03 USER	[*entry*]
	Use menu selection 4	MANSEL04	[*entry*]
	Use menu selection 5	MANSEL05	[*entry*]
	Use menu selection 6	MANSEL06	[*entry*]
	Use menu selection 7	MANSEL07	[*entry*]
	Use menu selection 8	MANSEL08	[*entry*]
	Use menu selection 9	MANSEL09	[*entry*]
	Use menu selection 10	MANSEL10	[*entry*]
MAP	Map a volume	MAP	[*volume* [*parameters*]]
MAY	Query the authorizations specified in the AUTHORIZ CONFIG file	MAY	[*entry* [*authwrds*]]
MDSKSCAN	Scan a user ID's minidisks	MDSKSCAN	[*entry*]
MODIFY	Modify the SFS allocation for a user ID	MODIFY	[*userid*]
MOVE2SFS	Copy data from a minidisk to SFS	MOVE2SFS MANAGE SFSADMIN	[*userid*]
MULTIPLE	Perform user ID maintenance on several user IDs at the same time	MULTIPLE	
	Create several user IDs at the same time	MULTIPLE NEWUSER**	
	Remove several user IDs at the same time	MULTIPLE REMOVE**	[*entry*]
	Place several user IDs on hold at the same time	MULTIPLE HOLD**	[*entry*]

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
	Reactivate several held user IDs at the same time	MULTIPLE ACTIVATE**	[*entry*]
NEWIPL	Change an IPL system name or device in all directory entries to a new IPL system name or device	NEWIPL	
NOLOG	Change a user ID's password to NOLOG	NOLOG	[*userid*]
OVERRIDE	Alter privilege classes without shutting down CA VM:Director	CPOVERID	
PAINT	Change a CA VM:Director screen	PAINT	[*screen*]
PASSWORD	Set passwords for a user ID	PASSWORD	
	Set only randomly generated passwords for a user ID	PASSWORD *userid* RANDOM**	
	Set only a specific password for a user ID	PASSWORD *userid* SPECIFIC**	
QLOCK	Display all CA VM:Director locks	QLOCK	
QPCB	List active CA VM:Director processes	QPCB	
QSTART	Display the time CA VM:Director was most recently started	QSTART	
QUERY		QUERY	[*userid*]
		QUERY ACCOUNT**	[*mgrid*]
	Display information about a manager's allocation space	QUERY ALLOC**	[*mgrid*]
	Display a user ID's privilege class	QUERY CLASS**	[*userid*]
	List the names of the file pools that CA VM:Director manages	QUERY FILEPOOL**	
	List user IDs on hold	QUERY HOLD**	[*userid*]

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
	List user IDs that are directory managers	QUERY MANAGERS**	
	List user IDs whose passwords have not changed for a specified number of days	QUERY PASSWORD**	
	List user IDs' directory entries that include a directory profile	QUERY PRFUSERS**	[*profile*]
	Display status information about the Servant Facility	QUERY SERVANT**	
	List the names of the file pools and user storage groups from which you can allocate file space	QUERY SFS**	
	List the skeleton files that a manager can use	QUERY SKELETON**	[*mgrid*]
	List the subpools that a manager can use	QUERY SUBPOOLS**	[*mgrid*]
	List the user IDs that a manager manages	QUERY USERS**	[*entry*]
	Display the CA VM:Director release level	QUERY VERSION**	
	Determine status of long-running commands	QUERY WORKUNIT**	[*userid*]
REBUILD	Condenses and defragments the CP object directory	REBUILD	[*userid*]
RECLAIM	Reclaim DASD space from MOVERO minidisks	RECLAIM	
REPENTRY*	Replace a directory entry or directory profile	REPENTRY	[*entry*]
	If the new entry adds a minidisk	ADDMDISK	[*entry*]
	If the new entry changes a minidisk	CHGMDISK	[*entry*]

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
	If the new entry deletes a minidisk	DELMDISK	[*entry*]
	Reset any password violation count	RESET	
RESET	Reset password violation counts for a user that occurred while verifying the password	RESET USERPASS**	[*userid*]
	Reset password violation counts that occurred while trying to create a directory link	RESET VMXLINK**	[*userid* [*userid* [*vaddr*]]
REVOKE AUTHORITY	Allow a user to revoke access to a file space for other users	REVOKE AUTHORITY *filespace*	[*userid*]
SUBCONFIG	None for menu access Processing of SUBCONFIGs controlled by USER authorization	See USER	
TAKEOVER	Force an AGENT server to become the MASTER	TAKEOVER	
TRACE	Trace execution of a CA VM:Director macro	TRACE	[*parameters*]
TRANSFER	Transfer a minidisk from one user ID to another	TRANSFER	[*entry* [*newowner*]]
ULIST	Display information about user IDs	ULIST	[*entry*]
UNLOCK*	Remove a CMS file, profile, or user ID lock	UNLOCK	

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
	Remove a lock from a CMS file	UNLOCK FILE**	[*fname* [*ftype* [*fmode*]]]
	Remove a lock from a profile	UNLOCK PROFILE**	[*profid*]
	Remove a lock from a user ID	UNLOCK USER**	[*userid*]
USER	Use all selections, 1 through 11, on the User Selection Menu (selections withheld are displayed as ***not available***)	USER	[*userid*]
	Use menu selection 1	USESEL01	[*userid*]
	Use menu selection 2	USESEL02	[*userid*]
	Use menu selection 3	USESEL03	[*userid*]
	Use menu selection 4	USESEL04	[*userid*]
	Use menu selection 5	USESEL05	[*userid*]
	Use menu selection 6	USESEL06	[*userid*]
	Use menu selection 7	USESEL07	[*userid*]
	Use menu selection 8	USESEL08	[*userid*]
	Use menu selection 9	USESEL09	[*userid*]
	Use menu selection 10	USESEL10	[*userid*]
	Use menu selection 11	USESEL11	[*userid*]
VMXBKP01	Create a USER DIRECT file representing a copy of the CA VM:Director directory database	BACKUP MAY	[*userid* [*authority*]]
VMXBKP02	Create a backup copy of the CA VM:Director directory database using DDR	BACKUP	
VMXBKP03	Create a backup copy of the CA VM:Director directory database using COPYFILE	BACKUP	

Command or Utility	Type of Authorization	Authority	Optional Parameters to Narrow Authority
VMXGNR	Generate the CA VM:Director directory database and converts your CP source directory file into CA VM:Director database format.	None required	
VMXIPL	Write an IPLable program on the IPLDISK minidisk; when the program is initialized, it interacts with CA VM:Director to update a user ID's logon password	None required	
VMXSRB	Generate a report of all audit data captured by CA VM:Director	None required	

* Part of the Application Programming Interface

** Cannot be used in LIST records

Chapter 8: Managing Directory Entries

Many aspects of managing directory entries can be performed by your directory managers, other system administrators and by you.

This chapter explains how to:

- Create directory entries with a Skeleton.

  Note: For information about skeleton files, see Skeleton Files (see page 165).

- Remove directory entries.
- Allow directory managers to create directory entries.
- Manage inactive or seldom used directory entries.

This section contains the following topics:

Creating Directory Entries (see page 143)
Allowing Directory Managers to Create Directory Entries (see page 148)
Checking User ID Password Expiration (see page 149)
Managing Inactive Directory Entries (see page 150)
Removing Directory Entries (see page 155)

Creating Directory Entries

CA VM:Director provides you with five commands to create directory entries. Each has advantages in different situations.

Method	Advantages
ADDENTRY command (line–mode)	No prompts. You enter all information in one command, making ADDENTRY useful for creating directory entries through programming applications.
DUPENTRY command (line-mode)	Lets you create a user ID by basing it on another user ID, without having to use a skeleton file. All minidisks in the original entry will be duplicated in the new entry.
EDIT command (prototype)	Lets you create a user ID by basing it on another user ID, without having to use a skeleton file.

Method	Advantages
MAINT command (line–mode)	Usually faster than full–screen methods. You enter directory entry attributes one-at-a-time in response to screen prompts but without waiting for full screens to change.
MANAGE command (full–screen)	Usually the easiest method for creating USER directory entries. You follow the instructions on the screens and fill in the blanks to define the attributes of the entry you are creating.
MULTIPLE command (control file)	Lets you use one command and a control file to create many directory entries. This method automates the process for large organizations.

Directory entry names must have one- to eight-characters and must be valid CMS file names.

Valid characters are:

- Letters: A–Z
- Digits: 0–9
- Symbols: $ + – (hyphen) _ (underscore)

The following *cannot* be directory entry names:

- **Words** including SYSTEM, USER, ALL, CMSUT3, SYSTEMMP, POSIX.
- **CA VM:Operator console names**. If you use the CA VM:Operator product, you should not have user IDs with the same names as CA VM:Operator consoles. The CONSOLE records in the VMOPER CONFIG file list the console names.

  Note: For more information about the VMOPER CONFIG file, see the *CA VM:Operator System Administration Guide*.
- Any other words listed as not recommended in the appropriate IBM planning and administration documentation.

Creating Directory Entries Through a Programming Application

To create a directory entry using a programming application, use the ADDENTRY command. It operates in line-mode and requires no prompts, so that it can be issued by a CMS program. Use one of the following forms of the command:

- Replace *name* with the name of the directory entry to create using the default skeleton file:

  `VMDIRECT ADDENTRY name`

- Replace name with the name of the directory entry to create and skelname with the name of a skeleton file with the new entry characteristics:

  `VMDIRECT ADDENTRY name skelname [(options]`

- Replace name with the name of the directory entry to create and filetype [filemode] with the type and mode of an existing directory entry used as a pattern for creating the new directory entry:

  `VMDIRECT ADDENTRY name filetype [filemode] (NOSKEL`

If you want to specify other options, such as an account number or a distribution code, include those options on the command line. They are explained fully under ADDENTRY Command in the *Reference Guide*.

Creating a User ID Based on Another User ID

If an existing user ID has most or all of the features that you want to assign to a new user ID, you can create the user ID based on the existing user ID. This method uses an existing user ID as a prototype to create a user ID.

To create a user ID based on another user ID

1. Enter the following command, replacing *userid* with the name of the user ID you want to create and *prototype* with the name of the existing user ID you want to use as the prototype:

   `VMDIRECT EDIT userid prototype`

   CA VM:Director copies the directory entry for the prototype user ID to a directory entry for the new user ID, removing all the original MDISK statements and passwords in the process, and places you in an XEDIT session with the new user ID.

2. Make any changes, other than creating minidisks for this new user ID that you need to for the new user ID.

   Note: Do not create minidisks when you use the EDIT command to create user IDs, unless those minidisks are ignored in the DASD CONFIG file or are on a volume that CA VM:Director does not control. To create minidisks for the user IDs, use the CA VM:Director MANAGE command.

3. Save the new directory entry and close the file.

CA VM:Director displays a message that it has updated the directory to include the new user ID.

Creating a Directory Entry by Answering Screen Prompts

If you prefer to create directory entries by answering screen prompts to provide information about the new directory entry, follow these steps:

1. Enter the following command, replacing *entry* with the name of the directory entry you are creating:

   `VMDIRECT MAINT MANAGE entry NEWUSER`

   CA VM:Director gives you a message that it is performing maintenance on the user ID.

   It next prompts you for a skeleton file.

2. Enter the name of the skeleton file to use. To use the default skeleton file, press Enter without any name. (The default skeleton file is the first skeleton file that your SKELETON record lists in the VMDIRECT MANAGERS file.)

   CA VM:Director prompts you for the account number.

3. Enter the account number under which this user ID will run. To use the default account number, press Enter without entering an account number. (The default account number is specified in the skeleton file on which this user ID is being created.)

   If you always use the default account number (from the skeleton file), all your users will have the same account number.

   CA VM:Director next prompts you for the distribution code.

4. Enter the distribution code under which this user ID will run. To use the default distribution code, press Enter without entering a distribution code. (The default distribution code is specified in the skeleton file on which this user ID is being created.)

   If you always use the default distribution code (from the skeleton file), all your users will have the same distribution code.

5. Enter the following:

   end

   CA VM:Director gives you the following message:

   `DIRECTORY UPDATED ONLINE.`

   This completes the creation of a user ID using screen prompts.

Creating a User ID by Filling in the Blanks on a Screen

You can create directory entries by filling in a screen. To do so, follow these steps:

1. Enter the following command, replacing *userid* with the name of the user ID that you are creating:

   `vmdirect manage userid`

   The Manager Selection Menu appears. Only one selection, Define/Reactivate a Userid, is available. Ignore the For Device field, because you are creating a user ID, not a minidisk.

2. Enter **1** in the Selection field.

   The next screen prompts you for the name of a skeleton file to use for creating this user ID. CA VM:Director lists the default, which is the first skeleton file listed on your SKELETON record in the VMDIRECT MANAGERS file.

3. To use a different skeleton file, type its name in the Skeleton Name field. To use the default, skip this field.

4. Type the following information in the appropriate fields:

   a. In the Account Number field, enter the account number to which to assign this user ID, or leave it blank to assign it to your default account number. (The default account number is specified in the skeleton file on which this user ID is being created.)

   b. In the Distribution Code field, enter the distribution code to which to assign this user ID, or leave it blank to assign it to your default distribution code. (The default distribution code is specified in the skeleton file on which this user ID is being created.)

5. Press Enter.

   The Manager Selection Menu appears.

6. Press Enter.

   CA VM:Director displays a message that the directory was updated to include the new user ID.

Creating Several Directory Entries at the Same Time

To be able to create many directory entries at the same time, you must build a control file. In the control file, you will add the name of each user ID and any other specifications that you do not want to default. You then use the MULTIPLE command and that control file to create the directory entries listed in the control file.

Large organizations with many users, such as universities, use this method to create directory entries easily.

To create several directory entries at one time with this method

1. Create a control file.

   Note: For information about how to create a control file, see Control File Records in MULTIPLE Command in the chapter "Command Reference" in the *Reference Guide*.

2. Enter the MULTIPLE command with the NEWUSER parameter and the CMS file name and file type of the control file. For example, if you created a control file named USERS with a filetype of FILE, enter the following command:

   ```
   vmdirect multiple newuser users file
   ```

Allowing Directory Managers to Create Directory Entries

You must include a specific or general GRANT record in the AUTHORIZ CONFIG file for each directory manager to whom you are giving the ability to create directory entries. If it is a specific GRANT record, it grants authority only to create directory entries. If it is a general GRANT record, it can grant authority that is more general.

Note: The GRANT record you need to include depends on the method of creating new directory entries that you want your directory managers to use.

In addition to the authorities listed in the following table, you also need to include the *NEWUSRS authorization on a GRANT record to allow your site's directory managers to create directory entries. For more information about using the *NEWUSRS authorization, see Granting Authorization to Create User ID (see page 108).

Note: For more information about the GRANT record and how to use it, see GRANT Record in the chapter "Configuration File Reference" in the *Reference Guide*.

Command to Create User ID	Type of Authority	GRANT Record Authority Requirements for a Directory Manager
ADDENTRY	Create directory entries with a skeleton	ADDENTRY SKELETON *NEWUSRS‡

Command to Create User ID	Type of Authority	GRANT Record Authority Requirements for a Directory Manager
	Create directory entries without a skeleton	ADDENTRY NOSKEL *NEWUSRS‡
EDIT†	Create user IDs only	EDIT *NEWUSRS
	General manage	EDIT *NEWUSRS
MAINT	Create directory entries only	MAINTMAN NEWUSER *NEWUSRS‡
	General manage	MAINTMAN *NEWUSRS
MANAGE	Create directory entries only	MANSEL01 *NEWUSRS
	General manage	MANAGE *NEWUSRS
MULTIPLE	Create directory entries only	MULTIPLE NEWUSER *NEWUSRS‡
	General manage	MULTIPLE *NEWUSRS

† It is not recommended that directory managers generally be authorized to use this command.
‡ Must be specified on its own GRANT record.

Checking User ID Password Expiration

To check when the password for a particular user ID will expire, use the GETPWEXP command. You can use this command in conjunction with the strategies for managing passwords as described in Managing Inactive Directory Entries (see page 150).

The GETPWEXP command reports the following types of information about a user ID's password expiration:

- Whether a password is set for no expiration
- The date and the number of days until a password will expire
- The date and the number of days since a password expired

You may want to give all users authority to use this command so that they can manage their own passwords, changing the passwords before they expire. Additionally, you can use the GETPWEXP command in an EXEC to automatically report information about user ID password expiration.

Note: For information about using this command and the contents of its display, see GETPWEXP Command in the chapter "Command Reference" in the *Reference Guide*.

Managing Inactive Directory Entries

An inactive user ID is a user ID that a user has not logged on in a specified amount of time. The AUTOEXP record determines the amount of time. The actual time over which a user has not logged on, or changed passwords, is tracked through its *LA=, *LL=, and *PW= special comments. CA VM:Director provides a number of tools to help you identify and manage inactive user IDs.

Consider the following to manage inactive userids:

1. Expire logon passwords automatically after a known amount of time.

   Doing so enables you to easily spot those user IDs that have not logged on in a while. For example, if you expire logon passwords after 90 days of inactivity, you can use the ULIST command to show the user IDs whose passwords are expired and when their passwords were last changed. A user ID whose password is expired and whose password was last changed 185 days ago has been inactive for a long time.

   You can use an IPLDISK procedure to expire passwords automatically.

2. Decide how long user IDs must remain unused to be considered inactive.
3. Define stages of inactivity and decide what will happen to user IDs at each stage of inactivity.

   For example, you can disable user IDs that have not logged on for four weeks. You can put on hold user IDs that have been inactive for at least four weeks, and if they remain inactive for the next six months, you may want to delete them.

Note: For information about the special comments, see the chapter Special Comment Reference (see page 233).

Expiring a Password Automatically

(CA VM:Secure only) You can automatically expire passwords with an IPLDISK procedure or with the Rules Facility. For more information about using the Rules Facility to automatically expire passwords, see the *Rules Facility Guide*.

(CA VM:Director only) You can automatically expire passwords with an IPLDISK procedure.

To use an IPLDISK procedure to expire passwords automatically, verify that all user ID directory entries contain a password history special comment (*PW=).

Follow these steps:

1. Create an IPLDISK minidisk by adding a minidisk to the CA VM:Director directory entry. Give the minidisk a label of IPLDSK. The default virtual address is 1B3. Only one cylinder is needed.
2. Write an IPLable program to the IPLDISK by using the VMXIPL utility.
3. Add an IPLDISK record to the PRODUCT CONFIG file.
4. Decide the number of days a user ID must be inactive before its password expires.
5. Create a CMS EXEC that lists the user IDs that have been inactive for at least that many days. These are the user IDs whose passwords you want to expire.

   To do so, enter the following command, replacing 30 with the number of days of inactivity after which an inactive user ID's password expires:

   ```
   vmdirect query password 30 (exec
   ```

6. Expire the passwords of the user IDs listed in the CMS EXEC by entering the following command:

   ```
   cms vmdirect expire
   ```

After you have expired the passwords of user IDs that seem to be inactive, check their status to see how long they remain with expired passwords.

Querying Inactive User IDs

After you expire logon passwords, use the ULIST command to identify inactive user IDs. The ULIST command shows the status of user ID passwords, such as those expired, and how many days have gone by since the user ID's password was changed. This can tell you if a user ID was temporarily inactive or may be permanently inactive. For example, a user ID that changed its expired password within two days of expiration was probably temporarily not used, while a user ID that has an expired password from 86 days ago may really be inactive.

Note: For information about using this command and the contents of its display, see ULIST Command in the chapter "Command Reference" in the *Reference Guide*.

Disabling User IDs

You can prohibit all logons to a user ID by changing its password to NOLOG. You may want to disable a user ID this way as an interim step between expiring its password because it is inactive and deleting it from the system.

User IDs can link to the minidisks of a user ID that is disabled in this manner, even though the users of these user IDs cannot log on except by having you re–enable their user IDs. You can determine whether a user ID is truly inactive by whether you are asked to reenable it.

To disable logons to a user ID by changing their passwords to NOLOG, enter the following command, replacing the user ID *userid* with the user ID that you want to disable:

```
vmdirect nolog userid
```

Re-enabling Disabled User IDs

A user whose user ID is disabled may decide to use that user ID again. The only way this can happen is for the user to ask you to reenable the user ID. You can do so by changing its password to something other than NOLOG.

You can take advantage of the fact that users must ask you to re-enable disabled user IDs. Their requests give you an easy way to track which user IDs are actually active, even though they may not be used often, and which ones are truly inactive.

To re-enable a disabled user ID, either use the PASSWORD command or follow these steps:

1. Enter the following command, replacing *userid* with the disabled user ID that you want to re-enable:

   ```
   vmdirect manage userid
   ```

2. Choose selection 3, Select User Menu, and press Enter.
3. Choose selection 1, Logon Storage Size and Password, and press Enter.
4. In the Password field, type a new password for the user ID.
5. Press Enter.

   CA VM:Director asks you to verify the password by entering the password again.

6. Enter the password again and press Enter.

   CA VM:Director takes you back to the User Selection Menu.

7. Press Enter to return to the Manager Selection Menu.
8. Press Enter to update the directory.

Putting User IDs on Hold

You can prohibit all access to a user ID by putting it on hold. A user ID on hold no longer appears in the source or object directory, but CA VM:Director preserves its minidisks and all the data they contain. Users cannot log on to a user ID on hold or link to the minidisks of a user ID on hold.

You may want to put a user ID on hold as a last step before deleting it from the system.

CA VM:Director provides three ways to place a user ID on hold. Each has advantages in different situations:

Method	Advantages
MAINT command (line–mode)	Usually faster than full–screen methods. You enter user ID status on one command line without waiting for screens to change.
MANAGE command (full–screen)	Usually the easiest method for putting user IDs on hold. You follow the instructions on the screens to hold the user ID.
MULTIPLE command (control file)	Lets you use one command and a control file to put many user IDs on hold at the same time. This method automates the process for large organizations.

Placing User IDs on Hold Using a Command

To put a user ID on hold, enter the following command, replacing *userid* with the name of the user ID that you want to put on hold:

```
vmdirect maint manage userid hold
```

Placing User IDs on Hold by Filling in the Blanks on a Screen

If you prefer to fill in blanks on a screen to do your work, you can place a user ID on hold this way. To do so, follow these steps:

1. Enter the following command, replacing *userid* with the name of the user ID that you want to place on hold:

   ```
   vmdirect manage userid
   ```

2. On the Manager Selection Menu, type 5 for Hold or Remove a Userid. Press Enter.
3. On the next screen, indicate that you want to place the user on hold.
4. When you press Enter, you are immediately returned to CMS.

Putting Several User IDs on Hold at the Same Time

To place many user IDs on hold at the same time, you must build a control file. In the control file, you must add the name of each user ID that you want placed on hold. Next, use the MULTIPLE command to hold the user IDs listed in the control file.

Large organizations with many users, such as universities, use this method to hold user IDs easily.

To hold several user IDs at one time with this method

1. Create a control file.

   Note: For information about how to create a control file, see Control File Records in MULTIPLE Command in the chapter "Command Reference" in the *Reference Guide*.

2. Enter the MULTIPLE HOLD command with the CMS file name and file type of the control file. For example, if you have a control file named USERS with the filetype FILE, enter the following command:

   ```
   vmdirect multiple hold users file
   ```

Reactivating User IDs

A user whose user ID is put on hold may decide to use that user ID again. The only way this can happen is for the user to ask you to reactivate the user ID.

You can take advantage of the fact that users must ask you to reactivate held user IDs. Their requests give you an easy way to track which user IDs are active, even though they may not be used often, and which ones are inactive.

CA VM:Director provides three ways to reactivate a user ID on hold. Each has advantages in different situations.

Method	Advantages
MAINT command (line–mode)	Usually faster than full–screen methods. You enter user ID status on one command line without waiting for screens to change.
MANAGE command (full–screen)	Usually the easiest method for reactivating user IDs. You follow the instructions on the screens to reactivate the user ID.
MULTIPLE command (control file)	Lets you use one command and a control file to reactivate many user IDs. This method automates the process for large organizations.

Reactivating User IDs by a Command

To reactivate a user ID on hold, enter the following command, replacing *userid* with the name of the user ID that you want to reactivate:

```
vmdirect maint manage userid activate
```

Reactivating User IDs by Filling in the Blanks on a Screen

To reactivate the user ID with the full–screen method

1. Enter the following command, replacing *userid* with the name of the user ID that you want to reactivate:

   ```
   vmdirect manage userid
   ```

2. On the Manager Selection Menu, type 1 for Define/Reactivate a Userid.
3. On the next screen, indicate that you want to reactivate the user ID.

When the user ID is reactivated, you are returned to the Manager Selection Menu. From there, you can perform other management functions or return to CMS.

Reactivating Several User IDs at the Same Time

You can reactivate several user IDs on hold at the same time by using a control file and the MULTIPLE command. You may even want to use the same control file to reactivate user IDs that you used to put them on hold previously.

Though the control file is the same for reactivating user IDs as for putting them on hold, the command is different. Enter the MULTIPLE ACTIVATE command to activate user IDs on hold, with the filename and filetype of the control file. For example, to reactivate user IDs in the file USERS, filetype FILE, enter the following command:

```
vmdirect multiple activate users file
```

Removing Directory Entries

After you have decided that a directory entry is no longer needed, you can remove it from your system. CA VM:Director provides four ways to remove a directory entry. Each has advantages in different situations.

Method	Advantages
MAINT command (line-mode)	Usually faster than full-screen methods. You enter parameters on one command line without waiting for screens to change.

Method	Advantages
DELENTRY command (line-mode)	No prompts. You enter all information in one command line, making DELENTRY useful for removing directory entries through programming applications.
MANAGE command (full-screen)	Usually the easiest method for deleting directory entries. You follow the instructions on the screens to remove the entry.
MULTIPLE command (control file)	Lets you use one command and a control file to remove many directory entries. This method automates the process for large organizations.

Removing Directory Entries by an Interactive Command

To remove a directory entry, verify that it is not on hold, then enter the MAINT MANAGE command with the name of the directory entry that you want to remove and the REMOVE parameter. For example, to remove the user ID TESTID, enter the following command:

```
vmdirect maint manage testid remove
```

If a user ID you want to remove is on hold, reactivate it first by following the procedure explained in Reactivating User IDs by a Command (see page 155).

Removing Directory Entries Through a Programming Application

To build a line-mode command to remove a user ID with no system prompts or that you can include in a program, use the following command, replacing *userid* with the name of the user ID to remove:

```
vmdirect delentry userid
```

You can also use this command to remove a user ID that is currently on hold by entering the following, replacing *heldusr* with the name of the user ID to remove:

```
vmdirect delentry heldusr (hold
```

Note: For more information about options for this command, see DELENTRY Command in the chapter "Command Reference" in the Reference Guide.

Removing Directory Entries by Filling in the Blanks on a Screen

To remove a user ID with the full-screen method

1. Enter the following command, replacing *userid* with the name of the user ID that you want to remove:

   ```
   vmdirect manage userid
   ```

2. On the Manager Selection Menu, type 5 for Hold or Remove a Userid.
3. On the next screen, indicate that you want to remove the user ID.

Removing Several Directory Entries at the Same Time

You can remove several directory entries at the same time using a control file and the MULTIPLE command. You may even want to use the same control file to remove directory entries that you used to put them on hold previously.

Note: Though the control file is the same for removing directory entries as for putting them on hold, the command is different.

To remove multiple directory entries, enter the MULTIPLE REMOVE command with the CMS file name and file type of the control file. For example, to remove the entries in the control file USERS, filetype FILE, enter the following command:

```
vmdirect multiple remove users file
```

Note: For information about how to create a control file, see Control File Records in the MULTIPLE Command in the chapter "Command Reference" in the *Reference Guide*.

Chapter 9: Directory Profiles

You must be a system administrator (or have ADMIN or ADMIN PROFILE authorization in the AUTHORIZ CONFIG file) to work with directory profiles.

A directory profile must contain a PROFILE statement, and the PROFILE statement must be the first statement. For information about which types of directory statements can be contained in a directory profile, see the *IBM CP Planning and Administration* manual for your release of VM. The IBM manual also describes how to use the INCLUDE statement to make directory profile statements part of a user directory entry.

This section contains the following topics:

Creating a Directory Profile (see page 159)
Editing a Directory Profile (see page 160)
Deleting a Directory Profile (see page 160)
Including a Directory Profile in a Directory Entry (see page 161)
Editing a Directory Entry That Includes a Directory Profile (see page 161)
Replacing a Directory Profile in a Directory Entry (see page 162)
Removing a Directory Profile from a Directory Entry (see page 163)

Creating a Directory Profile

You can create a directory profile outright or copy an existing one to use as the basis for a new one.

You can also create one programmatically from a file on your minidisk by using the ADDENTRY command with the PROFILE option. For more information about the ADDENTRY command, see the chapter "Command Reference" in the *Reference Guide*.

To create a directory profile outright

1. Log on to VMANAGER or a user ID that has ADMIN or ADMIN PROFILE authorization in the AUTHORIZ CONFIG file.
2. Enter the ADMIN PROFILE command with the name of the directory profile that you want to create. For example, to create a directory profile named TCPCMSU2, enter the following command:

   ```
   vmdirect admin profile tcpcmsu2
   ```

   This command inserts the correct PROFILE statement in the directory profile and places you into an XEDIT session.
3. Add the control statements that you want in the directory profile.
4. Save this XEDIT session and exit.

Using an Existing Directory Profile as a Prototype

If an existing directory profile contains many of the control statements that you want for a new directory profile, you can use the existing profile as a prototype for the new one.

To create a directory profile from an existing one

1. Enter the ADMIN PROFILE command, with the directory profile you want to create and the file name of the existing profile you want to copy. For example, to create the profile TCPCMSU1 from the existing profile TCPCMSU, enter the following command:

   ```
   vmdirect admin profile tcpcmsu1 tcpcmsu
   ```

   This command copies the prototype directory profile and places you in XEDIT.
2. Add, remove, or change the control statements that you want in the directory profile.
3. Save this XEDIT session and exit.

Editing a Directory Profile

You can change the contents of a directory profile with the ADMIN PROFILE command, including on the command the name of the directory profile you want to change. This is the same command you use to create a directory profile. The only difference is that you include on the command the name of an existing directory profile rather than a new one.

Changes you make to an existing profile are immediately applied to all directory entries that include that profile.

Deleting a Directory Profile

You can delete obsolete directory profiles if they are not used by any directory entries. This means that they cannot be referenced by an INCLUDE statement in any directory entries.

You can find all user IDs that are currently using a directory profile by entering the following QUERY command, replacing *profname* with the name of that directory profile:

```
vmdirect query prfusers profname
```

When you find which user IDs are using that directory profile, remove the INCLUDE statements that reference that directory profile from their directory entries. For more information about removing INCLUDE statements, see Removing a Directory Profile from a Directory Entry in this chapter.

To delete a directory profile that you no longer need, use the ADMIN command or the DELENTRY command. The ADMIN command lets you delete a directory profile for a specified user ID. The DELENTRY command lets you delete a specific directory profile, and can also be used programmatically.

Examples

- To use the ADMIN command to delete the profile PROFILE1, enter the following:

  `vmdirect admin profile profile1 (erase`

- To use the DELENTRY command to delete the profile PROFILE2, enter the following:

  `vmdirect delentry profile2 (profile`

Including a Directory Profile in a Directory Entry

To have a USER or IDENTITY directory entry reference a directory profile, add an INCLUDE statement to that directory entry. The INCLUDE statement specifies which directory profile is referenced by the directory entry.

To add an INCLUDE statement to a directory entry, enter the following GENINCL command, replacing *userid* with the user ID whose directory entry you want to change and *profname* with the directory profile that you want the user ID's directory entry to include:

`VMDIRECT GENINCL` *`userid profname`*

The GENINCL command adds the proper INCLUDE statement to the user ID's directory entry and removes any control statements that duplicate the statements in the directory profile.

You can create an EXEC to issue the GENINCL command to add directory profiles to multiple user IDs' directory entries.

Editing a Directory Entry That Includes a Directory Profile

To edit a directory entry that references a directory profile, use the EDX command. The EDX command places you in an XEDIT session, and lets you see the *effective* directory entry—the control statements in that directory entry and those from the directory profile. In contrast, the EDIT command lets you see only the control statements that actually appear in the directory entry.

Example:

The directory profile and directory entry for user ID SMITHE are different when you display them with the EDX command or the EDIT command:

If you view SMITHE's directory entry with the EDIT command, you see:

```
USER SMITHE RELIC 12M 32M 64
INCLUDE TESTPRO1
OPTION SETORIG
*FL= N
ACCOUNT 58000099 POLO
*PW=
LINK TESTUSR1 191 19C RR
```

In contrast, if you view SMITHE's directory entry with the EDX command, you see:

```
USER SMITHE RELIC 12M 32M 64
INCLUDE TESTPRO1
OPTION SETORIG
OPTION QUICKDSP
OPTION ACCT MIH
*FL= N
ACCOUNT 58000099 POLO
*PW=
IPL CMS PARM AUTOCR
CONSOLE 009 3215
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK CMS14 190 190 RR
LINK YDISK 19E 19E RR
LINK YDISK 1BF 1BF RR
LINK TESTUSR1 191 19C RR
```

Note: The duplicate LINK statement—the one that defined device 19C for CMS4—in the directory profile is not displayed.

EDX ignores control statements in the directory profile that are overridden by statements in the directory entry. CA VM:Director considers the link statement that the directory profile defines to be a duplicate and bypasses it.

Replacing a Directory Profile in a Directory Entry

Because a directory entry can contain only one INCLUDE statement, changing which directory profile a directory entry uses removes the INCLUDE statement that specified the old directory profile and replaces it with an INCLUDE statement that specifies the new directory profile.

To replace the directory profile specified in a directory entry, use the same GENINCL command you used initially to add the directory profile to the directory entry.

Note: When you added the directory profile to the directory entry, the GENINCL command removed duplicate control statements from the directory entry. These control statements are not returned to the directory entry even if they are no longer duplicates (they do not appear in the new directory profile). Verify that control statements are not *lost* or unintentionally omitted from a directory entry after you change its directory profile.

You can create an EXEC to issue the GENINCL command to replace directory profiles in multiple user IDs' directory entries.

Removing a Directory Profile from a Directory Entry

You can remove a directory profile from a directory entry if you do not want that directory entry to use the control statements in its directory profile. You do so by removing the INCLUDE statement from that directory entry.

To remove an INCLUDE statement from a directory entry, enter the GENINCL command, specifying the user ID whose directory entry you want to change but omitting the directory profile name.

Note: When you added the directory profile to the directory entry, the GENINCL command removed duplicate control statements from the directory entry. These control statements are not returned to the directory entry even though they are no longer duplicates. Verify that control statements are not lost or unintentionally omitted from a directory entry after you remove its directory profile.

You can create an EXEC to issue the GENINCL command to remove directory profiles from multiple user IDs' directory entries.

Chapter 10: Skeleton Files

A skeleton file is a prototype CP directory entry. It includes directory statements and special comments that describe a user ID, such as its links to the MAINT minidisk, its name and password, storage size, priority, and INCLUDE statements that effectively include the definition contained in a different directory profile.

This section contains the following topics:

Working with Skeleton Files (see page 165)
Deleting a Skeleton File (see page 169)
Allowing Directory Managers to Use a Skeleton File (see page 169)
Preventing Directory Managers from Using a Skeleton File (see page 170)

Working with Skeleton Files

The CA VM:Director installation process creates a skeleton file named GENERAL SKELETON. You can customize this file to your system's specifications. You can also create additional skeleton files to address the needs of your various departments.

The supplied GENERAL SKELETON file follows:

```
USER GENERAL GENERAL 16M 32M G      ** CP directory statement
*AC= 99999999                       ** VM:Director special comment
ACCOUNT 99999999 GENERAL            ** CP directory statement
IPL CMS PARM AUTOCR                 ** CP directory statement
CONSOLE 009 3215                    ** CP directory statement
SPOOL 00C 2540 READER *             ** CP directory statement
SPOOL 00D 2540 PUNCH A              ** CP directory statement
SPOOL 00E 1403 A                    ** CP directory statement
LINK MAINT 190 190 RR               ** CP directory statement
LINK MAINT 19E 19E RR               ** CP directory statement
```

To decide whether your system needs more than one skeleton file, consider the following:

- System needs of your various departments
- Virtual machine configurations for various groups of users
- Storage requirements, classes, and priorities required by different groups of users

If you need more skeleton files, assign them descriptive file names, such as SALES or PURCH, to help you use them correctly. They are always of file type SKELETON.

Note: For more information about creating skeleton files, see Creating Skeleton Files (see page 168).

After you create the skeleton files you need, decide which ones, if any, to allow your various directory managers to use when they create user IDs. By giving directory managers access to only certain skeleton files, you control their ability to create user IDs with specific CP privilege classes and other security authorizations.

Note: For information about enabling directory managers to use skeleton files, see Allowing Directory Managers to Use a Skeleton File (see page 169).

Skeleton files can contain:

- CP directory statements
- CA VM:Director special comments
- INCLUDE statements that reference a directory profile

Of the CP directory statements, most skeleton files contain at least three kinds. However, you can include in skeleton files most other CP directory statements that you want in directory entries.

The following CP directory statements are most commonly included in skeleton files:

USER statement

With place holders for the *userid* and logon *password* fields. CA VM:Director updates these fields when you or a directory manager defines a new user ID based on the skeleton file that contains this statement. The *password* is set to match the *userid* field, unless the password is one of the special CP passwords such as NOLOG or AUTOONLY. You can provide additional parameters as necessary, which will be included as specified in any new directory entries.

ACCOUNT statement

Defines the user's account number and distribution code; there can be up to eight account numbers. You can override the account numbers and distribution code specified on an ACCOUNT statement when you create a user ID.

CONSOLE statement

Or at least one other device record. If the USER statement defines a NOLOG user ID, no device record is required.

LINK statements

To link MAINT disks for CMS.

The following CA VM:Director special comments can be included in a skeleton file. They are explained in more detail in Special Comment Reference (see page 233).

ED= special comment

Records the last update to the directory entry that was created using the skeleton file in which this special comment appears. This comment is recommended for audits.

PW= special comment

Record a password history of the user ID that was created using the skeleton file in which this special comment appears. This comment is recommended for audits.

AC= special comment

Lists the account numbers the user ID is allowed to use. This comment is required only if you do not use the ACCOUNT user exit.

ENROLL comment

Enrolls a user ID in SFS, specifying the SFS file pool, user storage group, and amount of space to allocate to the new user ID.

SP= comment

Specifies the PROFILE EXEC file to use for a user ID. (CA VM:Director removes the *SP= statement from the user's directory entry when you create a 191 minidisk for the user, or enroll the user in SFS and update the IPL statement.)

US= comment

(CA VM:Secure only) Specifies a user word for the POSTRULE user exit. For information about user words in the POSTRULE user exit, see the chapter "User Exit Reference" in the *Reference Guide*.

The skeleton file can include MDISK statements as long as the MDISK statement defines a temporary minidisk (T–DISK), a virtual minidisk (V–DISK), or uses the automatic minidisk allocation parameters.

Note: For more information about automatic minidisk allocation, see ADDENTRY Command and ADDMDISK Command in the chapter "Command Reference" in the *Reference Guide*.

Note: Only the ADDENTRY command supports MDISK statements for permanent minidisks in a skeleton file. MANAGE and other commands that create user IDs ignore all permanent MDISK and minidisk–related statements.

Creating Skeleton Files

Skeleton files are created from prototype skeleton files only. When you create a skeleton file, CA VM:Director copies the prototype skeleton file you specify in the skeleton file you create. You can then change any statement, add, or remove statements from your new skeleton file. These changes do not affect the prototype file on which your new one was based.

To create a skeleton file

1. Enter the ADMIN SKELETON command with the name of the skeleton file to create and the name of the prototype skeleton file on which to base it. Do not include the file type for either the new file or the prototype file. See Example below.

   This command opens the new skeleton file in XEDIT.

2. Change whatever statements need to be modified, or add or remove statements in the new skeleton file.

3. Save your changes and exit from XEDIT.

You can now authorize yourself or any directory manager to use this skeleton file to create user IDs. To do so, see Allowing Directory Managers to Use a Skeleton File (see page 169).

Example:

You want to create a skeleton file for the directory manager of user IDs in the Sales Department to use when he creates Sales user IDs.

To create SALES SKELETON, based on the prototype GENERAL SKELETON, enter the following:

```
vmdirect admin skeleton sales general
```

Changing a Skeleton File

To change an existing skeleton file:

1. Enter the ADMIN SKELETON command with the name of the skeleton file to change. For example, to change the file SALES SKELETON, enter the following command:

   ```
   vmdirect admin skeleton sales
   ```

   Do not include the file type.

This command opens the skeleton file in XEDIT.

2. Change whatever statements need to be modified, or add or remove statements in the new skeleton file.
3. Save your changes and exit from XEDIT.

Deleting a Skeleton File

You can delete a skeleton file even if user IDs were created based on it, because these user IDs do not reference the skeleton file after they are created. However, you cannot delete a skeleton file if directory managers are authorized to use it, so you must first remove a skeleton file from each directory manager before you can remove it from the system.

To delete an existing skeleton file, enter the ADMIN SKELETON command with the name of the skeleton file to delete and the ERASE option. For example, to erase the file SKELFILE SKELETON, enter the following command:

```
vmdirect admin skeleton skelfile (erase
```

There is no need to confirm the deletion.

Allowing Directory Managers to Use a Skeleton File

To allow a directory manager to use a skeleton file when creating user IDs, you must include a SKELETON record in the VMDIRECT MANAGERS file for each directory manager who creates user IDs. On the SKELETON record, specify the names of the skeleton files the directory manager can use. The first skeleton file listed is the directory manager's default.

Note: For more information about the VMDIRECT MANAGERS file and the format of the SKELETON record, see VMDIRECT MANAGERS File Reference (see page 225).

To find out which skeleton files a directory manager is allowed to use, use the QUERY command.

Example:

To find out which skeleton files directory manager WOODYB is allowed to use, enter the following:

```
vmdirect query skeleton (manager woodyb
```

Preventing Directory Managers from Using a Skeleton File

You can remove a skeleton file from the list of skeleton files that a directory manager is authorized to use. Doing so prevents that directory manager from using that skeleton file when creating a user ID.

To prevent a directory manager from using a skeleton file, edit the VMDIRECT MANAGERS file and locate the SKELETON record for the directory manager. Then delete the name of the skeleton file from the skeleton files listed that you want to prevent the directory manager from using.

Note: For information about the VMDIRECT MANAGERS file and the format of the SKELETON record, see VMDIRECT MANAGERS File Reference (see page 225).

Chapter 11: Managing Space

This chapter explains how to create, analyze, move, and transfer minidisks.

This section contains the following topics:

Creating Minidisks (see page 171)
Duplicating a Minidisk (see page 173)
Analyzing Minidisks (see page 174)
Moving Minidisks or Changing Minidisk Information (see page 176)
Transferring a Minidisk from One User ID to Another (see page 178)
Protecting Minidisks from Being Scratched (see page 179)
Allocating DASD (see page 179)
Adding DASD to the CA VM:Director Configuration (see page 180)
Mapping a DASD Volume (see page 181)
Defragmenting a DASD Volume (see page 181)
Deleting Minidisks (see page 181)

Creating Minidisks

CA VM:Director provides the following commands to create minidisks. Each has advantages in different situations.

Method	Advantages
ADDMDISK command (line-mode)	No prompts. You enter all information in one command, making ADDMDISK useful for creating minidisks through programming applications.
DUPMDISK command (line-mode)	The easiest method for producing exact duplicates of minidisks. This is a good method to use when you are cloning guest images such as a Linux server.
MAINT command (line-mode)	Usually faster than full-screen methods. You enter minidisk attributes one at a time in response to screen prompts but without waiting for full screens to change.
MANAGE command (full-screen)	Usually the easiest method for creating minidisks. You follow the instructions on the screens and fill in the blanks to define the attributes of the minidisk you are creating.

Virtual Address Restrictions

Virtual addresses in the range 0600 to 06FF are used for CA VM:Director functions. You must not assign minidisks to CA VM:Director in this range because CA VM:Director will attempt to reclaim them.

Note: For more information about reclaiming DASD space from minidisks, see RECLAIM Command in the chapter "Command Reference" in the *Reference Guide*.

Virtual addresses in the range 0010 to 00FF are reserved by CA VM:Director for use as virtual printers in support of the CA VM:Director TRACE command. Virtual addresses 0FFD, 0FFE, and 0FFF are also reserved for the CA VM:Director service virtual machine. Do not assign minidisks or any other devices to addresses in this range because CA VM:Director will detach them during initialization.

Note: For more information about tracing printer spool files, see TRACE Command in the chapter "Command Reference" in the *Reference Guide*.

Creating Minidisks Using Programming Applications

Enter the following command:

`VMDIRECT ADDMDISK entry vaddr subpool size devtype`

entry

The directory entry, which will contain the new minidisk.

vaddr

The virtual address of the new minidisk.

subpool

The subpool from which to allocate space for the new minidisk.

size

The size of the new minidisk.

devtype

The device type of the new minidisk.

Note: For more information, see ADDMDISK Command in the chapter "Command Reference" in the *Reference Guide*.

You can also use the ADDENTRY command API, which calls ADDMDISK, to create a minidisk.

Note: For more information about the ADDENTRY command, see the chapter "Command Reference" in the *Reference Guide*.

Creating Minidisks by Answering Screen Prompts

Enter the following command:

`VMDIRECT MAINT MANAGE entry NEWDISK vaddr`

entry

The directory entry which will contain the new minidisk.

vaddr

The virtual address of the new minidisk.

CA VM:Director prompts you for the minidisk size, passwords, and format features of the new minidisk when it needs them. Enter the information one item at a time, when it asks you for it.

Creating Minidisks by Filling in the Blanks on a Screen

To create minidisks by filling in the blanks

1. Enter the MANAGE command with the directory entry that will contain the new minidisk. For example, to create a new minidisk for user ID ZLINUX01, enter the following command:

   `vmdirect manage zlinux01`

2. On the Manager Selection Menu, enter the virtual address of the new minidisk in the For Device field. The default is 191.
3. Choose selection 2, Define and Format a New Minidisk. Supply the minidisk size, passwords, and format parameters in response to prompts on the screens that appear.

Duplicating a Minidisk

The DUPMDISK command lets you create an exact duplicate of a specified existing minidisk. The resulting minidisk will be an exact duplicate of the input minidisk, having the same size, device type, label, and other attributes. If the target minidisk already exists, and has the same device type and size as the source disk, then an exact copy of the data on the source disk will be made on the target disk.

While the command can be used to duplicate a CMS minidisk, it also duplicates Linux formatted minidisks to assist in cloning Linux images on VM. By first making a master image and then using the DUPMDISK command to duplicate the directory entry and the minidisks, you can quickly set up multiple identical Linux images.

Note: For more information, see the DUPMDISK command, in the chapter "Command Reference" in the *Reference Guide*.

Analyzing Minidisks

Analyze your system's minidisks on a regular basis to determine which minidisks to enlarge so that users do not run out of space and which to reduce so that you do not waste DASD. Your system can often use DASD more effectively if you reblock the minidisk to accommodate the size of the files on the minidisk.

The MDSKSCAN command provides the following information about minidisks in your system:

- User ID that owns the minidisk
- Virtual address
- Minidisk format (CMS or non-CMS)
- Block size
- Number of files
- Total size
- Percent used
- Date and time of last update
- Manager of the user ID that owns the minidisk

The MDSKSCAN command calls the VMXAQD MODULE to access and query each minidisk in a read-only mode. If your site does not use the Servant Facility, make the VMXAQD MODULE accessible on a public minidisk.

If you want to enter the MDSKSCAN command from an EXEC, give the user ID that issues the command NOPASS authorization in the AUTHORIZ CONFIG file. For example:

```
GRANT NOPASS MDSKSCAN TO userid
```

Example:1

To get information about the minidisks for all user IDs that you, WOODYB, manage, enter the following:

```
vmdirect mdskscan *
```

CA VM:Director produces a report similar to the one below:

```
USERID  VADR TYPE BLKSZ FILES  SIZE   % DATE     TIME     MANAGER
------- ---- ---- ----- ------ -------- -------- -------- -------
SALE01  191  CMS  4096      1  300K   2 yy/mm/dd 13:07:22 WOODYB
SALE02  191  CMS  4096     25  300K  80 yy/mm/dd 13:07:26 WOODYB
SALESP  191  CMS  4096     15  300K  52 yy/mm/dd 13:07:33 WOODYB
TEST_1  191  CMS  4096      1  300K   2 yy/mm/dd 13:07:22 WOODYB
TEST_2  192  OS
TEST_4  191  CMS  4096     15  300K  52 yy/mm/dd 13:07:33 WOODYB
```

Example:2

To get information about the minidisks for all user IDs that begin with SALE, enter the following:

```
vmdirect mdskscan sale*
```

CA VM:Director produces a report similar to the following:

```
USERID  VADR TYPE BLKSZ FILES  SIZE   % DATE     TIME     MANAGER
------- ---- ---- ----- ------ ------ --- ------ -------- -------
SALES01 191  CMS  4096      1  300K   2 yy/mm/dd 13:07:22 WOODYB
SALES02 191  CMS  4096     25  300K  80 yy/mm/dd 09:25:11 WOODYB
SALESSP 191  CMS  4096     15  300K  52 yy/mm/dd 23:59:59 WOODYB
```

Example:3

To get a report on the minidisks of all user IDs that begin with TEST_ for all directory managers, enter the following:

```
vmdirect mdskscan test_* (manager *
```

The **manager *** option includes all user IDs for all managers, not only those that you manage.

CA VM:Director produces a report like the following. In this example, TEST_2 has a non-CMS 192 minidisk. It is in OS or DOS format.

```
USERID  VADR TYPE BLKSZ FILES  SIZE   % DATE     TIME     MANAGER
------- ---- ---- ----- ------ -------- -------- -------- -------
TEST_1  191  CMS  4096      1  300K   2 yy/mm/dd 13:07:22 WOODYB
TEST_2  191  CMS  4096     25  300K  80 yy/mm/dd 13:07:26 CARLAT
TEST_2  192  OS
TEST_3  191  CMS  4096      8  768K   4 yy/mm/dd 13:07:30 CARLAT
TEST_4  191  CMS  4096     15  300K  52 yy/mm/dd 13:07:33 WOODYB
```

Moving Minidisks or Changing Minidisk Information

CA VM:Director gives you three commands to move minidisks or change minidisk information. Each has advantages in different situations.

Method	Advantages
CHGMDISK command (line-mode)	No prompts. You enter all information in one command, making CHGMDISK useful for moving minidisks and changing minidisk information through programming applications.
MAINT command (line-mode)	Usually faster than full-screen methods. You enter new minidisk attributes, such as their locations, one at a time in response to screen prompts but without waiting for full screens to change.
MANAGE command (full-screen)	Usually the easiest method for moving minidisks or changing information. You follow the instructions on the screens to specify the appropriate information.

You can manage the CA VM:Director minidisks the same way you manage other user's minidisks; all these methods are valid on them.

Moving Minidisks or Changing Minidisk Information Using Programming Applications

Enter the following command:

```
VMDIRECT CHGMDISK entry vaddr subpool size devtype
```

entry

The directory entry containing the minidisk that you want to move.

vaddr

The virtual address of the minidisk that you want to move.

subpool

The subpool from which to allocate space for the minidisk.

size

The new size of the minidisk.

devtype

The device type of the minidisk.

Note: For more information about additional parameters and options for the CHGMDISK command, see the chapter "Command Reference" in the *Reference Guide*.

You can also use the REPENTRY command API, which calls CHGMDISK, to change minidisk information.

Note: For more information about the REPENTRY command, see the chapter "Command Reference" in the *Reference Guide*.

Moving Minidisks or Changing Minidisk Information by Answering Screen Prompts

Enter the following command:

```
VMDIRECT MAINT MANAGE entry DISKMOVE vaddr
```

entry

The directory entry containing the minidisk that you want to move.

vaddr

The current virtual address of the minidisk that you want to move.

Specify where to move the minidisk when prompted for this information.

Moving Minidisks or Changing Minidisk Information by Filling in Blanks on a Screen

To move minidisks or change minidisk information

1. Enter the MANAGE command with the name of the directory entry containing the minidisk that you want to move. For example, if you want to move a minidisk that belongs to user ID BEAGA01, enter the following command:

   `vmdirect manage beaga01`

2. On the Manager Selection Menu, enter the virtual address of the minidisk to move.
3. Choose selection 6, Move a Minidisk. Specify where to move the minidisk in response to prompts on the screens that appear.

Transferring a Minidisk from One User ID to Another

The TRANSFER command lets you transfer ownership of a minidisk from one user ID to another and change the virtual device address of a minidisk.

To transfer a minidisk, enter the following command:

`VMDIRECT TRANSFER` *`entry1 vaddr1 entry2 vaddr2`*

entry1

The directory entry containing the minidisk that you want to transfer.

vaddr1

The current virtual address of the minidisk that you want to transfer.

entry2

The new directory entry, which will contain the minidisk.

vaddr2

The new virtual address of the minidisk.

Example:

To move CLIFFC's 191 minidisk to REBECCAH as her 192 minidisk, enter the following:

`vmdirect transfer cliffc 191 rebeccah 192`

Protecting Minidisks from Being Scratched

Use the SCRATCH user exit to prevent accidental scratching or moving of vital minidisks or virtual machines. Code the exit to your system–specific standards, and it conditionally accepts or rejects a scratch or move request.

Note: For information about coding the SCRATCH user exit, see the chapter "User Exit Reference" in the *Reference Guide*.

Allocating DASD

When you allocate DASD for new minidisks, CA VM:Director searches only the subpool that you request by name for available space. However, you can have CA VM:Director search several subpools at the same time by requesting that it search categories of subpools. The categories can vary over time because they are based on the subpool definitions you include in the VMDIRECT MANAGERS file; if you change the subpool definitions in that file (for instance, if you change the subpools that a directory manager can allocate from), these keywords specify different subpools.

CA VM:Director searches a subpool list according to the order in which the MANAGERS file lists the subpools. Because you can change a manager's subpool list dynamically and the order of subpools (SUBPOOL records) in the DASD CONFIG file, you can change the search order dynamically.

Subpool Categories

Subpool categories are predefined variables that represent certain groups of subpools from which to allocate space. To use subpool categories, you must be a manager and have a subpool list defined in the MANAGERS file. You can specify subpool categories with the following commands:

- ADDENTRY (on an MDOPT record in the skeleton file or directory entry provided)
- ADDMDISK
- COMPRESS
- MAINT MANAGE
- MANAGE
- MULTIPLE (the second subpool parameter only)

- REPENTRY (on an MDOPT record in the entry that you provide)

The subpool categories are listed in the following table.

Variable Subpool Category	Subpools Specified†
*MINE	Your list of subpools
*MNGR	Your owning manager's list of subpools
*BOTH	Your owning manager's list of subpools and your list of subpools, in that order
*ANY‡	Your owning manager's list of subpools and your list of subpools in that order, then the remaining subpools in the reverse order of their appearance in the DASD CONFIG file

† Subpools specified by a category include all the subpools listed in the DASD CONFIG file that fall in that category except those defined with the EXPLICIT attribute on a SUBPOOL record.

‡ You must have ANYPOOL authorization to use the *ANY subpool category.

Note: For more information, see Authorizations for Command Processing (see page 129).

Adding DASD to the CA VM:Director Configuration

If you acquire more DASD or you want CA VM:Director to manage other volumes, you must identify these DASD volumes to CA VM:Director. To identify a DASD volume, add one VOLUME record and EXTENT records to the DASD CONFIG file.

To add DASD volumes to the DASD CONFIG file

1. Edit the DASD CONFIG file by entering:

   ```
   vmdirect config dasd
   ```

2. Find the VOLUME and EXTENT records in the file and add the records for your new DASD.

   Note: For more information about the format of the VOLUME Record and EXTENT Record, see the chapter "Configuration File Reference" in the *Reference Guide*.

Mapping a DASD Volume

The MAP command lets you map the disk space utilization for one or more DASD volumes. For a volume, CA VM:Director can map all extents that you identify to CA VM:Director and the minidisks you define within those extents. It does not map minidisks that you explicitly ignored.

With the MAP command, you can map:

- All minidisks, extents, and free disk space
- Minidisks only
- Extents only
- Free disk space only

Note: For more information about extents, the EXTENT record, and the MAP command, see the *Reference Guide*.

Defragmenting a DASD Volume

Use the COMPRESS command to control or eliminate excessive fragmentation of allocable disk space within extents. CA VM:Director moves minidisks in the extent you specify to fill up free areas from the beginning of the extent toward the end of the extent.

Note: For information about how to use the COMPRESS command, see the chapter "Command Reference" in the *Reference Guide*.

Deleting Minidisks

When you use one of the following commands to delete a minidisk that is on a volume under CA VM:Director control, CA VM:Director also deletes links to that minidisk from the directory:

- DELMDISK
- MAINT MANAGE *userid*—subfunction SCRATCH *vaddr*
- MAINT MANAGE *userid*—subfunction REMOVE
- MOVE2SFS with the DELETE option
- MANAGE—selection 5 ‘Hold or Remove a Userid’ (Remove function only)

- MANAGE—selection 4 ‘Scratch a Minidisk’
- REPENTRY - in cases where the new directory entry deletes a minidisk that existed in the old directory entry
- MULTIPLE REMOVE
- DELENTRY

Chapter 12: SFS Overview

You can have CA VM:Director manage the enrollment of users and allocation of space in SFS by implementing the various SFS configuration and management functions in CA VM:Director. These functions enable you to describe your SFS configuration to CA VM:Director, including which file pools and storage groups you want to manage through CA VM:Director and how much space can be allocated in each storage group. They further let you specify who can enroll users under CA VM:Director management, in which file pools they can enroll users, and how much space they can allocate to users in each storage group.

The user IDs that describe your SFS configuration to CA VM:Director are SFS administrators. The user IDs that perform user administration tasks in SFS through CA VM:Director management are SFS managers.

All but one of the configuration functions for CA VM:Director SFS are performed by SFS administrators. The only function that must be performed by the CA VM:Director system administrator rather than the SFS administrator is identifying the SFS administrators to CA VM:Director. Most of the daily management functions for SFS using CA VM:Director are performed by SFS managers.

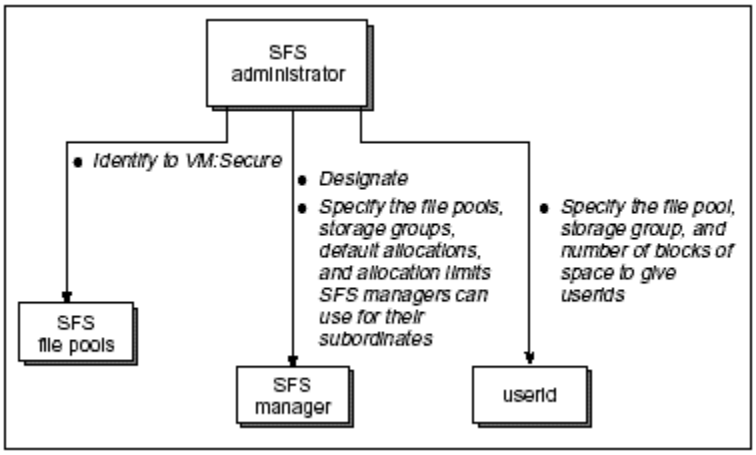
This section contains the following topics:

SFS Administrators (see page 184)
SFS Managers (see page 184)
CA VM:Director Configuration for SFS User Administration (see page 188)
Allocation of Space in User Storage Groups (see page 195)

SFS Administrators

SFS administrators in CA VM:Director perform three primary functions:

- Identify to CA VM:Director the SFS file pools and user storage groups on your system that it will manage
- Designate existing directory managers as SFS managers, and specify which file pools, storage groups, and default allocations those managers can use when they allocate SFS space under CA VM:Director
- Enroll any user ID in any SFS file pools under CA VM:Director control, giving that user ID any amount of file space in any storage group



Administrators can also remove SFS manager authority from user IDs and change or remove the file space allocated to individual user IDs.

SFS Managers

SFS managers in CA VM:Director are any directory managers that an SFS administrator designates to be SFS managers. Managers can add, enroll, or delete any user ID that they can manage in SFS.

The file pools, user storage groups, file space size, and maximum file space size that an SFS manager can assign to its subordinate user IDs are determined by an SFS administrator.

SFS managers can do the following:

- Enroll in SFS the user IDs they manage as directory managers
- Change SFS allocations for the user IDs they manage by giving those user IDs space in additional file pools or in file pools those user IDs already use
- Remove user IDs' allocations in individual file pools or remove those user IDs from SFS entirely

If the SFS administrator wants other people to oversee the daily administration of user IDs and SFS space, the administrator must specify SFS managers to perform this user administration.

Enrollment Limits

An SFS administrator sets up an SFS manager with limits. These limits include the following:

- From which file pools this manager is authorized to allocate space
- From which user storage groups in these file pools this manager is authorized to allocate space
- How much space this manager is authorized to allocate from each user storage group to all managed user IDs.

Example:

The SFS administrator authorized SFS manager BGEDDES to allocate space in the following file pools. The amount of space BGEDDES is authorized to allocate for each user storage group in each of these file pools appears in parentheses following the file pools:

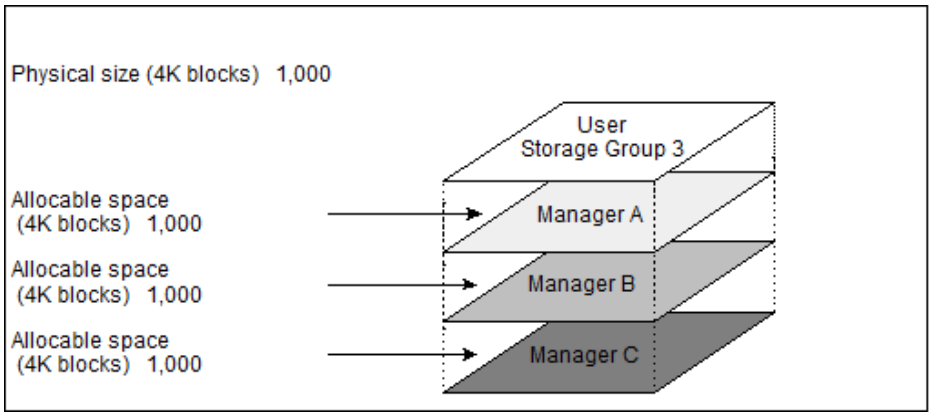
SFS Manager:	**BGEDDES**			
File pools:	VMXA	VMXB	VMXC	VMXD
User storage groups (number of blocks):	4 (5,000)	2 (10,000)	2 (8,000)	3 (5,000)
	5 (10,000)	3 (9,000)	4 (15,000)	4 (10,000)
	7 (50,000)	4 (1,000)	5 (15,000)	5 (1,500)

These file pools, the user storage groups, and the maximum allocable space BGEDDES has in each user storage group are his enrollment limits.

Summed Enrollment Limits for All SFS Managers

The space that an SFS administrator authorizes each SFS manager to allocate in a user storage group is literally that amount of space in the user storage group, up to his limit, and not a percentage of the total space available in the user storage group. (There is another factor determining how much space SFS managers can allocate, but more on this later.)

For example, the SFS administrator can specify that each of three managers can allocate 1,000 blocks of space in User Storage Group 3, which has 1,000 blocks of space:



Notice that the three managers are allowed to allocate a total of 3,000 blocks (their allocable space is 1,000 blocks + 1,000 blocks + 1,000 blocks) even though the available space is 1,000. This is because the managers' allocable space is not cumulative.

What their limits on allocation really mean is:

- Manager A is allowed to allocate up to 1,000 blocks of space in User Storage Group 3, for use by Manager A's user IDs only if the space is available when they want it
- Manager B is allowed to allocate up to 1,000 blocks of space in User Storage Group 3, for use by Manager B's user IDs only if the space is available when they want it
- Manager C is allowed to allocate up to 1,000 blocks of space in User Storage Group 3, for use by Manager C's user IDs only if the space is available when they want it.

Default Enrollment Values

An SFS administrator can make an SFS manager's enrollment functions easier by setting up enrollment defaults for that manager. Several defaults are available for an SFS administrator to define for an SFS manager, such as a default enrollment file pool, a default user storage group in each file pool, and a default allocation size in each user storage group.

The administrator sets up enrollment defaults when defining SFS managers, and can change the defaults at any time. The managers must have allocation authority to the file pool assigned to them as the default file pools, and to each user storage group defined as their default user storage group for each file pool. Additionally, they must be able to allocate at least as much space in each user storage group as the default amount of space set up in their defaults.

If an SFS manager accepts the enrollment defaults as the enrollment values for a user ID, they are actually passed on to SFS.

For example, these are the default enrollment values for SFS manager BGEDDES:

SFS manager:	**BGEDDES**
Default file pool:	VMXA
Default user storage group:	4
Default allocation size:	100 blocks

Her authority includes allocable space of 5,000 blocks in user storage group 4 in file pool VMXA, so these default enrollment values will help her enroll several user IDs in SFS easily.

When BGEDDES enrolls a user ID in SFS, these enrollment defaults are the ones that CA VM:Director displays for that user ID. She can change these default values for this user ID at the time she enrolls this user ID simply by typing over the default values she wants to change. For example:

SFS manager:	**BGEDDES**		
Enroll user ID:	SREED		
Initial enrollment values for user ID SREED (BGEDDES's default enrollment values):		Final enrollment values for user ID SREED (BGEDDES's changes at enrollment time):	
File pool:	VMXA	File pool:	VMXB
User storage group:	4	User storage group:	2
Allocation size:	100 blocks	Allocation size:	500 blocks

SFS itself now indicates that user ID SREED has 500 blocks of space in user storage group 2 of file pool VMXB.

Any subsequent changes an SFS administrator makes to BGEDDES's default enrollment values do not affect any user IDs that BGEDDES already enrolled in SFS with her previous default enrollment values.

If an administrator sets up a manager's enrollment defaults and then removes the default file pool from that manager's authority, the administrator will erase that manager's enrollment defaults. This does not affect any user IDs BGEDDES already enrolled in SFS with her previous default enrollment values, but does prevent her from enrolling user IDs in SFS. The SFS administrator must assign new enrollment defaults to this manager before she can enroll user IDs again.

CA VM:Director Configuration for SFS User Administration

The CA VM:Director configuration for SFS consists of two parts:

- Description of the SFS file pools and the user storage groups in those file pools
- Definition of the VM user IDs that will be SFS administrators and SFS managers, and the scopes of the managers' authorities

The description of a file pool is simply the file pool's name. The description of a user storage group is the user storage group's number under its file pool and its allocation limit.

The description of SFS to CA VM:Director includes only the file pools you defined for SFS that you want CA VM:Director to manage. For each file pool described, CA VM:Director needs only the user storage groups you want it to manage.

The definition of SFS administrators and SFS managers in CA VM:Director identifies which user IDs can change the CA VM:Director configuration for SFS and which user IDs can manage user IDs under SFS. SFS administrators' requirements and responsibilities are explained in the section, SFS Administrators, earlier in this chapter; SFS managers' requirements and responsibilities are explained in the SFS Managers section.

Automatic Configuration

If SFS is already in use at your site when you install CA VM:Director, you can identify each file pool that you want CA VM:Director to manage, then have CA VM:Director determine the configuration of this file pool and the user IDs who use and manage it.

This function is available on the same screen at which you identify file pools to CA VM:Director, through the PF2(Auto–Config) key.

Note: For more information about the actual procedure, see Adding a File Pool and User Storage Groups to the CA VM:Director Configuration (see page 200).

As part of this determination of the SFS configuration for each specified file pool, CA VM:Director checks the following:

All user storage groups

All user storage groups that CA VM:Director finds in the specified file pool that are in use by valid user IDs are identified to CA VM:Director so that user IDs can be assigned space in them through CA VM:Director. If there are user storage groups that you do not want managed through CA VM:Director, you can remove them from the CA VM:Director information after it completes its automatic configuration.

The allocation limit CA VM:Director sets for each user storage group identified to CA VM:Director is 300 percent. This means that each user storage group under the control of CA VM:Director, can be allocated to 300 percent of its physical space. (The section, Allocation of Space in User Storage Groups, explains how user storage groups can be allocated beyond their physical sizes.)

User IDs currently enrolled in SFS

The names of all user IDs currently using that file pool are added to the CA VM:Director list of enrolled user IDs. The user storage groups in which these user IDs are currently using space are listed in CA VM:Director as the ones they are authorized to use, and the amount of space they are currently using is their limit.

For instance, if user ID FSTONE is using 15 K blocks of space in user storage group 3 in file pool VMX004, FSTONE's enrollment information in CA VM:Director is for file pool VMX004, user storage group 3, with a space limit of 15 K blocks.

Directory managers for those user IDs

For each file space that is owned by a VM user ID, CA VM:Director finds the name of that user ID's directory manager and adds that directory manager to CA VM:Director as an SFS manager. Further, those VM user IDs are assigned their directory managers as their SFS managers.

Authority of SFS Managers Defined Through Automatic Configuration

The authority an SFS manager has, as defined through the CA VM:Director automatic configuration, is restricted to the file space used by the user IDs that SFS manager manages.

The first user ID identified as being managed by a particular directory manager causes that directory manager to also be made an SFS manager. The limits of that SFS manager's authority are the file pool and the user storage group in which that user ID's file space is located.

Example:

The automatic configuration of the file pool VMLGL finds the following for user ID PMASON, who is managed by DSTREET:

User ID:	**PMASON**
Directory manager:	DSTREET
File pool:	VMLGL
User storage group:	4
Space allocated:	80 blocks

Therefore, automatic configuration makes DSTREET an SFS manager, and authorizes DSTREET to enroll user IDs in user storage group 4 of file pool VMLGL:

User ID:	**PMASON**		
Directory manager:	DSTREET	SFS manager:	DSTREET
File pool:	VMLGL	File pool:	VMLGL
User storage group:	4	User storage group:	4
Space allocated:	80 blocks		

Further, now that DSTREET can allocate space in file pool VMLGL, user storage group 4, CA VM:Director assigns DSTREET a default allocation size (the amount of space DSTREET allocates by default to users she enrolls in VMLGL user storage group 4) and an allocation limit for that user storage group.

The default allocation size is 150 blocks, and the allocation limit is equal to the amount of space already allocated to PMASON:

User ID:	**PMASON**		
Directory manager:	DSTREET	SFS manager:	DSTREET
File pool:	VMLGL	File pool:	VMLGL
User storage group:	4	User storage group:	4
Space allocated:	80 blocks	Default allocation:	150 blocks
		Allocation limit:	80 blocks

All subsequent user IDs identified as being managed by that same directory manager, who is now an SFS manager, cause that SFS manager's limits of SFS authority to be expanded. The limits are expanded by the amount of space allocated to those user IDs in the current file pool, which is being automatically configured. For example, if automatic configuration next finds that the directory manager DSTREET also manages user ID PDRAKE, and PDRAKE is using 250 blocks of space in file pool VMLGL, user storage group 4, DSTREET's allocation authority is now expanded to this:

SFS manager:	**DSTREET**
File pool:	VMLGL
User storage group:	4
Default allocation:	150 blocks
Allocation limit:	330 blocks

If the SFS administrator runs automatic configuration on several file pools, the limits of an SFS manager are expanded also in the number of file pools if other file pools include space owned by user IDs that are managed by this SFS manager. For example, automatic configuration of the file pool VMPROD also finds the following:

User ID:	**MARYR**
Directory manager:	DSTREET
File pool:	VMPROD
User storage group:	2
Space allocated:	500 blocks

Because MARYR is also managed by DSTREET, DSTREET now becomes authorized to enroll user IDs in user storage group 2 of file pool VMPROD. DSTREET is assigned a default allocation size of 150 blocks in this user storage group (this is always the default allocation size for automatically configured SFS managers), and an allocation limit of 500 blocks (this is the current amount of space used by user ID MARYR):

SFS manager:	**DSTREET**
File pool:	VMLGL
User storage group:	4
Default allocation:	150 blocks
Allocation limit:	330 blocks
File pool:	VMPROD

SFS manager:	DSTREET
User storage group:	2
Default allocation:	150 blocks
Allocation limit:	500 blocks

Subsequent Configuration Changes

You can change the CA VM:Director configuration for SFS at any time, for instance, when you add file pools or need to remove an SFS manager. Remember, these changes may affect what your SFS managers, and sometimes the user IDs they manage, can do. For example, if you add file pools, you need to allow SFS managers to assign that space to have that space managed through CA VM:Director. If you need to remove an SFS manager, you must move all of that manager's reporting user IDs to another manager first.

Information that you change about SFS in the CA VM:Director configuration (after the automatic configuration) include any of the following:

- Add or remove user storage groups within known file pools
- Change the allocation limits on user storage groups
- Add or remove SFS managers
- Change the default file pool or allocation size for an SFS manager

Resynchronizing the CA VM:Director Configuration of SFS

SFS allocations that anyone makes outside CA VM:Director are unknown to CA VM:Director, making its SFS administration information unreliable. Therefore, you must verify that no allocations are made outside CA VM:Director or resynchronize the CA VM:Director configuration of each SFS file pool so that it reflects the actual current state of that file pool. To prevent outside allocations, make CA VM:Director the only SFS administrator defined for each file pool.

Resynchronizing the CA VM:Director configuration of SFS works much like the automatic configuration of CA VM:Director for SFS, with the following differences:

SFS manager authority

If an SFS manager no longer has any space allocated in the specified file pool or in a particular user storage group in the specified file pool, CA VM:Director does not remove that manager's authority to that file pool or that user storage group.

For example, SFS manager DSTREET is authorized to allocate space from user storage group 2 of file pool VMPROD. The only user ID she manages that used this user storage group is MARYR, who owned 330 blocks of file space in this user storage group. However, MARYR no longer owns space in this storage group, but is instead using 250 blocks of file space in user storage group 4:

User ID:	**MARYR**		
Directory manager: **DSTREET**			
Old Allocation:		New Allocation:	
File pool:	VMPROD	File pool:	VMPROD
User storage group:	2	User storage group:	4
Space allocated:	500 blocks	Space allocated:	250 blocks

You resynchronize the CA VM:Director configuration for file pool VMPROD, and DSTREET comes out of the resynchronization with allocation authority for both user storage group 2 and user storage group 4, even though none of the user IDs she manages now uses space in user storage group 2. She can allocate no space from user storage group 4, which the resynchronization added to her list of user storage groups, because her allocation limit in this newly added user storage group is 0 (this is always the case for user storage groups added to an SFS manager during resynchronization):

SFS manager:	**DSTREET**
File pool:	VMPROD
User storage group:	2
Default allocation:	150 blocks
Allocation limit:	330 blocks
File pool:	VMPROD
User storage group:	4
Default allocation:	150 blocks
Allocation limit:	0 blocks

Overallocation of space in user storage groups

If user IDs using the file pool have more space allocated to them in a given user storage group than their SFS managers are allowed to allocate through CA VM:Director, their allocations are summed and listed in their managers' current allocations, but their managers' allocation limits are not raised.

These managers cannot allocate more space in these overallocated file pools. They must ask their SFS administrators to raise their allocation limits beyond their current actual allocations if they need to allocate more space.

For example, manager DSTREET is authorized to allocate 500 blocks of space in user storage group 5 of file pool VMPROD:

SFS manager:	**DSTREET**
Allocable file pool:	VMPROD
User storage group:	5
Allocation limit:	500 blocks
Current allocation:	300 blocks

Resynchronization of this file pool reveals that half a dozen of DSTREET's user IDs were allocated 800 blocks of space in this user storage group. Her new information in CA VM:Director is as follows:

SFS manager:	**DSTREET**
Allocable file pool:	VMPROD
User storage group:	5
Allocation limit:	500 blocks
Current allocation:	800 blocks

She is no longer able to allocate space in this user storage group unless the SFS administrator expands her authority.

User ID directory entries

The directory entries of user IDs that are enrolled in SFS and that are managed in SFS through CA VM:Director include *FP= special comments. One *FP= comment lists several file pools, depending on the length of the file pool names. (There is no correspondence between the number of *FP= comments in a directory entry and the number of file pools a user ID is authorized to use.) All user IDs' directory entries are updated so that they have the correct *FP= special comments; file pools in which a user ID was enrolled outside CA VM:Director are added to that user ID's *FP= special comment, and file pools in which a user ID no longer owns space are removed.

Important! Do not change these *FP= comments or remove them from directory entries. Doing so invalidates the information CA VM:Director has about SFS and so makes SFS administration through CA VM:Director incomplete. For more information about this special comment, see *FP= (File Pool) in Special Comment Reference (see page 233).

Allocation of Space in User Storage Groups

An SFS administrator can specify an allocation limit for each user storage group in any file pool known to CA VM:Director. The allocation limit is the amount of storage that *all* SFS managers can allocate from a particular user storage group, and is normally greater than the physical amount of storage.

For instance, you can set the allocation limit of a user storage group to 100 percent. This means that SFS managers can allocate the space in this user storage group up to its physical size. If the size is 10,000 blocks, then SFS managers can allocate 10,000 blocks of its space. However, you can also set the allocation limit of a user storage group to 500 percent. In this case, SFS managers can allocate the space in this user storage group up to five times its physical size; if the size is 10,000 blocks, then SFS managers can allocate 50,000 blocks of space.

Allocation of a user storage group beyond its physical size allows for the most efficient use of space on your system. It is similar to the practice of overbooking airline flights: not everyone with an airline reservation shows up for the flight, so that reserving several more seats than actually exist on a plane increases the chances that the flight will be as full as possible. Similarly, not every user ID with an allocation in a user storage group uses its full amount of space, so that allocating more space than actually exists in a user storage group increases the chances that the space will be used as fully as possible.

You can allow a user storage group to be allocated by any percentage of the physical amount of space, from 100 percent to 9999 percent, where 100 percent means that the space can be allocated only to its physical size.

Because this allocation limit is a percentage of the user storage group's physical size, you can increase the size of a user storage group without having to adjust this allocation limit. For instance, if you allow 200 percent allocation for a user storage group that has 10,000 blocks, that storage group has 20,000 blocks of effectively allocable space. If you later increase the physical size of this storage group to 25,000 blocks, its effectively allocable space becomes 50,000 blocks.

Interaction Between an SFS Manager's Allocation Limit and a User Storage Group's Allocation Limit

The combination of allocation limits on user storage groups and allocation limits on managers allows for the most efficient distribution and use of space on your system. These limits allow for the incomplete use that user IDs make of the space they can use in a user storage group.

CA VM:Director does not know how many blocks of space exist in a user storage group, or how many are available at any time, until it tries to process an SFS manager's request to allocate space to a user ID. *At that time*, it considers two factors:

- Does this allocation raise the SFS manager's allocation from the specified user storage group beyond the SFS manager's allocation limit?
- Does this allocation raise the user storage group's allocation beyond its allocation limit?

Example:1

The following tables show the size and limits for User Storage Group 3 and the amount of space Manager A, Manager B, and Manager C can allocate from User Storage Group 3:

User Storage Group Limits	**Effective Allocable Space (4K blocks)**
User Storage Group 3	
Physical size (4K blocks)	1,000
Allocation limit (%)	300
Effective allocable space (4K blocks)	3,000

SFS Manager Limits for User Storage Group 3	**Effective Allocable Space (4K blocks)**
Manager A allocable space (4K blocks)	1,000
Manager B allocable space (4K blocks)	1,500
Manager C allocable space (4K blocks)	1,500
Effective allocable space (4K blocks)	4,000

If the physical size of user storage group 3 remains 1,000 blocks and its allocation limit remains 300 percent, at least one of these managers will not be able to allocate his full amount of allocable space. This is because the managers' effective allocable space exceeds the user storage group's effective allocable space.

Example:2

The SFS configuration known to CA VM:Director includes the following:

		SFS manager:	**BGEDDES**
File pool:	VMXA	Allocable file pool:	VMXA
User storage group:	2	User storage group:	2
Allocation limit:	**150%**	Allocation limit:	25000 blocks

SFS manager BGEDDES enrolls user ID SREED in user storage group 2, allocating him 3,000 blocks of space, with this command:

```
vmdirect enroll sreed 3000 2 vmxa
```

CA VM:Director finds the following in processing this command request:

		SFS manager:	**BGEDDES**
File pool:	VMXA	Allocable file pool:	VMXA
User storage group:	2	User storage group:	2
Allocation limit:	150**%**	Allocation limit:	25000 blocks
Current size:	100,000 blocks		
Current allocation:	75,000 blocks	Current allocation:	20,000 blocks
Current allocation limit:	150,000 blocks	Allocation limit:	25,000 blocks

It then calculates the proposed values and determines the request can be filled:

Current allocation limit:	150,000 blocks	Allocation limit:	25,000 blocks
Proposed allocation:	78,000 blocks	Proposed allocation:	23,000 blocks
Less than current allocation limit? YES		Less than current allocation limit? YES	

Chapter 13: Configuring CA VM:Director for SFS

Configuring CA VM:Director for SFS means performing two separate types of functions, describing each to CA VM:Director:

- Configuring your SFS file pools and the user storage groups in those file pools
- Configuring the VM user IDs that will be SFS administrators and SFS managers

This section contains the following topics:

Before You Begin (see page 199)
Describing SFS to CA VM:Director (see page 200)
Resynchronizing the CA VM:Director Configuration of SFS (see page 207)
Defining SFS Administrators and Managers (see page 208)
Granting and Revoking Access to Files Spaces (see page 216)
Migrating Minidisk Files to SFS (see page 217)

Before You Begin

Before you begin the configuration tasks, be sure to address the following items:

- You must provide the CA VM:Director service virtual machine with access to the MAINT 193 minidisk, or this minidisk's equivalent. CA VM:Director needs the SFS utility FILEPOOL EXEC, which normally resides on this minidisk. If you moved this utility to another minidisk, provide the CA VM:Director service virtual machine with access to that minidisk.

  Important! If you removed this utility in the past (before you started using SFS), you must restore it before you can use CA VM:Director for SFS administration.

  Note: During initialization, CA VM:Director reports error 0021E if your site configured CA VM:Director for SFS, and the access mode for MAINT 193, where the FILEPOOL EXEC is found, conflicts with a mode specified for the VMDIRECT product minidisks. Help ensure that the access mode for MAINT 193 is different from any mode specified on an ACCESS record in the PRODUCT CONFIG file.

- User ID VMDIRECT must have ADMIN authorization in the Filepool DMSPARMS file.
- CA VM:Director must be using the Servant Facility to perform some commands on SFS file spaces, such as DELETE.
- If an ESM is being used for SFS authorization, CA VM:Director servants must be able to access any SFS directory read and write

Important! Additionally, you may notice a file named SFS CONFIG on the CA VM:Director service virtual machine 191 minidisk. CA VM:Director alone maintains this file, based on the descriptions and enrollments you add to CA VM:Director. If you edit this file directly you will invalidate the CA VM:Director functions for SFS administration.

Describing SFS to CA VM:Director

You must describe each SFS file pool and all the user storage groups for each of these file pools to CA VM:Director so that CA VM:Director can manage the space. If you have file pools or some user storage groups within file pools that you do not want managed by CA VM:Director, do not describe them to CA VM:Director.

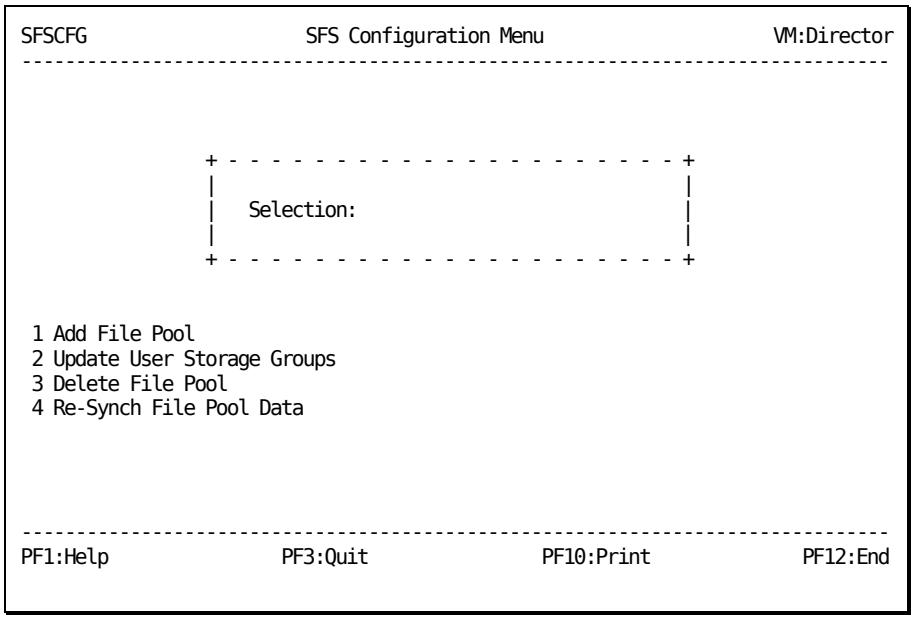
If you change the configuration of SFS itself, for instance by adding or removing file pools or user storage groups, after you have described SFS to CA VM:Director, you must describe those changes to CA VM:Director separately by updating the CA VM:Director configuration for SFS. For adding file pools or user storage groups, this means simply telling CA VM:Director the name of the new file pool or user storage group. For removing file pools or user storage groups, this means removing the file pool or user storage group from the CA VM:Director configuration.

Adding a File Pool and User Storage Groups to the CA VM:Director Configuration

1. From CMS, enter the following:

   ```
   vmdirect config sfs
   ```

   This command displays the SFS Configuration Menu, which contains options for describing the SFS configuration to CA VM:Director.



2. Select option 1, Add File Pool.

   This option displays an entry screen for the name of one file pool you want to describe to CA VM:Director.

3. Enter the name of a file pool to add to the CA VM:Director SFS configuration.

   You must know the name of the file pool to add to CA VM:Director management. CA VM:Director cannot search for the names of file pools it does not know about. Further, although CA VM:Director checks for the file pool when you enter its name here, it cannot distinguish between inactive file pools and nonexistent file pools. You may therefore get a message from CA VM:Director that the file pool you just added is not responding. This does not mean you entered the file pool name incorrectly—the file pool may simply be inactive at the moment or VMDIRECT does not have ADMIN authorization for this filepool.

4. You can add configuration information to CA VM:Director about this file pool one user storage group at a time or all user storage groups at once.

   To add configuration information about one user storage group at a time, continue with Adding One User Storage Group at a Time.

   To add configuration information about all user storage groups in the current file pool, continue with Adding All User Storage Groups.

Adding One User Storage Group at a Time

To add one user storage group and its allocation limit at a time

1. Press Enter.

   Pressing Enter displays a screen where you can add user storage groups to the information CA VM:Director has about your SFS configuration.

2. Press PF2 (Add) to add a user storage group to the CA VM:Director configuration information for the current file pool.

3. On the next screen, enter the number of the new user storage group and an allocation limit.

   This allocation limit is the percentage by which this user storage group can be allocated, compared to its physical size. It must be a whole number from 100 (allocation is limited to the actual physical space) through 9999. For example, if you estimate that users will use no more than half the space they are allocated, you can allocate twice the space you actually have so that the space will be efficiently used (set this allocation limit as 200).

   This limit applies to *all* the SFS managers you allow to allocate space from this user storage group, regardless of which of these managers allocates the space; one SFS manager can allocate all the allocable space, even though you may have allowed five to allocate space from this user storage group.

   Note: For more information about allocation limits for user storage groups, see Allocation of Space in User Storage Groups (see page 195) in the chapter SFS Overview (see page 183).

   After you enter the user storage group number and allocation limit and press Enter, CA VM:Director returns you to the screen that lists the known user storage groups for the current file pool.

4. To add more user storage groups for the current file pool, repeat from Step 4.

5. When you have added all the user storage groups you need to for this file pool, press PF3.

   Doing so returns you to the SFS Configuration Menu. From there, you can either perform other SFS configuration functions for CA VM:Director or return to CMS by pressing PF3 again.

6. To add more file pools, press PF3 until you return to the screen at which you enter the name of a file pool to add, described in Step 3. Repeat from Step 3 for each additional file pool you want to add.

7. When you have added all the file pools you need, press PF3.

   Doing so returns you to the SFS Configuration Menu. From there, you can perform other SFS configuration functions for CA VM:Director or return to CMS by pressing PF3 again.

Adding All User Storage Groups

To update CA VM:Director with the configuration of all the user storage groups available in this file pool and allocation limits for each user storage group, press PF2(Auto-Config).

PF3 tells CA VM:Director to go to SFS and find for itself all the user storage groups in the current file pool. It then adds to its own configuration information about SFS according to what it finds.

Note: For more information about what Auto–Config looks for and what it brings to the CA VM:Director configuration information for SFS, see Automatic Configuration (see page 188) in the chapter SFS Overview (see page 183).

Changing a User Storage Group in a Known File Pool

You can change a user storage group for a file pool already described to CA VM:Director by either adding it or changing its allocation limit.

To do either of these

1. From CMS, enter the following:

   ```
   vmdirect config sfs
   ```

   This command displays the SFS Configuration Menu.

2. Select option 2, Update User Storage Groups.

   This option displays a list of file pools you have described to CA VM:Director.

3. Using the Tab key, move the cursor over the name of the file pool that contains the user storage group to change. Press Enter.

   This displays the list of user storage groups defined for this file pool. The list shows the number of each user storage group and its current allocation limit. It looks similar to this:

User Storage Group	Allocation Limit (%)
2	200
3	1500
5	250
7	300

4. You can either add a new user storage group to the CA VM:Director configuration or change the allocation limit for a user storage group that is already defined:

   - To add a user storage group, press PF2(Add).

     PF2 brings you to the same screen at which you can add user storage groups for file pools you are adding.

     OR

   - To change the allocation limit for a user storage group already described to CA VM:Director, use the Tab key to move the cursor over the user storage group to change, and press Enter.

     Enter takes you to another entry screen. On this screen, in the Allocation limit (%) field, enter the new allocation limit to assign to this user storage group.

     This allocation limit must be a whole number from 100 (allocation is limited to the actual physical space) through 9999.

     When you press Enter to enter the new allocation limit and CA VM:Director accepts that limit, you are returned to the screen that displays all user storage groups for the current file pool.

5. You can do either of two things from this point:

   - To either add a new user storage group to the CA VM:Director configuration or change the allocation limit for a user storage group that is already defined, press PF2 to return to step 4 and repeat the procedure from there.
   - To return to CMS, press PF3 repeatedly.

Deleting a User Storage Group from the CA VM:Director Configuration

Deleting a user storage group from the CA VM:Director configuration for SFS removes that user storage group from administration by CA VM:Director; it does not delete it from SFS itself. If you delete a user storage group, you must make provisions for managing it outside CA VM:Director.

You can delete a user storage group from the CA VM:Director configuration for SFS only if there are no space allocations from it made through CA VM:Director.

If you need to delete a user storage group from a file pool, do the following:

1. From CMS, enter the following:

   `vmdirect config sfs`

   This command displays the SFS Configuration Menu.

2. Select option 2, Update User Storage Groups.
3. Using the Tab key, move the cursor over the name of the file pool that contains the user storage group to delete, and press Enter.

   This displays the list of user storage groups defined for this file pool.
4. Use the Tab key to move the cursor over the user storage group to delete, and press PF4.
   - If there are still allocations made through CA VM:Director for this user storage group, you are not allowed to delete it, and CA VM:Director displays a message telling you this. This screen remains displayed.
   - If this user storage group is available for deleting (it has no allocations in it through CA VM:Director), the following screen asks for confirmation so that you can back out of this request if necessary.
5. Enter yes on the confirmation screen if you want to remove this user storage group from CA VM:Director management.

   If you do not want to remove the user storage group from CA VM:Director management, enter **no**.

   When you press Enter to confirm or reject the deletion and that deletion is accepted, you are returned to the screen that displays all user storage groups for the current file pool.
6. You can do any of several things from this point:
   - To delete more user storage groups from this file pool, return to Step 4 and repeat the procedure from there.
   - To delete user storage groups from another file pool, press PF3 to return to Step 3 and repeat the procedure from there.
   - To return to CMS, press PF3 repeatedly.

Removing a File Pool from the CA VM:Director Configuration

You may want to remove a file pool from the CA VM:Director SFS configuration if you no longer want it managed by CA VM:Director or if you have removed it from the system.

Removing a file pool from the CA VM:Director configuration for SFS deletes that file pool from administration by CA VM:Director; it does not delete it from SFS itself. If you remove a file pool, you must make provisions for managing it outside CA VM:Director.

You can remove a file pool from the CA VM:Director configuration for SFS only if there are no space allocations for it made through CA VM:Director. This means that you must delete all the *FP= records that reference the file pool. If the *FP= record contains only the file pool to delete, delete this record. If the *FP= record references the file pool to delete and another file pool, delete only the reference to the file pool to delete.

Example:

Suppose you want to delete file pool VMXA from the CA VM:Director configuration but leave file pool VMXB. User ID FRED contains this record in the source directory entry. Change the following record:

```
*FP= VMXA VMXB
```

To the following:

```
*FP= VMXB
```

User ID BARNEY contains this record in the source directory entry:

```
*FP= VMXA
```

Delete the entire record.

The easiest way to delete *FP= records is to execute the VMXBKP01 utility to create a USER DIRECT file. Look at this file to come up with a list of all user IDs that have *FP= records that refer to the user ID you want to delete. Then use the VMDIRECT EDIT command to edit the affected user IDs and update or delete the *FP= records as described above.

Data on removed file pools remain owned by the owners who used that space when it was managed by CA VM:Director.

To remove a file pool from the CA VM:Director SFS configuration

1. From CMS, enter the following:

   `vmdirect config sfs`

   This command displays the SFS Configuration Menu.

2. Select option 3, Delete File Pool.
3. Using the Tab key, move the cursor over the name of the file pool to delete, and press Enter.
   - If there are still allocations made through CA VM:Director for this file pool, you are not allowed to delete it, and CA VM:Director displays a message telling you this. This screen remains displayed.
   - If this file pool is available for deleting (it has no allocations in it through CA VM:Director), the following screen asks for confirmation so that you can back out of this request if necessary.

4. Enter yes on the confirmation screen if you want to remove this file pool from CA VM:Director.

   If you do not want to remove the file pool from CA VM:Director administration, enter **no**.

   When you press Enter to confirm or reject the deletion and that deletion is accepted, CA VM:Director returns you to the screen that displays known file pools.

5. You can do either of the following from this point:

   - To delete more file pools, return to Step 3 and repeat the procedure from there.
   - To return to CMS, press PF3 repeatedly.

Querying the Status of File Pools

You can check the availability and location of each file pool on your system that CA VM:Director administers by entering this command:

```
vmdirect query filepool
```

The display in response to this command looks similar to this one:

File Pool	Status	Location
TESTPL05	Available	Local
VMSYSS	Available	Local
VMX002	Unavailable	Unknown
VMXDEVO	Available	Remote

Resynchronizing the CA VM:Director Configuration of SFS

If other people perform user ID administration tasks for SFS outside CA VM:Director, CA VM:Director may be left with an inaccurate picture of SFS. If you know that all SFS–related user ID functions are performed only through CA VM:Director, there will never be a problem with the information CA VM:Director has on SFS. However, if you are unsure of possible SFS allocations made outside CA VM:Director, resynchronize the CA VM:Director configuration of SFS so that it reflects the actual current state of SFS.

1. From CMS, enter the following:

   ```
   vmdirect config sfs
   ```

   This command displays the SFS Configuration Menu.

2. Select option 4, Resynchronize File Pool Data.
3. Using the Tab key, move the cursor over the name of the file pool to resynchronize, and press Enter.

   Note: This command can take a considerable amount of time to execute.

Defining SFS Administrators and Managers

You must define SFS administrators separately from defining file pools to CA VM:Director. You can remove an administrator from CA VM:Director at any time.

You can have CA VM:Director determine who your SFS managers are when you use the Auto–Config function at the time you add file pools to CA VM:Director (see Automatic Configuration (see page 188)). However, you can add more SFS managers later on, even if you have already added all the file pools to CA VM:Director that you need. You can also remove a manager from CA VM:Director or change the scope of a manager's authority at any time.

Adding a New SFS Administrator

You can designate any existing VM user ID as an SFS administrator in CA VM:Director. To do so, add one GRANT record for each user ID that will have SFS administrator authority (authority SFSADMIN) to the CA VM:Director AUTHORIZ CONFIG file.

Note: For more information about user authorizations and granting user authorizations, see Authorizations (see page 105). For more information about the GRANT record and how to use it, see the chapter "Configuration File Reference" in the *Reference Guide*.

SFS administrators have specific authorities that allow them to configure CA VM:Director for SFS and to perform SFS user administration functions.

Note: For more information about the authority and power of SFS administrators, see the chapter SFS Overview (see page 183).

Removing an SFS Administrator

You can remove SFS administrator authority from a user ID by deleting the GRANT SFSADMIN record for that user ID from the CA VM:Director AUTHORIZ CONFIG file.

Adding a New SFS Manager

Any VM user ID that is a directory manager can be made an SFS manager. To designate a user ID as an SFS manager, you must be an SFS administrator; your status as a system administrator is of no consequence.

When you add new SFS managers to CA VM:Director, you also specify the following information:

- A file pool from which they can allocate space to the user IDs they manage.
- A user storage group in this file pool from which they can allocate space.
- The default number of 4 K blocks they will allocate to user IDs in this user storage group.

  The default number of 4 K blocks allows managers to enroll their user IDs more quickly. You can make the default whatever size you think is a reasonable amount of space for most user IDs to use. If the default number of blocks is inappropriate for a particular user ID, that user ID's manager can specify a different number of blocks.
- The maximum number of 4 K blocks they can allocate from this user storage group to all the user IDs they manage

  The managers are allowed to divide this maximum number of blocks among their user IDs in any proportion, or even not use the maximum number of blocks for all the user IDs they manage.

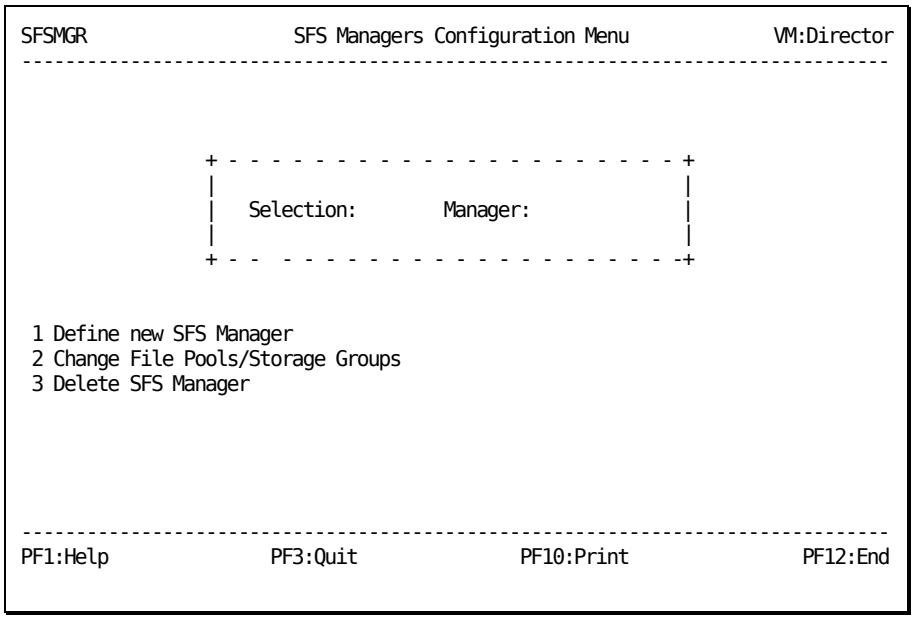
  You can set this allocation limit to any size. CA VM:Director does not check it against the actual size or the allocation limit of the user storage group until this manager actually tries to allocate space from this user storage group.

To add an SFS manager to CA VM:Director

1. From CMS, enter the following:

   `vmdirect admin sfsmgrs`

   This command displays the SFS Managers Configuration Menu.



2. In the Selection field, type **1** to select the Define New SFS Manager option. In the Manager field, enter the user ID you want to make an SFS manager.

   This option displays a list of available file pools, known to CA VM:Director that are available for assignment to this new SFS manager.

3. Using the Tab key, move the cursor over the name of the file pool to authorize the new SFS manager to allocate space from. Press Enter.

   Doing so displays the list of user storage groups defined for this file pool.

4. Using the Tab key, move the cursor over the number of the user storage group to authorize the new SFS manager to allocate from. Press Enter. This displays the Set Storage Group Settings screen.

   This screen lets you define the scope of SFS manager's authority in the user storage group. The top four fields are for display only; they identify the SFS manager and the user storage group for which you are defining authority.

```
SFSSG2                  Set Storage Group Settings                  VM:Secure
--------------------------------------------------------------------------------

                              Manager: MAINT
                            File Pool: IL
                   User Storage Group: 2
                   Current Allocation: 0


   Default Allocation Size (4K Blocks): 0
          Allocation Limit (4K Blocks): 0
       Change to Default Group? (Y/N): Y









--------------------------------------------------------------------------------
PF1:Help             PF3: Quit             PF10:Print             PF12:End
```

5. In the Default Allocation Size (4K Blocks) field, enter the number of 4 KB blocks that this SFS manager should assign to user IDs by default when allocating space from this user storage group. (The SFS manager can override this default size and assign a user ID a different amount of space when allocating space to a user ID.)

   This default allocation size must be a whole number from zero (no allocation) through 4,294,967,295.

6. In the Allocation limit (4K Blocks) field, enter the total number of 4 KB blocks you want this SFS manager to be able to allocate from this user storage group to all user IDs he manages.

   This allocation limit must be a whole number from zero (no limit) through 18,446,744,073,709,551,615.

   This manager's allocation limit works in conjunction with the actual amount of space available and with the user storage group allocation limit when the SFS manager tries to allocate it. Imposing no limit on what the manager can allocate means that the allocations he makes in this user storage group are restricted only by the amount of space actually available for allocation and the allocation limit you specified for this user storage group.

7. In the Change to Default Group? (Y/N) field, enter **n** if you do not want this user storage group to be the one that this manager assigns to a user ID by default *when assigning this file pool to a user ID*. Keep the value as Y if you do want this user storage group to serve as the default for this file pool for this manager.

   Only one user storage group can be the default for a given file pool for a manager. If you authorize a manager to allocate from three user storage groups in the same file pool, and you keep this setting as Y for each of them, only the last group from which you authorize this manager to allocate will be the default user storage group for this manager in this file pool.

8. Press Enter to save this authorization.

   You are returned to the SFS Managers Configuration Menu.

   You can define more authority for this new SFS manager, by:

   - Adding more file pools to the one just added to this SFS manager
   - Adding more user storage groups in the file pool you just added

   Both of these functions are discussed in the following section.

9. Give the SFS manager authority to use the CA VM:Director SFS commands on the user IDs he manages.

   Note: For more information, see Authorization for SFS Managers (see page 128).

   Additionally, if you did not already give the SFS manager MANAGE authority to use all selections on the Manager Selection Menu, give the SFS managers MANSEL10 authority on a GRANT record in the AUTHORIZ CONFIG file.

Reviewing or Changing an SFS Manager's Authority

You can review the file pools and user storage groups in which you authorized your SFS managers to allocate space, and change (add or remove) the file pools in which SFS managers are authorized to allocate space.

To review or change an SFS manager's SFS allocation authority

1. From CMS, enter the following:

   ```
   vmdirect admin sfsmgrs
   ```

   This command displays the SFS Managers Configuration Menu.

2. In the Selection field, type **2** to select the Change File Pools/User Storage Groups option. In the Manager field, enter the user ID whose file pool or user storage group allocation authorities you want to change.

   This option displays a list of file pools in which you authorized this manager to allocate space.

3. You can authorize this manager to use more file pools, delete a file pool from a manager's list of allocable file pools, or review or change the authorizations this manager already has for file pools.

   Note: Reviewing or changing the authorizations a manager has for file pools really means adding or removing user storage groups from which this manager can allocate space, or change the default values for a user storage group in which the manager can already allocate space.

To authorize this manager to use more file pools, continue with Authorizing a Manager to Allocate Space from Other File Pools

To delete file pools from this manager's authority, continue with Deleting File Pools from a Manager's Authority.

To change this manager's authority in a file pool from which he can already allocate space, continue with Changing a Manager's Authority for a File Pool.

Authorizing a Manager to Allocate Space from Other File Pools

To authorize this manager to allocate space from more file pools, press PF2 to display the list of remaining file pools—the ones in which this manager is not yet authorized to allocate space.

On this list, use the Tab key to move the cursor over the name of the file pool in which you want to authorize the SFS manager to allocate space. Press Enter.

The remainder of this procedure is the same as the one you followed for adding the first file pool to this manager's list of allocable file pools.

Deleting File Pools from a Manager's Authority

To delete a file pool from the list of file pools from which this manager can allocate space, move the cursor over the name of the file pool to delete and press PF4.

- If user IDs under this manager still have allocations to this file pool through CA VM:Director, you are not allowed to delete it, and CA VM:Director displays a message telling you this. This screen remains displayed.
- If this file pool is available for deleting from the manager's list of authorized file pools (it has no allocations in it through CA VM:Director by this manager), the following screen asks for confirmation so that you can back out of this request if necessary.

  Enter yes on the confirmation screen if you want to remove this file pool from this manager's list of authorized file pools.

  If you do not want to remove the file pool from this manager's list of authorized file pools, enter **no**.

Changing a Manager's Authority for a File Pool

To change the authorizations this manager already has for a file pool, move the cursor over the name of the file pool to change and press Enter.

The Select Storage Group for a File Pool screen appears. This screen displays details about the user storage groups in the current file pool from which the manager is authorized to allocate space.

```
SFSSG1              Select Storage Group for a File Pool              VM:Director
--------------------------------------------------------------------------------

                              Manager: VMANAGER
                            File Pool: IL

                              Default          Allocation       Allocation
            User          Allocation Size        Limit            Used
         Storage Group      (4K Blocks)       (4K Blocks)      (4K Blocks)
 Default->       2               150                0              500




             Use the TAB key to position the cursor to the Group.
                   PF2:Add     PF4:Delete     ENTER:Change


--------------------------------------------------------------------------------
 PF1:Help      PF3:Quit      PF7:Back      PF8:Frwd      PF10:Prt      PF12:End
```

You can authorize this manager to use more user storage groups in this file pool, delete a user storage group from the manager's list of allocable user storage groups in this file pool, or review or change the authorizations this manager already has for user storage groups in this file pool.

To authorize this manager to use more user storage groups in this file pool, continue with Authorizing a Manager to Allocate Space from Other User Storage Groups.

To delete file pools from this manager's authority, continue with Deleting User Storage Groups from a Manager's Authority.

To change this manager's authority for a user storage group from which the manager can already allocate space, continue with Changing a Manager's Authority for a User Storage Group.

Authorizing a Manager to Allocate Space from Other User Storage Groups

To authorize this manager to allocate from more user storage groups, press Tab to display the list of remaining user storage groups—the ones in which this manager is not yet authorized to allocate space.

On this list, use the Enter key to move the cursor over the number of the user storage group in which you want to authorize the SFS manager to allocate space. Press PF4 to display an entry screen.

The remainder of this procedure is the same as the one you followed for adding the first user storage group to this manager's list of allocable user storage groups for this file pool. See those instructions if you need a reminder on how to add user storage groups to a manager.

Deleting User Storage Groups from a Manager's Authority

To delete a user storage group from the list of user storage groups in which this manager can allocate space, move the cursor over the number of the user storage group to delete and press PF4.

- If user IDs under this manager still have allocations to this user storage group through CA VM:Director, you are not allowed to delete it, and CA VM:Director displays a message telling you this. This screen remains displayed.
- If this user storage group is available for deleting from the manager's list of authorized user storage groups (it has no allocations in it through CA VM:Director by this manager), the following screen asks for confirmation so that you can back out of this request if necessary.

  Enter yes on the confirmation screen if you want to remove this user storage group from this manager's list of authorized user storage groups.

  If you do not want to remove the user storage group from this manager's list of authorized user storage groups, enter **no**.

Changing a Manager's Authority for a User Storage Group

To change the authorizations this manager already has for a user storage group, move the cursor over the number of the user storage group to change and press Enter. This displays an entry screen.

The remainder of this procedure is the same as the one you followed for adding the first user storage group to this manager's list of allocable user storage groups for this file pool. See those instructions if you need a reminder on how to add user storage groups to a manager.

Deleting an SFS Manager

Deleting an SFS manager removes that user ID's authority to perform any SFS administration functions, such as enrolling user IDs in SFS and removing file space from SFS users. It does not affect that user ID's status or authorizations for any other functions, either inside or outside SFS administration.

You can delete an SFS manager only if that manager is not managing any user IDs in SFS. If there are user IDs enrolled in SFS under a manager you want to delete, you must first reassign those user IDs to another SFS manager. You can do so with the ASSIGN command; see the *Reference Guide*. If these user IDs are not using any file space, you do not have to reassign them to another manager but they must be enrolled under another SFS manager when they want to use SFS.

To delete an SFS manager

1. From CMS, enter the following:

   ```
   vmdirect admin sfsmgrs
   ```

   This command displays the SFS Managers Configuration Menu.

2. In the Selection field, type **3** to select the Delete SFS Manager option. In the Manager field, enter the user ID you want to delete as SFS manager.

   The following screen asks for confirmation so that you can back out of this request if necessary.

3. Enter yes on the confirmation screen if you want to delete this user ID's SFS management authority from CA VM:Director.

   If you do not want to remove the user ID's SFS management authority from CA VM:Director, enter **no**.

4. To remove more SFS managers, repeat from step 2.

5. When you have removed all the SFS managers you need to, press PF3 to return to CMS.

Granting and Revoking Access to Files Spaces

Using the CA VM:Director GRANT AUTHORITY and REVOKE AUTHORITY commands, an authorized user can grant and revoke access to files and directories in one or more SFS file spaces in any file pool controlled by CA VM:Director. The issuing user does not need to own the file spaces or have SFS administrator authority.

The CA VM:Director GRANT AUTHORITY and REVOKE AUTHORITY command syntax is identical to the syntax of the CMS GRANT AUTHORITY and REVOKE AUTHORITY commands.

Note: For an explanation of the GRANT AUTHORITY and REVOKE AUTHORITY commands and how to use them, see the chapter "Command Reference" in the *Reference Guide*.

Preparing to Use GRANT AUTHORITY and REVOKE AUTHORITY Commands

- Give the CA VM:Director service virtual machine SFS administrator authority on an ADMIN record in the DMSPARMS configuration file of every file pool for which GRANT AUTHORITY and REVOKE AUTHORITY commands will be issued.
- Give GRANT AUTHORITY and REVOKE AUTHORITY authorization to the users who will be issuing the commands.

  Note: For information about granting authorizations to the GRANT AUTHORITY and REVOKE AUTHORITY commands, see the chapter Authorizations (see page 105).

- Enable the CA VM:Director Servant Facility.
- CA VM:Director uses a servant to issue the GRANT AUTHORITY and REVOKE AUTHORITY commands. Additionally, CA VM:Director dynamically gives to (and revokes from) the servant the required SFS administrator authorization.
- If required, verify CA VM:Director has access to the DMSJNE file.

  DMSJNE is an optional customer–written routine that resolves user IDs on other systems to a local user ID when a nickname is specified on a command. If your site uses the DMSJNE routine, CA VM:Director must have access to it because CA VM:Director calls DMSJNE to help resolve nicknames and user IDs on other systems.

  Note: For additional information about DMSJNE and nickname resolution, see the appropriate IBM *SFS and CRR Planning and Administration* documentation.

Migrating Minidisk Files to SFS

If you are migrating from minidisk–based file storage to SFS–based file storage, you can have CA VM:Director help you move your data from minidisks to SFS. In this move, you can also delete the minidisks from users' directory entries after the minidisks are processed; this prevents inadvertent use of duplicate data. You can also enroll these users as SFS users under CA VM:Director if they are not already enrolled.

Migrating data from one storage type to another, deleting old minidisks, and enrolling user IDs in SFS through CA VM:Director is possible through the MOVE2SFS command.

Note: For information about the MOVE2SFS command, including all its options and how to use them, see the chapter "Command Reference" in the *Reference Guide*.

You can also specify where you want the data from minidisks placed when it is moved to SFS, and you can manipulate the data in some ways after it is moved. These two functions are available using the PREMOVE and POSTMOVE user exits.

Note: For information about using the PREMOVE user exit and the POSTMOVE user exit for specifying the location of migrated data and for other manipulations you can make on migrated data, see the chapter "User Exit Reference" in the *Reference Guide*.

Chapter 14: Improving CA VM:Director Performance

To improve performance on virtual machines

- Use the CP SET SRM command to tune CP performance–related system parameters. For more information about the SET SRM and QUERY SRM commands, see the appropriate IBM CP system command reference.
- Use the SET SHARE command (or the equivalent SHARE directory control statement) to give the CA VM:Director service virtual machine a higher priority in system processing.

  A high number gives the CA VM:Director service virtual machine a high priority on the system. The system performs CA VM:Director requests before other service virtual machine commands with lower numbers.

  Note: For more information about the SET SHARE command, see the appropriate system command reference.

This section contains the following topics:

Service Virtual Machine Performance (see page 219)
Performance with SFS (see page 220)

Service Virtual Machine Performance

To improve performance on the CA VM:Director service virtual machine

- Rewrite heavily used user exits in Assembler for faster execution.
- Use the MACLOAD record in the PRODUCT CONFIG file for frequently used macros, or better yet, all macros.
- Do not create macros that issue long–running commands on the CA VM:Director machine. Instead, run the commands on the issuing user ID's machine. When a process starts on CA VM:Director it runs until the process is complete or until it prompts a user for information. At that point, the next process in the queue is started.
- Use the AUDIT user exit to decrease the number of audit records that CA VM:Director creates.

Performance with SFS

To improve CA VM:Director performance when using SFS, include the following macro names on a MACLOAD record in the PRODUCT CONFIG file:

- SFSCFG
- SFSCSL
- SFSSUS
- SFSPING
- SFSSFP

Chapter 15: Macros

CA VM:Director implements many commands and functions with REXX routines named *macros*. These macros call primitive functions written in Assembler to manipulate a directory and a user's virtual machine by issuing commands to the command environment that CA VM:Director creates.

Tracing Macros

When a CA VM:Director command does not complete execution normally, you can find out where the associated macro failed by using the TRACE command. The TRACE command lets you trace the execution of a macro. This debugging information is valuable if you need to call Technical Support to report macro errors.

To use the TRACE command, you must have the appropriate authorization in the AUTHORIZ CONFIG file.

To enter the following command, you must have at least TRACE MACRO ULIST authorization:

```
vmdirect trace macro ulist
```

Any of the following records in the AUTHORIZ CONFIG file can give you the required authorization. Replace *userid* with the userid who will issue the TRACE command:

```
GRANT TRACE MACRO ULIST TO userid
GRANT TRACE MACRO TO userid
GRANT TRACE TO userid
```

To trace a macro

1. Spool your console to yourself:

   ```
   cp spool console start to *
   ```

   This step saves the trace so you can analyze it later.

2. Enter the TRACE command with the optimized macro name that you want traced. For example, to trace the ULIST command when issued by user ID TESTUSR and send the trace information to user ID MONUSR, enter the following command:

   ```
   vmdirect trace macro ulist user testusr to monusr
   ```

3. Note the trace number that is provided in the command response.

4. Enter the command to trace. For example, to list all the information about user ID OPERATOR, enter the following:

   ```
   vmdirect ulist operator
   ```

5. When the command fails, stop the trace by entering TRACE STOP followed by the trace number from step 3. For example, to stop trace 3, enter the following command:

   ```
   vmdirect trace stop 3
   ```

6. Close your console and send the output to your reader by entering:

   ```
   cp spool console close
   ```

Chapter 16: Security Features

This chapter describes the CA VM:Director security features available to you.

These features include:

- Verifying passwords with another security system
- Verifying directory link statements
- Verifying minidisk link modes and passwords
- Verifying logon passwords

This section contains the following topics:

Verifying Passwords with Another Security System (see page 223)
Verifying Directory Link Statements (see page 223)
Verifying Minidisk Link Modes and Passwords (see page 224)
Verifying Changes to Logon Passwords (see page 224)

Verifying Passwords with Another Security System

To verify logon passwords that users enter while executing CA VM:Director commands, use the CHKPASS user exit. This user exit is essential for sites that use security packages where the logon password is not stored in the CP directory. The user exit coding can verify for valid passwords using the security package, instead of looking in the source directory for the user ID's password.

Note: For more information about the CHKPASS user exit, see the chapter "User Exit Reference" in the *Reference Guide*.

Verifying Directory Link Statements

To verify a directory link to a virtual machine, use the LINK user exit. This user exit can prevent users from defining directory links to user IDs you consider sensitive, such as MAINT or OPERATOR.

Note: For more information about the LINK user exit, see the chapter "User Exit Reference" in the *Reference Guide*.

Verifying Minidisk Link Modes and Passwords

To accept or reject changes made to a minidisk's access mode and link passwords, use the MINIDISK user exit. This user exit can secure minidisks that you consider sensitive or essential.

Note: For more information about the MINIDISK user exit, see the chapter "User Exit Reference" in the *Reference Guide*.

Verifying Changes to Logon Passwords

To verify specific logon password changes, use the PASSWORD user exit. This user exit lets you define standards for how many characters a password can have and what those characters can be.

Note: For more information about the PASSWORD user exit, see the chapter "User Exit Reference" in the *Reference Guide*.

Chapter 17: VMDIRECT MANAGERS File Reference

You define whom your directory managers are and describe the range of their authority to CA VM:Director through the VMDIRECT MANAGERS file. Each directory manager user ID at your site is represented by a group of records in this file. This group of records can include any of the records summarized in the table that follows.

Note: In addition to defining directory managers in this file, you must also authorize these directory managers to use certain CA VM:Director commands, such as ADDENTRY, so that they can use the specifications you enter here. For instance, directory managers cannot use the default device type (which you specify on a DEVTYPE record in this file) unless you let the directory managers create user IDs. You do so by authorizing the directory managers to use the ADDENTRY command.

Note: For more information about creating and enabling directory managers, see Directory Managers (see page 43).

This section contains the following topics:

VMDIRECT MANAGERS File Record (see page 225)
DEFSIZE Record (see page 228)
DEVTYPE Record (see page 229)
IDLIMIT Record (see page 230)
MANAGER Record (see page 231)
SKELETON Record (see page 232)

VMDIRECT MANAGERS File Record

The following records define the limits of a manager's authority and the default values that CA VM:Director uses when that directory manager performs directory maintenance functions.

Record	Purpose	Use
DEFSIZE	Specifies a directory manager and the default minidisk size that directory manager will allocate.	Optional

Record	Purpose	Use
DEVTYPE	Specifies a directory manager and the directory manager's default device type for allocating minidisks.	Optional
IDLIMIT	Specifies a directory manager and maximum number of user IDs that directory manager can manage.	Optional
MANAGER	Specifies a directory manager and its default subpool from which to allocate DASD.	Required
SKELETON	Specifies a directory manager and its default directory skeleton.	Required

A record in the VMDIRECT MANAGERS file consists of a record name followed by parameters. Records having an asterisk as the first non–blank character are comments; except for the *ED= special comment. Records beginning with asterisks are ignored for processing. Blank lines are also ignored.

You can include an edit special comment (*ED=) in the VMDIRECT MANAGERS file.

CA VM:Director maintains this record with the following information:

- Date and time of last update of the VMDIRECT MANAGERS file
- User ID that last updated the file
- Process used to update the file
- Date the file was first updated

The VMDIRECT MANAGERS file physical line length is 71 characters. The maximum logical line length, a record continued over several lines, in the VMDIRECT MANAGERS file is 255 characters.

If you need to make a record longer than 71 characters:

- Split it over as many physical lines as you need, breaking it between record words.
- Place a comma at the end of each physical line. The comma is the record continuation character and continues the record to the next line that contains text.

  This is true even if the next line begins with an asterisk, which is otherwise the symbol for a comment line.

Example:

The next two lines are read as one record, and not a record and a comment line. They are equivalent to the third line.

```
MANAGER CLYDE,
* MAIN
MANAGER CLYDE * MAIN
```

Note: You cannot extend a comment line (a line beginning with an asterisk). If you have a comment that extends beyond one line, make two comment lines.

CA VM:Director reads the VMDIRECT MANAGERS file as part of its initialization process. If it encounters an invalid MANAGER or SKELETON record, it sends a diagnostic message to the CA VM:Director console and to the CA VM:Director system operator. This operator is a user ID you specify on the SYSOPER record in the PRODUCT CONFIG file.

If CA VM:Director encounters a MANAGER record that specifies a user ID that does not appear in the directory, initialization is terminated with the appropriate error messages.

You can edit the VMDIRECT MANAGERS file while CA VM:Director is running using the ADMIN MANAGERS command.

The Sample VMDIRECT MANAGERS File, shows a sample VMDIRECT MANAGERS file. According to this sample file:

- DIANEC can allocate an unlimited amount of disk space in the MAIN subpool, as indicated by the asterisk on the MANAGER record
- She is authorized to use the GENERAL skeleton file
- Her default device type is 3390 and displays on the Manager Selection screens
- She can manage up to 10 user IDs at any given time
- When DIANEC uses the Application Program Interface (API) commands and automatic disk allocation, the default size of a new minidisk is 1 MB

The other managers' authorizations are apparent after reading those for directory manager DIANEC in the entry which follows:

```
*ED=
* THIS MANAGERS FILE DEFINES THE DIRECTORY MANAGERS FOR THE XYZ
* DIVISION
MANAGER VMANAGER * POOL1 MAIN
   SKELETON VMANAGER GENERAL
   DEVTYPE VMANAGER 3390
   IDLIMIT VMANAGER 65525
   DEFSIZE VMANAGER 2M
MANAGER DIANEC * MAIN
   SKELETON DIANEC GENERAL
   DEVTYPE DIANEC 3390
   DEFSIZE DIANEC 1M
   IDLIMIT DIANEC 10
MANAGER JOE * FINAPOOL
   SKELETON JOE FINANCE
   DEVTYPE JOE 3390
   IDLIMIT JOE 8
   DEFSIZE JOE 2M
MANAGER MARY *DEVLPOOL
   SKELETON MARY DEVELOP
   DEVTYPE MARY 3380
   DEFSIZE MARY 2M
MANAGER DAVE * POOL5
   SKELETON DAVE GENERAL
   DEVTYPE DAVE 3370
   IDLIMIT DAVE 65525
   DEFSIZE DAVE 23M
```

DEFSIZE Record

Assigns a directory manager a default minidisk allocation size.

DEFSIZE *mgrid size*

Definitions

mgrid

The directory manager user ID assigned this default minidisk allocation size.

size

The directory manager's default minidisk allocation size. This size can be any integer of kilobytes (follow this integer with K) or megabytes (follow this integer with M).

Description

The DEFSIZE record specifies the default allocation size directory managers assign when they allocate a new minidisk. CA VM:Director uses this default minidisk allocation size when a directory manager defines or changes a minidisk using the ADDENTRY command (using the ADDMDISK command), the ADDMDISK command, the CHGMDISK command, and the REPENTRY command (using the CHGMDISK command).

This default allocation size usually saves time because directory managers generally allocate new minidisks of the same size. However, directory managers can override this default size by providing a different size at the time they allocate a new minidisk if another size is required.

The DEFSIZE record must follow the MANAGER record of the same directory manager user ID. This record is optional, and there is no default value.

Note: For more information, see the following sections in the chapter "Command Reference" in the *Reference Guide*:

- ADDENTRY Command for information about creating user IDs.
- ADDMDISK Command for information about creating minidisks.
- CHGMDISK Command for information about changing minidisks.
- REPENTRY Command for information about replacing user IDs.

DEVTYPE Record

Assigns a directory manager a default allocation device type when using the MANAGE command to create new minidisks.

DEVTYPE *mgrid devtype*

Definitions

mgrid

The directory manager user ID to which this default device type is assigned.

devtype

The directory manager's default device type shown on the Define a New Minidisk screen through the MANAGE command. This can be any currently z/VM supported minidisk device type.

Description

The DEVTYPE record specifies the default device type that directory managers allocate from when allocating a new minidisk using the Define a New Minidisk screens using the MANAGE command.

A directory manager can override this default device type by providing a different one in the TYPE field in the Define a New Minidisk screen.

CA VM:Director does not use the DEVTYPE record when a directory manager changes an existing minidisk using the Move a Minidisk option in the MANAGE screens. The Device Type field must be filled in. The DEVTYPE record is not used when minidisks are changed using the ADDMDISK or CHGMDISK commands. The directory manager must provide the device type or take the default based on the location determined by the subpool.

The DEVTYPE record must follow the MANAGER record. The DEVTYPE record is optional. There is no default value. If a directory manager has no default device type, device type fields are left blank on any screens that the directory manager uses. Based on the location of the allocation, the device type is determined by the subpool used.

IDLIMIT Record

Limits the number of user IDs that a directory manager can manage.

`IDLIMIT` *`mgrid limit`*

Definitions

mgrid

The directory manager user ID to which to assign this limit.

limit

The maximum number of user IDs that the directory manager can own. This limit can be any integer from 1 through 99999999.

Description

The IDLIMIT record limits the number of user IDs that a directory manager can own at any given time.

This limit applies to the following user IDs:

- User IDs this directory manager creates
- User IDs someone else creates for this directory manager

- User IDs anyone assigns to this directory manager using the ASSIGN command

The DEFSIZE record must follow the MANAGER record of the same directory manager user ID.

Note: This record is optional, and there is no default value. If you do not create an IDLIMIT record for a directory manager, the directory manager can own an unlimited number of user IDs.

MANAGER Record

Use the MANAGER record to identify a user ID as a directory manager.

```
MANAGER mgrid limit subpool1 ... subpooln
```

Definitions

mgrid

The user ID to define as a directory manager.

limit

The directory manager's limit on total minidisk space allocation. This limit can be any integer of kilobytes (follow this integer with K) or megabytes (follow this integer with M), or an asterisk. This means there is no limit on the directory manager's disk space allocation other than that imposed by available DASD.

subpool1 ... subpooln

The list of subpools from which the directory manager can allocate space. These subpools must be defined by a SUBPOOL record in the DASD CONFIG file. The first subpool in the list is the directory manager's default subpool.

Description

The MANAGER record identifies which user ID is a directory manager, that directory manager's limit on the total amount of minidisk space the directory manager can have allocated at any time, and the subpools from which the directory manager can allocate disk space.

The amount of space allocated by a directory manager is incremented according to the default block size defined for the device type on which the space resides.

The MANAGER record must precede all other records in the VMDIRECT MANAGERS file for a given directory manager user ID. This record is required, and there are no default values.

Note: For more information about the DASD CONFIG file, see Configuration Files and Records (see page 49). For information about defining subpools to CA VM:Director, see SUBPOOL Record in the chapter "Configuration File Reference" in the *Reference Guide*.

SKELETON Record

Use the SKELETON record to assign a directory manager the skeleton files to create user IDs.

SKELETON *mgrid skeleton1 ... skeletonn*

Definitions

mgrid

The directory manager user ID to which to assign these skeleton files.

skeleton1 ... skeletonn

The list of skeleton files the directory manager can use to create user IDs. These skeleton files must already exist on the CA VM:Director user ID's 191 disk. The first skeleton file in the list is the directory manager's default skeleton file.

Description

Skeleton files are template directory entries. You can create multiple skeleton files to define user IDs with different virtual machine configurations. The SKELETON record specifies which skeleton files a directory manager can use to create user IDs.

Only one SKELETON record is valid in the VMDIRECT MANAGERS file for each user ID defined as a directory manager.

The SKELETON record must follow the MANAGER record of the same directory manager user ID. This record is required, and there are no default values.

Note: For more information about the contents and purpose of skeleton files, see Skeleton Files (see page 165).

Chapter 18: Special Comment Reference

CA VM:Director recognizes several special comments that are not comments in the usual sense. While the usual comments are physical lines with notes or instructions for human use that are not processed by CA VM:Director or by CMS, these special comments are processed by CA VM:Director.

Some of these special comments are created by CA VM:Director, either automatically because of system conditions or in response to administrative or management functions you used. Others you must add to the appropriate user source directory, directory profile, or skeleton file.

CA VM:Director sets the values for many of these special comments. Some of the values it sets are in response to user actions. Other values it sets are in response to commands that you, another system administrator, or a directory manager issues.

Note: Do not edit a special comment directly.

Summary of CA VM:Director Special Comments

Comment types, where they are allowed, and how they are added and maintained are listed in the following table. Their functions are described in the following sections.

Special Comment	Allowed In				Added to File By	Value Set By
	MANAGERS File	Skeleton Files	Directory Profiles	Directory Entries		
*AC= account comment		Y	Y	Y	System administrator	System administrator
*ED= edit history comment	Y	Y	Y	Y	System administrator VMXGNR utility	CA VM:Director (only when you use CA VM:Director commands to modify the directory entry)
*ENROLL enroll comment		Y			System administrator	System administrator

Special Comment	Allowed In				Added to File By	Value Set By
	MANAGERS File	Skeleton Files	Directory Profiles	Directory Entries		
*EX= expire comment				Y	CA VM:Director (when someone expires a user ID with the EXPIRE command)	CA VM:Director (when someone expires a user ID with the EXPIRE command)
*FP= file pool comment				Y	CA VM:Director	CA VM:Director (based on the file pools in which the user ID is enrolled)
*PW= password history comment		Y		Y	System administrator VMXGNR utility	CA VM:Director
*PWnn= Traditional password values				Y	CA VM:Director	CA VM:Director
*SP= sample profile comment		Y		Y	System administrator	System administrator (CA VM:Director removes this comment from a directory entry when that user ID's 191 minidisk is created or when that user ID is enrolled in SFS and the IPL statement is updated with the file pool name)

*AC= [Account Number]

Lists the account numbers a user ID is allowed to use. Specify these account numbers in the following format when you add a *AC= comment to a directory profile or skeleton file:

*AC= *accountnum ...*

accountnum ...

List of account numbers that user IDs created by way of the skeleton file containing this comment will be able to use. For directory profiles, this is a list of account numbers that the user ID using this directory profile can use.

You can specify as many account numbers as will fit in 255 characters, the limit on logical line length for comments.

A change in the account number that a user ID tries to use is validated against those listed in the *AC= comment or specified by the ACCOUNT user exit. If both exist for a user ID, the account numbers listed in its *AC= comment take precedence.

*ED= [Edit]

Lists when the most recent change to a directory entry:

- When it was made (date and time)
- Who made the change (name of the user ID)
- How it was done (which macro was invoked for the update)

You, or a directory manager, add this comment to the MANAGERS file or a directory entry for which you need to know who makes changes and when those changes are made, or to a skeleton file so that it gets incorporated in directory entries for which you must know this information, in the following format:

*ED=

CA VM:Director updates the values stored in this comment each time a file or directory entry in which it appears is edited using an CA VM:Director command.

aa

The information CA VM:Director maintains in this comment looks like this:

```
*ED=yy/mm/dd  14:33:54  VMANAGER  EDIT  yy/mm/dd
```

yy/mm/dd

The date when the file in which this comment appears was last edited.

14:33:54

The time when this file was last edited.

VMANAGER

The user ID that last edited this file.

EDIT

The command used for the last edit to this file.

yy/mm/dd

The date when *ED= was added to this file.

*END= [End]

Delineates the end of a directory profile or source directory entry. It is automatically added by the VMXBKP01 utility, and automatically deleted by VMXGNR utility.

```
*END
```

*ENROLL= [Enroll in SFS]

List the file pool, user storage group, and space allocation with which a user ID is enrolled in SFS. Specify these enrollment features in the following format when you add a *ENROLL comment to a skeleton file:

```
*ENROLL [blocks [userstoragegroup [filepool]]] [(IPLstmt]
```

blocks

Number of 4 KB blocks to allocate in SFS to a user ID created by way of a skeleton file containing this comment. The default number of blocks, or the number of blocks allocated if you specify it as an asterisk, is your default enrollment number of blocks.

userstoragegroup

Number of the SFS user storage group in which to allocate space to a user ID created by way of a skeleton file containing this comment. The default user storage group, or the number of the user storage group if you specify it as an asterisk, is your default enrollment user storage group.

filepool

Name of the SFS filepool in which to enroll a user ID created by way of a skeleton file containing this comment. The default filepool is your default enrollment filepool.

IPLstmt

Updates the IPL statement in the user ID's directory entry to make this file pool the user ID's default one (active when a user logs on to this user ID). If a sample profile is available or specified on a *SP= special comment in the directory entry, that profile is copied to the top–level directory for the user as PROFILE EXEC.

You must specify one *ENROLL comment for each file pool in which you want new user IDs initially enrolled. However, you or these user IDs' SFS managers can enroll these user IDs in other file pools at a later time.

When you create user IDs by way of a skeleton file that contains a *ENROLL comment, the file pools in which you enroll them are immediately reflected in the directory entry for each new user ID in its *FP= comment. For more information about this comment, see *FP= (FilePool) in this chapter.

Note: For information about default enrollment values, see the chapter SFS Overview (see page 183).

*EX= [Expire]

Lists the date and time when a user's logon password was expired through the EXPIRE command, and who issued the EXPIRE command (name of the user ID) against that user ID.

The server adds this comment to the directory entry of a user ID whose logon password was expired. It removes this comment when the user ID's expired logon password is changed.

Note: Do not enter values for this comment.

Note: For more information about the EXPIRE command, how it works, and how you can use it, see the chapter "Command Reference" in the *Reference Guide*.

*FP= [File Pool]

Lists the SFS file pools in which a user ID is enrolled.

CA VM:Director itself adds this comment to the directory entry of each user ID enrolled in SFS. It updates this comment when you or the user ID's SFS manager changes the file pools in which the user ID is enrolled. You must not enter values for this comment.

Specify these file pools in several ways for user IDs, and you can use more than one way to enroll a given user ID in file pools or remove a given user ID from file pools:

- Use the *ENROLL comment in a skeleton file that you use to create user IDs.
- Use the ENROLL and MANAGE commands on the user ID directly.

*PW= [Password Change Date]

Lists the date when a user's password was most recently changed.

In previous releases, this record also contained password history data. Any password history data will be removed from this record whenever a user's password is next changed. Password history data may have been encrypted. The history data will be moved to the *PW*nn*= records described below.

Add this comment, in the following format, to a directory entry for which you need to know old passwords and the most recent password change date, or to a skeleton file so that it is incorporated in directory entries for which you must know this information:

```
*PW=
```

CA VM:Director updates the date stored in this comment in a user directory entry each time a logon password for this user ID is changed.

Note: Do not enter values for this comment.

The information CA VM:Director maintains in this comment looks like this:

```
*PW= yy/mm/dd
```

yy/mm/dd

The date on which the user ID of this directory entry last changed his password.

*SP= [Sample Profile]

Lists a PROFILE EXEC file to give to a user ID when a 191 minidisk is defined or when a new user is enrolled in SFS and the IPL statement will be updated with the file pool name. Specify the profile file name in the following format when you add a *SP= *comment* to a directory entry or skeleton file:

```
*SP= filename filetype
```

filename filetype

Name and type of the PROFILE EXEC to assign to a user ID created by way of a skeleton file containing this comment.

CA VM:Director removes the *SP= comment from a user's directory entry when a 191 minidisk is created for that user ID or when that user ID is enrolled in SFS and the IPL statement is updated with the file pool name, because its information is no longer required.

If the specified sample profile can be found, CA VM:Director copies the profile to the 191 minidisk or top-level directory. If no *SP= special comment is found, CA VM:Director behaves as if *SP= SAMPLE PROFILE was specified. CA VM:Director looks for the specified or default profile as follows:

- First, on your A-disk (or the A-disk of the user ID adding the 191 minidisk or enrolling the user ID)
- Second, on any of the CA VM:Director minidisks in CMS search order. If CA VM:Director cannot find the profile, no profile is copied

To copy any sample profile to an SFS directory, whether specified on a *SP= special comment or defaulted, the Servant Facility must be active. This requirement is not necessary for 191 minidisks.

Copying a sample profile depends on whether you are copying to a 191 minidisk or to an SFS file space:

- When copying to a 191 minidisk:

  CA VM:Director copies the sample profile whenever the minidisk is created.

- When copying to an SFS file space:
  - If a user ID is being enrolled, either explicitly or implicitly, and the IPL statement is going to be updated with the file pool name, CA VM:Director copies the sample profile to the new top-level directory.
  - When the MOVE2SFS command is used and the IPL statement is updated, CA VM:Director copies the sample profile to the top-level directory (which may not be new), regardless of the destination directory of the minidisk being migrated to SFS. The user may already be enrolled in the file pool, and the top-level directory may already have files in it. If the PROFILE EXEC file already exists, no sample profile is copied.

These rules for adding a PROFILE EXEC to a user ID's new 191 minidisk also apply to the directory entries CA VM:Director creates using the ADDENTRY command with the NOSKEL option. In that case, CA VM:Director uses the directory entry specified by the command in place of a skeleton file.

Chapter 19: VMDIRECT GLOBALS File Reference

The VMDIRECT GLOBALS file supports the global definitions (GLOBALDEFS) section in the CP object directory introduced in VM/ESA 2.1.0.

This section contains the following topics:

VMDIRECT GLOBALS File (see page 241)
Changing the Default Machine Mode (see page 242)

VMDIRECT GLOBALS File

The VMDIRECT GLOBALS file resides on the CA VM:Director DRCT minidisk and contains the global system settings. Use this file, instead of the USER DIRECT file, to specify default settings that can apply to all users on the system.

Each GLOBALOPT record in the VMDIRECT GLOBALS file identifies a specific setting.

Note: For information about the format and use of the GLOBALOPT record, see the appropriate IBM *Planning and Administration* documentation.

The VMDIRECT GLOBALS file can also contain comments, blank lines, and the *ED= special comment.

You can include an edit special comment (*ED=) in the VMDIRECT GLOBALS file. CA VM:Director maintains this comment with the date and time of last update of the VMDIRECT GLOBALS file, the userid that last updated the file, the process used to update the file, and the date the file was first updated.

CA VM:Director reads the VMDIRECT GLOBALS file as part of its initialization process. If it encounters an invalid statement, it sends a diagnostic message to the CA VM:Director console and to the CA VM:Director system operator. This operator is a userid you specify on the SYSOPER record in the PRODUCT CONFIG file.

You can edit the VMDIRECT GLOBALS file while CA VM:Director is running by using the ADMIN GLOBALS command. To use this command, you must have at least ADMIN GLOBALS authorization through a GRANT record in the AUTHORIZ CONFIG file. You can also use the ADMIN, ADMIN *, or ADMIN *ALL authorizations to provide the necessary level of security.

Changing the Default Machine Mode

To change the default machine architecture for any USER definition that does not contain a MACHINE statement, use the ADMIN GLOBALS command. This command opens the VMDIRECT GLOBALS file in XEDIT.

Note: You must have at least ADMIN GLOBALS authorization to use the ADMIN GLOBALS command.

For information about the MACHINE global definition, see the appropriate the appropriate IBM *Planning and Administration* documentation. For more information about the ADMIN GLOBALS command, see the *Reference Guide*.

Example:

To change the default machine mode, enter the following:

```
vmdirect admin globals
```

The VMDIRECT GLOBALS file opens in XEDIT. If there already is a GLOBALOPT MACHINE statement, the current definitions display.

Edit (or add) a GLOBALOPTS statement to change the machine mode from 370 to ESA:

```
GLOBALOPT   MACHINE   ESA
```

Save and exit the file.

Chapter 20: POSIX Support in CA VM:Director

CA VM:Director supports the following POSIX objects in the CP object directory:

POSIX Object	Definition
POSIX UID	POSIX user ID. An integer that represents the VM user ID to the POSIX system.
POSIX GID	POSIX group ID. An integer that represents the POSIX group in which the VM userid is a member.
POSIX GNAME *gname*	*gname* is a character string that represents a GID. Many *name*s can represent the same GID.
POSIX supplementary group name list	A list of GIDs and GNAMEs to which the POSIX userid is eligible to be a member.
POSIX file system root, initial program (shell), and initial working directory	Various POSIX configuration items.

This section contains the following topics:

Maintaining POSIX Group Names and Group IDs Using CA VM:Director (see page 243)
POSIX Information in a Directory Entry or Profile (see page 244)
VMDIRECT POSIX File (see page 245)

Maintaining POSIX Group Names and Group IDs Using CA VM:Director

CA VM:Director keeps POSIX group information in the VMDIRECT POSIX file on the DRCT minidisk. This file enables you to use CA VM:Director to define new POSIX groups, and to make changes to existing POSIX groups. You use this file to define POSIX groups instead of specifying POSIXGROUP statements in the USER DIRECT file.

To work with POSIX group names, use the ADMIN POSIX command. This command opens the VMDIRECT POSIX file in XEDIT.

Note: You must have at least ADMIN POSIX authorization to use the ADMIN POSIX command. For information about the VMDIRECT POSIX file and the format of the POSIXGROUP statement, see VMDIRECT POSIX File (see page 245).

Example:

To define a new POSIX group, enter the following:

```
vmdirect admin posix
```

The VMDIRECT POSIX file opens in XEDIT. To add the new POSIX group named FINANCE with a GID of 002, add the following statement:

```
POSIXGROUP FINANCE 002
```

Save and exit the file.

POSIX Information in a Directory Entry or Profile

CA VM:Director supports the following POSIX–related directory control statements, introduced as part of VM/ESA Release 2.1.0. You can use these statements in directory entries and in directory profiles:

Directory Statement	Description
POSIXINFO	Specifies a user's POSIX information. Specifically, userid (UID), group ID (GID/GNAME), initial working directory (IWDIR), initial user program (IUPGM), and file system root (FSROOT).
POSIXGLIST	Lists the names of the POSIX groups of which the user is a member. Groups can be specified by either GID or GNAME.
POSIXOPT	Specifies a user's POSIX options.

Querying and Changing POSIX Information

To query or change the POSIX information in a directory entry, use the CA VM:Director EDIT, EDX, REPENTRY, or GETENTRY commands.

Note: For syntax and usage information for the POSIXINFO, POSIXGLIST, and POSIXOPT directory control statements, see IBM's *CP Planning and Administration* guide for your release of VM.

VMDIRECT POSIX File

The VMDIRECT POSIX file, which resides on the CA VM:Director DRCT minidisk, contains the POSIX group definitions. Use this file, instead of the USER DIRECT file, to define POSIX groups.

Each record in the VMDIRECT POSIX file identifies a POSIX group.

Note: For information about the format of the POSIXGROUP record, see IBM's *CP Planning and Administration* guide.

The VMDIRECT POSIX file can also contain comments, blank lines, and the *ED= special comment.

You can include an edit special comment (*ED=) in the VMDIRECT POSIX file. CA VM:Director maintains this comment with the date and time of last update of the VMDIRECT POSIX file, the userid that last updated the file, the process used to update the file, and the date the file was first updated.

CA VM:Director reads the VMDIRECT POSIX file as part of its initialization process. If it encounters an invalid POSIXGROUP statement, it sends a diagnostic message to the CA VM:Director console and to the CA VM:Director system operator. This operator is a userid you specify on the SYSOPER record in the PRODUCT CONFIG file.

CA VM:Director validates all GIDs and GNAMEs in the source directory against those defined in the VMDIRECT POSIX file. If CA VM:Director encounters a GID or GNAME that is not defined, initialization is terminated with the appropriate error messages.

CA VM:Director ignores the VMDIRECT POSIX file when running on z/VM systems that do not support it.

You can edit the VMDIRECT POSIX file while CA VM:Director is running by using the ADMIN POSIX command. To use the POSIX parameter, you must have at least ADMIN POSIX authorization through a GRANT record in the AUTHORIZ CONFIG file. You can also use the ADMIN, ADMIN *, or ADMIN *ALL authorizations to provide the necessary level of security.

The following figure is a sample VMDIRECT POSIX file that shows the GNAME in the second column and the GID in the third column, as follows:

```
*ED=
* POSIX GROUPS FOR THE XYZ DIVISION

POSIXGROUP      Admin         101
POSIXGROUP      FINANCE       102
POSIXGROUP      MARKETING     103
POSIXGROUP      Sales         104
POSIXGROUP      TechPubs      105

* POSIX GROUPS FOR THE ACM DIVISION

POSIXGROUP      AdminA        201
POSIXGROUP      AdminB        202
POSIXGROUP      ENGI          203
POSIXGROUP      TCOM          204
```

Chapter 21: Systems Management API Support

CA VM:Director provides support for the IBM Systems Management Application Programming Interface (SMAPI) component of z/VM. With this support, a set of interface routines allow you to use CA VM:Director to perform directory management functions called as part of the Systems Management API.

This support replaces calls to IBM-supplied routines with calls to CA VM:Director commands. After you set up a Systems Management API server environment, you will use the instructions in this chapter to add CA VM:Director specific components that receive control when the SMAPI server environment receives a request from a SMAPI client.

> For more information about the server and the Systems Management APIs, see the IBM *Systems Management Application Programming* manual.

This section contains the following topics:

Configure the CA VM:Director Server Environment (see page 247)
Install CA VM:Director API Components in the SMAPI Server (see page 249)
Files Provided by CA VM:Director for the SMAPI Server (see page 250)
Understanding a Directory Manager Specific Return Code (596) from the Socket SMAPI Server (see page 250)
CA VM:Director Command Authorization (see page 251)

Configure the CA VM:Director Server Environment

Several CA VM:Director facilities must be available in order to use CA VM:Director with the SMAPI server environment. The required settings are listed as follows:

To Configure the server environment

1. Configure CA VM:Director to use APPC communications between a client issuing CA VM:Director commands, and the CA VM:Director server virtual machine. Change the configuration as follows:

   a. Add a RESID record in PRODUCT CONFIG file. For example:

      ```
      RESID resourcename
      ```

resourcename

A unique name assigned to an APPC resource identified on an IUCV statement.

Note: For more information about the RESID record, see the *Reference Guide*.

b. Add an IUCV *IDENT *resourcename* LOCAL statement to the VMDIRECT directory entry. For example:

```
IUCV *IDENT resourcename LOCAL
```

resourcename

The unique name of an APPC resource.

2. Use the CONFIG DASD command to add a unique *extentname* field to each EXTENT statement in the file.

   Note: For more information, see EXTENT Record in the chapter "Configuration File Reference" in the *Reference Guide*.

3. Configure CA VM:Director to use the Servant Facility if your site is not already using it.

   Note: For information about setting up the Servant Facility, see Servant Facility (see page 84).

Install CA VM:Director API Components in the SMAPI Server

These instructions refer to the *IBM Systems Management Application Programming* manual, and they apply only to the Socket version of the SMAPI server. See the separate section, later in this chapter, for information on the RPC Socket server, if you use that version of SMAPI support. To install the CA VM:Director API components in the Socket SMAPI server, you need to perform the following steps:

To install the CA VM:Director API components in the VSMWORKn virtual machines

1. Install the VSMREQIN and VSMWORKn user IDs as described in the chapter “Setting up and Configuring the Server Environment” in the *IBM Systems Management Application Programming manual*. Install the VSMREQIU server if you plan to use IUCV to perform SMAPI requests.
2. SMAPI provides various types of worker virtual machines. The SMAPI workers which must be modified for use with CA VM:Director are the VSMGUARD ID and the "long call" workers VSMWORK2, VSMWORK3, etc.
3. All CA VM:Director components needed by the SMAPI workers are distributed on the CA VM:Director public files disk, usually the VMANAGER 193 disk. Some CA VM:Director customers copy the files on the 193 to a public disk such as the MAINT 19E Y-Disk. If your installation does not do this, then:

   a. Add this statement to the directory entry of each SMAPI worker ID:

      ```
      LINK VMANAGER 193 293 RR
      ```

   b. Add this statement to the PROFILE EXEC for each SMAPI worker ID:

      ```
      ACCESS 293 H
      ```

   Note: If you have an alternate method of accessing files on the CA VM:Director PUBLIC disk, implement that method for each SMAPI worker ID.
4. Allow each SMAPI worker ID to issue CP Diagnose code X’D4’. With no External Security Manager, privilege class "B" is needed in each directory entry. If you are using an External Security Manager then use the appropriate controls in the ESM to allow each SMAPI worker to issue the diagnose.
5. Update the SMAPI Server Configuration File as documented in the section “The Server Configuration File” in the chapter “Setting up and Configuring the Server Environment” in the *IBM Systems Management Application Programming* manual.

   You are changing the value of the configuration file attribute named “DM_Exit” from its default value of “DMSSIXDM” to the value “VMXSIXDM” so that the VMXSIXDM routine supplied by CA VM:Director will be used as the “Directory Manager Exit Routine”.

Files Provided by CA VM:Director for the SMAPI Server

The following files, installed using the instructions in the previous section, provide the support to use CA VM:Director routines with the Systems Management API.

VMXSIXDM EXEC

The VMXSIXDM EXEC is a large SELECT construct, with a WHEN clause for each API routine. It retrieves the original CSL parameters to a stem ‘p’, where p.1 is the first argument, p.2 is the second, and so on. The arguments to each API routine are validated and then used to construct CA VM:Director commands to carry out the requested system management function.

VMXHVCD4 MODULE

This MODULE is called by VMXSIXDM to issue a CP Diagnose Code X’D4’.

VMXSML2P REXX

Used by VMXSMAPI EXEC to convert logical file line images to REXX parameters.

VMXSMP2L REXX

Used by VMXSMAPI EXEC to convert REXX parameters to logical file line images.

Understanding a Directory Manager Specific Return Code (596) from the Socket SMAPI Server

When CA VM:Director commands issued by the Socket SMAPI Server do not have a corresponding return code defined by the API, then the SMAPI server responds to the caller with a special 596 Return Code value. When a 596 Return Code is presented, a special Reason Code will be provided to indicate the nature of the problem. The Reason Code consists of the return code from the CA VM:Director command called by the API added to a unique number for each command. For instance, reason code 900015 would be issued if the API issued an ADDMDISK command (encoded as 900000) which responded with a return code value of 15.

The following table defines the encoded value associated with a specific CA VM:Director command:

Command Name	Associated encoded value
NOTIFY	200000
TAG	300000
SCAN	400000
WORKUNIT	500000

Command Name	Associated encoded value
ADDENTRY	600000
DELENTRY	700000
DUPMDISK	800000
ADDMDISK	900000
DELMDISK	1000000
LOCK	1100000
UNLOCK	1200000
PASSWORD	1300000
GETENTRY	1400000
REPENTRY	1500000
VMXDSD00	1600000
MAP	1700000
ADMIN	1800000
VERSION	1900000
QUERY	2000000

CA VM:Director Command Authorization

Authorization to execute CA VM:Director commands is needed for two types of user IDs in order to utilize the SMAPI services:

1. The user ID passed as the authorized user on a Socket Server API call
2. The user ID for the SMAPI worker IDs VSMGUARD, VSMWORK2, VSMWORK3, etc

These user IDs must be authorized as follows:

Specify the user ID in the VSMWORK1 AUTHLIST file for the SMAPI Socket Server.

- Specify the user ID on a “GRANT *ALL TO userid” record in the AUTHORIZ

  CONFIG file, using the CONFIG command.
- Use the ADMIN MANAGERS command to configure the user ID as a valid CA VM:Director Directory Manager.

Chapter 22: ESM Support (CA VM:Director only)

This chapter describes how to enable access control and auditing for CA VM:Director commands using a VM external security manager (ESM) product as an alternative to the GRANT/WITHHOLD command authorization support.

Any ESM that supports a RACROUTE macro interface can be used. This includes products like RACF from IBM or CA Top Secret for z/VM from CA.

This section contains the following topics:

ESM Command Authorization (see page 253)
Setting Up User Exits for Co-Existence of CA VM:Director with an ESM (see page 266)
Enabling CA VM:Director Access to ESM-Controlled Resources (see page 267)
Enabling End-User Access to ESM-Controlled CA VM:Director Commands (see page 269)

ESM Command Authorization

The new ESM facility allows the customer to control user access to product commands through an ESM product. The ESM facility provides an alternative to the existing GRANT/WITHHOLD command authorization support. Administrators now have a choice of using the existing command authorization facility or the new ESM facility. This ESM facility gives system administrators the ability to control and audit the use of CA VM:Director product commands.

Typically, access is allowed to three types of end users:

- Everyone
- Directory managers
- System administrators

To use this feature, a RACROUTE-compliant ESM product, such as CA Top Secret for z/VM, must be installed. ESM rules must be defined for each CA VM:Director command, specifying which users are authorized to issue the command.

The ESM will do one of the following:

- ALLOW a command to be used by an authorized user
- DENY the ability to issue a particular command

- DEFER the authorization decision back to the existing GRANT/WITHHOLD logic

ESM command authorization enables CA VM:Director administrators to provide specifics about users and the commands you want to authorize for their use. The specifications are then run against an ESM database containing the rules under which users are granted or denied the ability to use the specified commands.

The CA VM:Director product issues the standard mainframe RACROUTE macro to match submitted resource names to the rules in the ESM database. Resource names are generally grouped in *classes*. The one- to eight-character class name is specified on each RACROUTE macro. The *class name* is defined using the ESM configuration statement. For more information, see ESM Configuration Statement (see page 265) in this chapter. The *resource names* for command authorization testing are strings of one- to eight-character tokens separated by periods. For an example of how to use resource names, see Creating Command Authorization Rules (see page 254).

Creating Command Authorization Rules

The following example authorizes a user to issue the ADDMDISK command to add a disk for a specific target user, user ID *xxxxx01*, who is managed by directory manager VMANAGER.

`CADIRECT.ADDMDISK.`*`xxxxxx01`*`.VMANAGER`

This format allows a variety of access rules to be written, using the resource name wildcard matching capabilities of the ESM.

The following table shows examples where an "*" is a wildcard character meaning *any* token:

ESM Rule	Rule Description
CADIRECT.ADDMDISK*.*	Allows an ADDMDISK command with any target user ID to be issued.
CADIRECT.ADDMDISK.*xxxxxx01*.*	Allows an ADDMDISK command for only the *xxxxxx01* user ID.
CADIRECT.ADDMDISK.*.VMANAGER	Allows an ADDMDISK command for any user ID managed by the VMANAGER user, who is a directory manager.
CADIRECT.*.*.*	Allows a user to use any CA VM:Director command.

Note: Trailing wildcard characters can be omitted.

Although the resource names formats can define specialized command authorizations for system administrators, additional formats are needed for two unique groups of requestors:

- General users
- Directory manager users

Granting Authorization to General Users

For general users, authorization is typically granted to those commands available for manipulation of their own user IDs (indicated by a shorthand notation). Rather than using the tokens *userid.manager*, the tokens *SYSTEM.SELF* are substituted.

The following table shows rules (or a subset of them determined by installation policy) that are granted to general users for self-administration of user IDs:

ESM Rule	Rule Description
CADIRECT.MAINT.LINK.SYSTEM.SELF	Use of MAINT subfunction LINK.
CADIRECT.MAINT.MDPW.SYSTEM.SELF	Use of MAINT subfunction MDPW.
CADIRECT.MAINT.MGRID.SYSTEM.SELF	Use of MAINT subfunction MGRID.
CADIRECT.MAINT.PASSWORD.SYSTEM.SELF	Use of MAINT subfunction PASSWORD.
CADIRECT.MAINT.REVIEW.SYSTEM.SELF	Use of MAINT subfunction REVIEW.
CADIRECT.MAINT.RLINK.SYSTEM.SELF	Use of MAINT subfunction RLINK.
CADIRECT.MAINT.STORAGE.SYSTEM.SELF	Use of MAINT subfunction STORAGE.
CADIRECT.QUERY.ACCOUNT.SYSTEM.SELF	Use of QUERY ACCOUNT for self.
CADIRECT.QUERY.LOGMSG.SYSTEM	Use of QUERY LOGMSG SYSTEM.
CADIRECT.TRANSFER.SYSTEM.SELF	TRANSFER own Mdisk to another.
CADIRECT.USESEL01.SYSTEM.SELF	Use of USER menu option 1—Logon password and storage size.
CADIRECT.USESEL02.SYSTEM.SELF	Use of USER menu option 2—Mdisk Link mode and password.
CADIRECT.USESEL06.SYSTEM.SELF	Use of USER menu option 6—Screen colors and highlighting.

ESM Rule	Rule Description
CADIRECT.USESEL07.SYSTEM.SELF	Use of USER menu option 7—Link another's Mdisk.
CADIRECT.USESEL08.SYSTEM.SELF	Use of USER menu option 8—Review or remove other's LINKs.
CADIRECT.USESEL10.SYSTEM.SELF	Use of USER menu option 10—Delete a link to other user's MDISK.
CADIRECT.USESEL11.SYSTEM.SELF	Use of USER menu option 11—Review the Directory Entry.

Granting Authorization to Directory Managers

For directory managers, authorization is usually granted to directory manager commands. A shorthand notation is also used here. Instead of the using the tokens *userid.manager*, the tokens **.manager* is substituted to authorize use of the command on any user ID whose directory manager is the specified manager.

The following rules are granted to directory managers, or a subset of them, determined by installation policy so that they can operate on the user IDs they manage (but not those managed by other managers):

ESM Rule	Resource
CADIRECT.ASSIGN.*.manager	Change of manager for a user ID.
CADIRECT.CHANGE.*.manager	Change of user ID name.
CADIRECT.DISPLINK.*.manager	Display of user's disk LINKs.
CADIRECT.EXPIRE.*.manager	Expiration of user's password.
CADIRECT.MANAGE.*.manager	Use of MANAGE menu for user.
CADIRECT.MAINTMAN.*.manager	Use of MAINT MANAGE command.
CADIRECT.MDSKSCAN.*.manager	Report on MDISKs of user.
CADIRECT.PASSWORD.*.manager	Change password of user.
CADIRECT.QUERY	Any QUERY command operand.
CADIRECT.RESET.USERPASS.*.manager	Clear user's logon password violation count.
CADIRECT.RESET.VMXLINK.*.manager	Clear user's LINK password violation count.

ESM Rule	Resource
CADIRECT.TRANSFER.*.manager	Give user's Mdisk to another user.
CADIRECT.ULIST.*.manager	List user IDs managed by the calling manager.
CADIRECT.USER.*.manager	Use of USER menu for the user ID being managed.

Resource definitions that enable the creation of new user IDs require a specialized format. For these, a token sequence of *SYSTEM.NEWUSER* is substituted in resource definitions that allow user ID creation.

The following tables displays corresponding RACROUTE resource names and the user ID creation authority each has:

RACROUTE Resource Name	Authorization
CADIRECT.ADDENTRY.SYSTEM.NEWUSER	Creation of a new user ID.
CADIRECT.MANSEL01.SYSTEM.NEWUSER	Creation of a new user ID with MANAGE.
CADIRECT.MULTIPLE.NEWUSER	Creation of a new user with MULTIPLE.

RACROUTE Resource Names

The following table contains a listing of CA VM:Director commands and utilities and the corresponding RACROUTE resource name, which is passed to the ESM product:

CA VM:Director Command or Utility	Type of Authorization	RACROUTE Resource Name
ABEND	Terminate CA VM:Director operation abnormally	CADIRECT.ABEND
ADDENTRY	Create a user ID or profile from a skeleton file	CADIRECT.ADDENTRY.SYSTEM.NEWUSER.SKELETON
	Create a user ID or profile from an input file	CADIRECT.ADDENTRY.SYSTEM.NEWUSER.NOSKEL
ADDENTRY (w/MDISK)	If the input file creates a minidisk	CADIRECT.ADDMDISK.*userid.manager*
ADDMDISK	Add a minidisk for a user ID	CADIRECT.ADDMDISK.*userid.manager*

CA VM:Director Command or Utility	Type of Authorization	RACROUTE Resource Name
ADMIN	Use all functions of the ADMIN command	CADIRECT.ADMIN
	Edit the VMDIRECT MANAGERS file	CADIRECT.ADMIN.MANAGERS
	Edit the VMDIRECT GLOBALS file	CADIRECT.ADMIN.GLOBALS
	Edit a subpool entry	CADIRECT.ADMIN.POOL.*poolname*
	Edit the VMDIRECT POSIX file	CADIRECT.ADMIN.POSIX
	Edit a directory profile	CADIRECT.ADMIN.PROFILE.*profname*
	Edit a skeleton file	CADIRECT.ADMIN.SKELETON
	Define or change SFS managers' enrollment defaults or enrollment limits	CADIRECT.ADMIN.SFSMGRS
ASSIGN	Assign a user ID to a different manager	CADIRECT.ASSIGN.*userid.manager.newmanager*
AUDITEXT	Extract current audit information	CADIRECT.AUDITEXT
CHANGE	Change a user ID	CADIRECT.CHANGE.*userid.manager*
CHGMDISK	Move or change a minidisk	CADIRECT.CHGMDISK.*userid.manager*
CHGVOLNM	Change the label and all references to the volser of any DASD volume controlled by CA VM:Director	CADIRECT.CHGVOLNM.*oldvolser*
CLASS	Assign a CP privilege class	CADIRECT.CLASS.*classletter.userid.manager*
CMD	Route command to another node	CADIRECT.CMD
CMS	Execute a CMS or CP command on the CA VM:Director service virtual machine	CADIRECT.CMS[.*word*[.*word2*[.*word3*[.*word4*]]]]
COMPRESS	Defragment disk storage	CADIRECT.COMPRESS.*volser*
CONFIG	Edit the CA VM:Director configuration files	CADIRECT.CONFIG
	Edit the AUTHORIZ CONFIG file	CADIRECT.CONFIG.AUTHORIZ
	Edit the DASD CONFIG file	CADIRECT.CONFIG.DASD
	Edit the PRODUCT CONFIG file	CADIRECT.CONFIG.PRODUCT
	Edit the SECURITY CONFIG file	CADIRECT.CONFIG.SECURITY

CA VM:Director Command or Utility	Type of Authorization	RACROUTE Resource Name
	Edit the CA VM:Director SFS configuration	CADIRECT.CONFIG.SFS
CPFMTXA	Change allocation on object directory volume	CADIRECT.CPFMTXA
DELENTRY	Delete an existing user ID or profile	CADIRECT.DELENTRY.*userid.manager*
DELETE	Delete file space for an active user ID	CADIRECT.DELETE.*userid.manager*
DELMDISK	Delete a user ID's minidisk	CADIRECT.DELMDISK.*userid.manager*
DISPLINK	Display links to a user's minidisks	CADIRECT.DISPLINK.*userid.manager*
DUPENTRY	Create a new user ID based off of an existing ID	CADIRECT.DUPENTRY.*existingID.newID*
DUPMDISK	Create an exact copy of an existing minidisk	CADIRECT.DUPMDISK.*userid*
	For the entry that holds the target minidisk	CADIRECT.ADMDISK.*userid*
EDIT	Edit a user ID's directory entry	CADIRECT.EDIT.*userid.manager*
EDX	Edit a user ID's directory entry, expanding any INCLUDE statement	CADIRECT.EDIT.*userid.manager*
END	Terminate CA VM:Director immediately only	CADIRECT.END.FORCE
	Terminate CA VM:Director operation only after current processes complete	CADIRECT.END.NOFORCE
ENROLL	Enroll a user ID in an SFS file pool	CADIRECT.ENROLL.*userid.manager*
ENTRY	Update or query a directory entry	CADIRECT ENTRY entry subcommand
EXPIRE	Expire a user ID's logon password	CADIRECT.EXPIRE.*userid.manager*
EXTRACT	Extract directory information	CADIRECT.EXTRACT
GENINCL	Add, change or delete an INCLUDE statement to a user ID's directory entry	CADIRECT.GENINCL.*userid.manager.oldprofile* CADIRECT.GENINCL.*userid.manager.newprofile*
GETENTRY	Retrieve current copy of a user ID's directory entry or a directory profile	CADIRECT.GETENTRY.*userid.manager*
		CADIRECT.REPENTRY.*userid.manager*
		CADIRECT.GETENTRY.*profilename*

CA VM:Director Command or Utility	Type of Authorization	RACROUTE Resource Name
GETPWEXP	Display user ID password expiration information	CADIRECT.GETPWEXP.*userid*
GRANT	Allow a user to grant access to a file space for other users	CADIRECT.GRANT.AUTHORITY.*filepool.filespace.user id.manager*
JOURNAL	Display password violations and reset password violation count	CADIRECT.JOURNAL
	Display password violations	CADIRECT.JOURNAL.LIST.*word1.word2.word3*…
	Reset a password violation count to zero	CADIRECT.JOURNAL.RESET.*word1.word2.word3*…
LISTAUTH	Query the authorizations specified in the AUTHORIZ CONFIG file	CADIRECT.LISTAUTH.*authwords*
LOCK	Prevent updates to any object	CADIRECT.LOCK
	Prevent updates to a CMS file	CADIRECT.LOCK.FILE.*fname.ftype.fmode*
	Prevent updates to a user ID	CADIRECT.LOCK.USER.*userid.manager*
	Prevent updates to a profile	CADIRECT.LOCK.PROFILE.*profname*
MACLOAD	Load a macro to the CA VM:Director service virtual machine	CADIRECT.MACLOAD
MAINT	Perform line-mode user functions	CADIRECT.MAINT.*subfunction*
MAINT	Perform line-mode management functions	CADIRECT.MAINTMAN.*userid.manager.subfunction*
	Perform line-mode user functions for another user ID	CADIRECT.MAINTMAN.*userid.manager.USER.subfun ction*
MANAGE	Create user IDs (part of selection 1)	CADIRECT.MANSEL01.SYSTEM.NEWUSER
MANAGE	Create user IDs (part of selection 1)	CADIRECT.MANSEL01.*userid.manager*
	Use menu selection 2	CADIRECT.MANSEL02.*userid.manager*
	Use menu selection 3	CADIRECT.MANSEL03.*userid.manager*
	Use menu selection 4	CADIRECT.MANSEL04.*userid.manager*
	Use menu selection 5	CADIRECT.MANSEL05.*userid.manager*
	Use menu selection 6	CADIRECT.MANSEL06.*userid.manager*
	Use menu selection 7	CADIRECT.MANSEL07.*userid.manager*

CA VM:Director Command or Utility	Type of Authorization	RACROUTE Resource Name
	Use menu selection 8	CADIRECT.MANSEL08.*userid.manager*
	Use menu selection 9	CADIRECT.MANSEL09.*userid.manager*
	Use menu selection 10	CADIRECT.MANSEL10.*userid.manager*
MAP	Map a volume	CADIRECT.MAP
		CADIRECT.MAP.*volser.parameters*
MAY	Query authorizations on the AUTHORIZ CONFIG	CADIRECT.MAY.*userid.authwords*
MDSKSCAN	Scan a user ID's minidisks	CADIRECT.MDSKSCAN.*userid.manager*
MODIFY	Modify the SFS allocation for a user ID	CADIRECT.MODIFY.*userid.manager*
MOVE2SFS	Copy data from a minidisk to SFS	CADIRECT.MOVE2SFS.*userid.manager*
		CADIRECT.MANAGE.*userid.manager*
		CADIRECT.SFSADMIN
		CADIRECT.MOVERO
		CADIRECT.DELMDISK.*userid.manager*
MULTIPLE	Create several user IDs at the same time	CADIRECT.MULTIPLE.NEWUSER
	Remove several user IDs at the same time	CADIRECT.MULTIPLE.REMOVE.*userid.manager*
	Place several user IDs on hold at the same time	CADIRECT.MULTIPLE.HOLD.*userid.manager*
	Reactivate several held user IDs at the same time	CADIRECT.MULTIPLE.ACTIVATE.*userid.manager*
NEWIPL	Change an IPL system name or device in all directory entries to a new IPL system name or device	CADIRECT.NEWIPL
NOLOG	Change a user ID's password to NOLOG	CADIRECT.NOLOG.*userid.manager*
OVERRIDE	Alter privilege classes without shutting down CA VM:Director	CADIRECT.CPOVERID
PAINT		CADIRECT.PAINT.*screenname*

CA VM:Director Command or Utility	Type of Authorization	RACROUTE Resource Name
PASSWORD	Set only randomly generated passwords for a user ID	CADIRECT.PASSWORD.*userid.manager*.RANDOM
	Set only a specific password for a user ID	CADIRECT.PASSWORD.*userid.manager*.SPECIFIC
QLOCK	Display all CA VM:Director locks	CADIRECT.QLOCK
QPCB	List active CA VM:Director processes	CADIRECT.QPCB
QSTART	Display the time CA VM:Director was most recently started	CADIRECT.QSTART
QUERY	Use all parameters on the QUERY command	CADIRECT.QUERY
	Display your user ID's account number	CADIRECT.QUERY.ACCOUNT.SYSTEM.SELF
	Display a user ID's account number	CADIRECT.QUERY.ACCOUNT.*userid.manager*
	Display information about a manager's allocation space	CADIRECT.QUERY.ALLOC.*manager*
	Display a user ID's privilege class	CADIRECT.QUERY.CLASS.*userid.manager*
	List the names of the file pools CA VM:Director manages	CADIRECT.QUERY.FILEPOOL
	List user IDs on hold	CADIRECT.QUERY.HOLD.*userid.manager*
	List user IDs that are directory managers	CADIRECT.QUERY.MANAGERS
	List user IDs whose passwords have not changed for a specified number of days	CADIRECT.QUERY.PASSWORD.*manager*
	List user IDs' directory entries that include a directory profile	CADIRECT.QUERY.PRFUSERS.*profile*
	Display status information about the Servant Facility	CADIRECT.QUERY.SERVANT
	List the names of the file pools and user storage groups from which you can allocate file space	CADIRECT.QUERY.SFS
	List the skeleton files a manager can use	CADIRECT.QUERY.SKELETON.*manager*
	List the subpools a manager can use	CADIRECT.QUERY.SUBPOOLS.*manager*

CA VM:Director Command or Utility	Type of Authorization	RACROUTE Resource Name
	List the user IDs a manager manages	CADIRECT.QUERY.USERS.*userid.manager*
	Display the CA VM:Director release level	CADIRECT.QUERY.VERSION
	Determine the status of long running commands	CADIRECT.QUERY.WORKUNIT.*userid.manager*
REBUILD	Condenses and defragments the CP object directory	CADIRECT.REBUILD
RECLAIM	Reclaim DASD space from MOVERO minidisk	CADIRECT.RECLAIM
REPENTRY	Replace a directory entry or directory profile	CADIRECT.REPENTRY.*userid.manager*
REPENTRY (with MDISK functions)	Replace a directory entry or directory profile if the new entry adds a minidisk	CADIRECT.ADDMDISK.*userid.manager*
	Replace a directory entry or directory profile-if the new entry changes a minidisk	CADIRECT.CHGMDISK.*userid.manager*
	Replace a directory entry or directory profile-if the new entry deletes a minidisk	CADIRECT.DELMDISK.*userid.manager*
RESET	Reset password violation counts for a user that occurred while verifying the password	CADIRECT.RESET.USERPASS.*userid.manager*
	Reset password violation counts that occurred while trying to create a directory link	CADIRECT.RESET.VMXLINK.*userid.manager*
REVOKE	Allow a user to revoke access to a file space for other users	CADIRECT.REVOKE.AUTHORITY.*filepool.filespace.userid.manager*
TAKEOVER	Convert Agent to Master in SSI	CADIRECT.TAKEOVER
TRACE	Trace execution of a CA VM:Director macro	CADIRECT.TRACE.*word1.word2.word3…*
TRANSFER	Transfer a minidisk from one user ID to another	CADIRECT.TRANSFER.*userid.manager*
ULIST	Display information about user IDs	CADIRECT.ULIST.*userid.manager*

CA VM:Director Command or Utility	Type of Authorization	RACROUTE Resource Name
UNLOCK	Remove a CMS file, profile, or user ID lock	CADIRECT.UNLOCK
	Remove a lock from a CMS file	CADIRECT.UNLOCK.FILE
	Remove a lock from a profile	CADIRECT.UNLOCK.PROFILE
	Remove a lock from a user ID	CADIRECT.UNLOCK.USER.*userid.manager*
USER	Use menu selection 1	CADIRECT.USESEL01.*userid.manager*
	Use menu selection 2	CADIRECT.USESEL02.*userid.manager*
	Use menu selection 3	CADIRECT.USESEL03.*userid.manager*
	Use menu selection 4	CADIRECT.USESEL04.*userid.manager*
	Use menu selection 5	CADIRECT.USESEL05.*userid.manager*
	Use menu selection 6	CADIRECT.USESEL06.*userid.manager*
	Use menu selection 7	CADIRECT.USESEL07.*userid.manager*
	Use menu selection 8	CADIRECT.USESEL08.*userid.manager*
	Use menu selection 9	CADIRECT.USESEL09.*userid.manager*
	Use menu selection 10	CADIRECT.USESEL10.*userid.manager*
	Use menu selection 11	CADIRECT.USESEL11.*userid.manager*
VMXBKP01	Create a USER DIRECT file representing a copy of the CA VM:Director directory database	CADIRECT.BACKUP
VMXBKP02	Create a backup copy of the CA VM:Director directory database using DDR	CADIRECT.BACKUP
VMXBKP03	Create a backup copy of the CA VM:Director directory database using COPYFILE	CADIRECT.BACKUP
VMXMACUP	Install a new or revised CA VM:Director macro, EXEC, or MESSAGES file	CADIRECT.MACLOAD

Authorizations for Command Processing

Authorizations for command processing are those authorizations that do not relate to any one command. They refer instead to the way processing takes place for different commands.

The following table shows authorities and their corresponding RACROUTE resource names:

Authority	RACROUTE Resource Name
User can allocate minidisks in any existing subpool.	CADIRECT.ANYPOOL
User can add, delete, or change a minidisk on a cached device.	CADIRECT.CACHED
User can move a minidisk with read-only links.	CADIRECT.MOVERO
User can move a minidisk without copying the data.	CADIRECT.NOCOPY
User can add, delete, and move a minidisk without formatting.	CADIRECT.NOFORMAT CADIRECT.NOFORMAT.ADD CADIRECT.NOFORMAT.DELETE CADIRECT.NOFORMAT.MOVE
Skip the check for a password when *command* is issued.	CADIRECT.NOPASS.*command*
User can override rejection by or choice of the NEWUSER user exit	CADIRECT.OVERRIDE.NEWUSER
User can configure for SFS.	CADIRECT.SFSADMIN

ESM Configuration Statement

External security manager support is enabled using a configuration file statement that requests *rule-based* command authorization from an ESM product. This statement requests that an ESM product be used to validate that a requesting user ID meets the specified criteria to issue the command and operands defined by the *resource name*.

The syntax for the ESM configuration statement is:

ESM *resourceclass resourceprefix*

resourceclass

A one- to eight-character name that groups similar resources.

A suggested resource class name for CA products is CAVMCMD.

resourceprefix

A one- to eight-character name used as the first-level qualifier in the resource name. A RACROUTE resource name is a character string formed by tokens separated by a period delimiter. The first token is the resource prefix.

A suggested resource prefix name for the CA VM:Director application is CADIRECT.

Security Impact

If the rules defined for an ESM are equivalent to the rules defined for the native application command authorization facility, then the two configurations have the same level of security.

Setting Up User Exits for Co-Existence of CA VM:Director with an ESM

Note: For detailed information about these exits and procedures, see the chapter "User Exit Reference" in the *Reference Guide*.

The "User Exit Reference" chapter describes how to customize and configure any product user exit. However, the specific user exit recommendations for using CA VM:Director with an ESM product are described below.

Suppressing CP Commands Using the PASSWORD User Exit

CA VM:Director uses several menus and commands that modify the passwords in the CP directory. These commands and menus have no influence on the passwords maintained by an ESM.

Examples of these commands are as follows:

- EXPIRE
- NOLOG

- PASSWORD
- USER
- MANAGE

For any installation that wants to suppress these commands, a simple implementation of the PASSWORD user exit can be used to suppress them. Use the following statements as a REXX implementation of the PASSWORD user exit:

```
/* PASSWORD User Exit */
push 'Passwords are controlled by an external security manager.'
Exit 8
```

Synchronizing ESM Passwords with CP Directory Passwords Using the PASSCHNG User Exit

The PASSCHNG user exit provides notification of a change to a user ID's logon password. This exit enables installations to synchronize ESM controlled passwords with those maintained in the CP Directory by CA VM:Director.

Note: For more information about the PASSCHNG user exit, see the chapter "User Exit Reference" in the *Reference Guide*.

Verifying the Identity of the Command Originator Using the CHKPASS User Exit

CA VM:Director protects against commands issued from unattended terminals by prompting for and checking passwords during command execution. With an ESM in place, passwords are controlled by the security manager rather than by CA VM:Director. To match passwords against those controlled by the security manager, use the CHKPASS user exit. The provided sample exit program (VMXEXIT9 XASSEMBL) can be compiled and installed without any changes to provide this capability.

Enabling CA VM:Director Access to ESM-Controlled Resources

Before the CA VM:Director server can operate with an ESM, the ESM must be configured to enable the server to have access to certain system resources. The way in which access is granted depends on the specific ESM product being used.

The following lists the CA Top Secret for z/VM commands needed to give CA VM:Director access to the resources it needs to operate.

The ESM provides access to:

- The virtual machines for CA VM:Director and its servant facility are:

```
TSS CREATE(VMDIRECT) NAME('VMDIRECT') DEPT(VMSYSTM)
PASS(password) TYPE(USER)
TSS CREATE(VMX$0001) NAME('VMX$0001') DEPT(VMSYSTM)
PASS(password) TYPE(USER)
TSS CREATE(VMX$0002) NAME('VMX$0002') DEPT(VMSYSTM)
PASS(password) TYPE(USER)
```

- Each virtual machine's owned minidisks by that machine:

```
TSS ADDTO(VMDIRECT)  VMMDISK(VMDIRECT.)
TSS ADDTO(VMX$0001)  VMMDISK(VMX$0001.)
TSS ADDTO(VMX$0001)  VMMDISK(VMX$0001.)
```

- A minidisk that maps the CP directory space from the server:

```
TSS PERMIT(VMDIRECT) VMMDISK($DASD$.004) ACCESS(MREAD)
```

- The server 194 common disk by each servant virtual machine:

```
TSS PERMIT(VMX$0001) VMMDISK(VMDIRECT.0194) ACCESS(READ)
TSS PERMIT(VMX$0002) VMMDISK(VMDIRECT.0194) ACCESS(READ)
```

- Bring up the product server by system control virtual machines:

```
TSS ADDTO(xxxxxxxx)  VMMACH(VMDIRECT)
TSS PERMIT(OPERATOR) VMMACH(VMDIRECT) ACCESS(AUTOLOG)
TSS PERMIT(AUTOLOG1) VMMACH(VMDIRECT) ACCESS(AUTOLOG)
```

- Bring up the servant machines from the product server:

```
TSS ADDTO(xxxxxxxx)  VMMACH(VMX$0001)
TSS ADDTO(xxxxxxxx)  VMMACH(VMX$0002)
TSS PERMIT(VMDIRECT) CPCMD(AUTOLOG)
TSS PERMIT(VMDIRECT  VMMACH(VMX$0001) ACCESS(AUTOLOG)
TSS PERMIT(VMDIRECT  VMMACH(VMX$0002) ACCESS(AUTOLOG)
```

- Issue Diagnose X'A0' from the product server:

```
TSS PERMIT(VMDIRECT) DIAG(A0) ACTION(VMPRIV)
```

Enabling End-User Access to ESM-Controlled CA VM:Director Commands

The following sample file is available on the VMDIRECT 176 disk. It shows the CA Top Secret for z/VM commands needed to authorize access to CA VM:Director commands for different kinds of users. You should tailor the sample according to your installation's requirements.

Note: For more information about these commands, see the CA Top Secret for z/VM documentation.

```
/* TSSVME sample file */
/* Use RESCODE in the range 01-3F to define an RIE resource.  */
/* It is a general resource, rather than a prefixed resource. */

/* A default ACLST is implied with two levels - ALL and NONE. */
/* The default access level (DEFACT) value is ALL.            */

"TSS ADDTO(RDT) RESCLASS(CAVMCMD) RESCODE(3F)" ,
"ATTR(MASK,DEFPROT)" ,
"MAXLEN(80)"

/* Define the cadirect. resource name */
"TSS ADDTO(CAIMAINT) CAVMCMD(CADIRECT.)"

/* Define the facility connection resource */
"TSS ADDTO(CAIMAINT) IBMFAC(ICHCONN)"

/* Allow VMDIRECT to issue RACROUTE requests */
/* Access(Read) allows REQUEST=AUTH          */
/* Access(Update) allows REQUEST=VERIFY      */
"TSS PERMIT(VMDIRECT) IBMFAC(ICHCONN) ACCESS(UPDATE)"

/* System Admin Authorizations */

"TSS PERMIT(MAINT) CAVMCMD(CADIRECT.)"
"TSS PERMIT(VMANAGER) CAVMCMD(CADIRECT.)"

/* Directory Manager Authorizations                        */

/*  This set of authorizations is issued for each user who */
/*  is a CA VM:Director Directory Manager. The sample shows a */
/*  set of authorizations for VM user IDs:                  */
/*                DIRMGR1 and DIRMGR2.                      */
/*  Substitute the user IDs of your own Directory Manager   */
/*  users.                                                  */

/*   Authorizations for DIRMGR1 who is a Directory Manager */

"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.ASSIGN.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.CACHED)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.CHANGE.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.DELENTRY.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.DISPLINK.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.EXPIRE.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANAGE.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MAINTMAN.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MDSKSCAN.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.PASSWORD.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.QUERY)"
```

```
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.RESET.USERPASS.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.RESET.VMXLINK.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.TRANSFER.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.USER.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.ADDENTRY.SYSTEM.NEWUSER.*)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL01.SYSTEM.NEWUSER)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MULTIPLE.NEWUSER)"

"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL02.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL03.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL04.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL05.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL06.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL07.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL08.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL09.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.MANSEL10.*.DIRMGR1)"

"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.USESEL01.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.USESEL02.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.USESEL06.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.USESEL07.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.USESEL08.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.USESEL10.*.DIRMGR1)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.USESEL11.*.DIRMGR1)"

/*   Authorizations for DIRMGR2 who is a Directory Manager */

"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.ASSIGN.*.DIRMGR2)"
"TSS PERMIT(DIRMGR1) CAVMCMD(CADIRECT.CACHED)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.CHANGE.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.DELENTRY.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.DISPLINK.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.EXPIRE.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANAGE.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MAINTMAN.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MDSKSCAN.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.PASSWORD.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.QUERY)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.RESET.USERPASS.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.RESET.VMXLINK.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.TRANSFER.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.USER.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.ADDENTRY.SYSTEM.NEWUSER.*)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL01.SYSTEM.NEWUSER)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MULTIPLE.NEWUSER)"

"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL02.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL03.*.DIRMGR2)"
```

```
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL04.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL05.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL06.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL07.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL08.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL09.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.MANSEL10.*.DIRMGR2)"

"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.USESEL01.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.USESEL02.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.USESEL06.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.USESEL07.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.USESEL08.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.USESEL10.*.DIRMGR2)"
"TSS PERMIT(DIRMGR2) CAVMCMD(CADIRECT.USESEL11.*.DIRMGR2)"

/* General User Authorizations */

"TSS PERMIT(ALL) CAVMCMD(CADIRECT.MAINT.LINK)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.MAINT.MDPW)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.MAINT.MGRID)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.MAINT.PASSWORD)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.MAINT.REVIEW)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.MAINT.RLINK)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.MAINT.STORAGE)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.MAINT.TRANSFER)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.QUERY.ACCOUNT.SYSTEM.SELF)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.QUERY.LOGMSG.SYSTEM)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.USESEL01.SYSTEM.SELF)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.USESEL02.SYSTEM.SELF)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.USESEL06.SYSTEM.SELF)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.USESEL07.SYSTEM.SELF)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.USESEL08.SYSTEM.SELF)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.USESEL10.SYSTEM.SELF)"
"TSS PERMIT(ALL) CAVMCMD(CADIRECT.USESEL11.SYSTEM.SELF)"
```

Chapter 23: Upgrade in Place Installation

This chapter documents the steps to follow when using the process documented in the section on Upgrade Installation in the *z/VM Installation Guide*. This process is new in z.VM 6.3.0.

Upgrading z/VM 6.2.0 to z/VM 6.3.0

Follow the steps in this section to upgrade z/VM 6.2.0 to z/VM 6.3.0:

1. Add an ACCOUNT User Exit to CA VM:Director that lets MIGMAINT add any account code.

   The exit EXEC file can be as simple as the following example:

   ```
   /* MYACTXIT EXEC * /
   Exit 0
   ```

   Place the file on the VMDIRECT A-disk, and add this line to your PRODUCT CONFIG file (substituting your exit filename):

   ```
   USEREXIT ACCOUNT MYACTXIT EXEC
   ```

   Note: You can remove this exit after the upgrade.

2. Verify that all DASD volumes in the 6.2.0 system which will hold minidisks created during the upgrade process are configured in the DASD CONFIG file.

   Note: The volumes are specified on the worksheets you fill in by following the IBM procedures in the "Plan Your Installation Upgrade" chapter of the *z/VM Installation Guide Version 6 Release 3*.

3. Add GRANT *ALL TO MIGMAINT to the AUTHORIZ CONFIG file.

4. Define the necessary resource authorizations before proceeding. For instructions, refer to the section "Review External Security Manager (ESM)" in the chapter "Finish the STAGE1 Upgrade" in the IBM *z/VM Installation Guide Version 6 Release 3.*

5. Run the steps in the following chapters of the *z/VM Installation Guide*:

   a. Chapter 20: "Plan Your Upgrade Installation"

   b. Chapter 21: "Set Up for Your Upgrade Installation"

   c. Chapter 22: "Install a z/VM V6.3 Work System for Your Upgrade Installation".

6. Perform these changes in Chapter 23, "Generate the STAGE1 Changes File":
   a. Before you issue the INSTUPGR STAGE1 (PRIME command, LINK and ACCESS the CA VM:Director PUBLIC disk at a filemode earlier in the search order than the Y-disk, unless you already deployed the PUBLIC files to the Y-disk.
   b. Issue the following command:

      ```
      INSTUPGR STAGE1 ( PRIME
      ```

      If errors occur, CA VM:Director logs any problems in the following file:

      ```
      UPGDMIXT $LOGFILE E
      ```

      If CA VM:Director reports any problems, correct them and reissue INSTUPGR.
7. Perform these changes in Chapter 24, "Update Your Current System With the STAGE1 Changes":
   a. Before you issue the INSTUPGR STAGE1 (COMMIT command, examine files with filetype $DIRADD$, generated by the STAGE1 PRIME processing.
   b. Look at all MDISK statements to determine the volume serial numbers of all 630RLx volumes where minidisks exist.
   c. Add all the 630RLx volumes to the CA VM:Director DASD CONFIG file so that minidisks on those volumes can be allocated.
   d. Issue the CONFIG DASD product command to configure the 630RLx volumes in the same allocation subpool containing the 620RLx volumes.

      Add an IGNORE statement in the DASD CONFIG file so that any full minidisks in the MAINT630 directory entry can be created.
   e. Issue VMDIRECT CONFIG DASD and add a statement like the following example:

      ```
      IGNORE MAINT630 131
      ```
   f. Check the MAINT630 $DIRADD$ file on the MIGMAINT 2CF0 disk to verify that all full volume minidisks have an IGNORE statement in the DASD CONFIG file.
   g. LINK and ACCESS the CA VM:Director PUBLIC disk at a filemode earlier in the search order than the Y-disk, unless you already deployed the PUBLIC files to the Y-disk.
   h. Issue the following command:

      ```
      INSTUPGR STAGE1 ( COMMIT
      ```

      If errors occur, CA VM:Director logs all problems in the following file:

      ```
      UPGDMIXT $LOGFILE E
      ```

      If CA VM:Director reports any problems, correct them and re-issue INSTUPGR.
8. Define the necessary resource authorizations for new user IDs added for z/VM 6.3.0. Refer to the section "Review External Security Manager (ESM)" in the chapter "Finish the STAGE1 Upgrade in the IBM *z/VM Installation Guide Version 6 Release 3.*

   Rework your local modifications.

9. Run the steps in Chapter 26 of the IBM *z/VM Installation Guide Version 6 Release 3*, "Stop Your Production Workload and Backup Your System".
10. Perform these changes from Chapter 27, "Generate the STAGE2 Changes File":
    a. Before you issue the INSTUPGR STAGE2 (PRIME command, LINK and ACCESS the CA VM:Director PUBLIC disk at a filemode earlier in the search order than the Y-disk, unless you already deployed the PUBLIC files to the Y-disk.
    b. Issue the following command:

       ```
       INSTUPGR STAGE2 ( PRIME
       ```

       If errors occur, CA VM:Director logs any problems in the following file:

       ```
       UPGDMIXT $LOGFILE E
       ```

       If CA VM:Director reports any problems, correct them and re-issue INSTUPGR.
11. Perform these changes in Chapter 28, Update your current system with the STAGE2 changes:
    a. LINK and ACCESS the CA VM:Director PUBLIC disk at a filemode earlier in the search order than the Y-disk, unless you already deployed the PUBLIC files to the Y-disk before issuing:

       ```
       INSTUPGR STAGE2 ( COMMIT
       ```
    b. After the IPL 190 step, LINK and ACCESS the CA VM:Director PUBLIC disk at a filemode in front of the Y-disk, unless you already deployed the PUBLIC files to the Y-disk before reissuing:

       ```
       INSTUPGR STAGE2 ( COMMIT
       ```

       If errors occur, CA VM:Director logs any problems in the following file:

       ```
       UPGDMIXT $LOGFILE E
       ```

       If CA VM:Director reports any problems, correct them and reissue INSTUPGR.
12. Perform the remaining steps in the IBM documentation.

Index

*

* authorization • 129
*AC= special comments
  conflict with ACCOUNT user exit • 235
  definition • 235
  in skeleton files • 165
  searching for when assigning account numbers to user IDs • 88
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*ALL authorization • 129
*ANY authorization • 129
*ED= special comments
  definition • 235
  in MANAGERS file • 225
  in skeleton files • 165
  in VMSECURE GLOBALS and VMDIRECT GLOBALS file • 241
  in VMSECURE POSIX and VMDIRECT POSIX file • 245
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*ENROLL special comments
  definition • 236
  in skeleton files • 165
  IPLSTMT option • 236
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*EX= special comments
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*FL= special comments
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*FP= special comments
  definition • 237
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*HS= special comments
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*LA= special comments, where allowed • 233
*LL= special comments
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*NB= special comments
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*NEWUSRS authorization • 108
*PW= special comments
  automatic password expiration • 150
  definition • 238
  in skeleton files • 165
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*SP= special comments
  definition • 238
  in skeleton files • 165
  to enroll a user • 236
  where allowed • 233
  who adds to files • 233
  who sets value • 233
*US= special comments
  where allowed • 233
  who adds to files • 233
  who sets value • 233

6

600 series minidisks, created when minidisks moved • 129

A

abends, causing with the ABEND command • 22
account comments • 235
account numbers
  assigning to user IDs • 88
  changing for user IDs • 165
  validating • 88
ACCOUNT statement in skeleton files • 165

ADDENTRY command, creating user IDs with • 145
adding
  file pools • 209, 213
  INCLUDE statements to directory entries • 161
  SFS administrator • 208
  SFS managers • 209
  user storage groups • 214
ADDMDISK command, creating a minidisk • 172
ADMIN command
  changing skeleton files • 168
  creating skeleton files • 168
  deleting skeleton files • 169
ADMIN GLOBALS authorization • 241
ADMIN POSIX authorization • 245
ALL keyword user ID • 114
ALL, reserved user ID • 143
allocating minidisks
  interactively • 173
  through programming applications • 172
  to existing user IDs • 173
allocation limits
  changing for SFS user storage groups • 203
  definition • 195
  for automatically configured SFS user storage groups • 188
  setting for SFS managers on user storage groups • 209
allowing logons by changing user ID passwords • 152
ALTVOL startup option • 20
analyzing minidisk usage and space requirements • 174
ANYPOOL authorization • 129
ANYSKEL authorization • 129
AUDINV startup option • 20
audit records, backing up • 80
auditing
  AUDT minidisk • 62, 80
  including invalid passwords in audit records • 20
auditing:in VM:Director • 41
auditing:in VM:Secure • 41
AUDT minidisk
  backing up • 81
  changing size of • 62
authority lists
  general information • 117
  purpose • 105, 118
  using in other authorization lists • 119
AUTHORIZ CONFIG configuration file
  overview • 51
  sample file • 52
authorization, granting to
  directory managers • 256
  general users • 255
authorizations
  creating • 117, 118
  for command processing • 129
  for SFS administrator • 128
  for SFS managers • 128
  granting to parts of a command • 107
  guidelines • 105
  OVER in GRANT AUTHORITY and REVOKE AUTHORITY commands • 113
  overview • 105
  password requirement, waiving • 107
  pattern matching • 122
  querying • 123
  recommended for • 125, 128
  required • 106, 148
  sample • 124
  SFS administrator • 40
  SFS managers • 215
  special queries • 124
  Systems Management API routines • 251
  to change password in object directory • 129
  to create user IDs • 108
  to use commands • 106, 113
  to use utilities • 106
  using user ID lists and authority lists • 120
  withholding • 116
authorizations, required to
  authorize others access to SFS files or directories • 110
  change • 110
  remove authorization for others to access SFS files or directories • 110
  use the Systems Management API routines • 251
  work with • 241, 245
authorizations, specific
  * • 129
  *ALL • 129
  *ANY • 129
  *NEWUSRS • 108
  ADMIN GLOBALS • 241
  ADMIN POSIX • 245
  ANYPOOL • 129
  ANYSKEL • 129
  CACHED • 129
  CHPW • 129

DIAGPCHK • 129
GRANT AUTHORITY • 110
MAINTMAN • 108
MOVERO • 129
NOCOPY • 129
NOFORMAT • 129
NOFORMAT ADD • 129
NOFORMAT DELETE • 129
NOFORMAT MOVE • 129
NOPASS • 107, 129
OVERRIDE NEWUSER • 129
REVOKE AUTHORITY • 110
SURROGAT • 129
automatic password expiration • 150

B

backing up
audit records • 80
AUDT minidisk • 80
COMMON minidisk • 80
customized user exits • 81
DRCT minidisk • 78
HOLD minidisk • 81
LOCAL minidisk • 81
modified macros • 81
Servant Facility information • 80
skeleton files • 169
source directory • 78
system-specific files • 81
user IDs on hold • 81
VMSECURE MANAGERS and VMDIRECT MANAGERS file • 81
VMSI minidisk • 82
backing up:VM:Secure and VM:Director miniidsks • 78, 80
backups
card image • 79
mirror image • 79
braces • 26
brackets • 26

C

CACHED authorization • 129
calculating
cylinder numbers for CP directory changes • 66
page numbers for CP directory changes • 73
card image backup • 79
changing
account numbers • 165
distribution codes • 165
minidisk information • 176
minidisk owner • 178
minidisk virtual address • 178
passwords • 152
POSIX information • 244
SFS manager authorities • 209, 213, 214
user storage group allocation limits • 203
checking user ID password expiration • 149
CHGMDISK command, moving a minidisk • 177
CHNGPASS user exit • 267
CMSUT3, reserved user ID • 143
command authorizations, configuration file that stores them • 51
commands
authorizations required to use • 106
creating lists of user IDs authorized to use specific commands • 117
granting • 106, 125, 128
help with • 25
listing which ones are currently executing • 21
tracing execution • 221
user–entered • 26
withholding authorizations for • 116
COMMON minidisk
backing up • 82
when required • 85
compressing disk space • 181
configuration files
AUTHORIZ CONFIG contents • 51
backing up • 81
DASD CONFIG contents • 52
dynamic reconfiguration based on • 36
for DASD • 52
for user authorizations for commands • 51
types • 36
configuration statement for ESM • 265
CONSOLE statement, in skeleton files • 165
control blocks
calculating size of • 72
gaining more paging space for • 71
control files
creating user IDs • 148
putting user IDs on hold • 154
reactivating user IDs • 155
removing user IDs • 157
control statements
allowed in directory profiles • 159

not allowed in directory profiles • 159
PROFILE, where allowed • 159
conventions
examples • 26
format • 26
copyright screen, preventing display of • 20
CP commands, specific
SET SHARE • 219
SET SRM • 219
CP directory
changing allocation of • 64
moving • 64
statements, in skeleton files • 165
CP object directory
increasing size of • 64
moving • 64
CPFMTXA command • 69
creating
directory profiles • 159, 160
dumps • 22
minidisks • 172, 173
skeleton files • 168
user IDs • 145, 148
customizing user authorizations • 105
cylinder numbers, calculating starting and ending numbers for CP directory changes • 66

D

DASD
compressing • 181
configuration file that stores it • 52
management, overview • 38
DASD CONFIG configuration file
adding DASD to • 180
effects of changing records • 55
overview • 52
record sequence requirements • 52
DASD:adding to the VM:Secure or VM:Director configuration • 180
deactivating user IDs • 153
defining SFS to VM:Secure or VM:Director:adding • 208, 209
defining SFS to VM:Secure or VM:Director:changing • 203, 212
defining SFS to VM:Secure or VM:Director:purpose • 188
defining SFS to VM:Secure or VM:Director:removing • 205, 215
defragmenting DASD volumes • 181
DEFSIZE record • 228
DELENTRY command
removing user IDs on hold with • 156
removing user IDs with • 156
DELETE authorization • 129
deleting
directory managers • 43
directory profiles • 160
file pools • 205, 213
INCLUDE statements from directory entries • 163
minidisks • 181
several user IDs at the same time • 157
skeleton files • 169
user IDs • 156, 173
DEVTYPE record
order in DASD CONFIG file • 53
DIAGPCHK authorization • 129
required for DIAGNOSE ’A0’ subcodes X’04’ and X’0C’ • 129
directory database
locking • 21
minidisks in • 128
overview • 35
recommended virtual addresses for minidisks • 128
directory entries
adding profile statements to • 161
editing • 161
expanding • 161
POSIX information • 244
purpose • 188
special comments for • 233
specify a PROFILE EXEC for user IDs through • 238
tracking • 237
directory links, verifying • 223
directory management • 38
directory managers
allowing to use skeleton files • 169
authorization required to create user IDs • 148
authorizations • 40
changing • 45
creating • 43
define • 231
deleting • 46
for system administrators • 48
identify • 231
limiting • 43, 231

listing skeleton files they can use • 169
naming restrictions • 43
prevent from using skeleton files • 170
recommended authorization • 125
responsibilities • 22
sample definition of • 45
setting default device type • 230
specifying • 43, 231
user ID limit • 230
directory managers:adding to VM:Secure or VM:Director • 43
directory profiles
adding to directory entries • 161
advantages of using • 38
control statements • 159
copying into new ones • 160
creating • 159, 160
deleting • 160
editing • 160
expanding directory entries to include • 161
including in directory entries • 161
listing user IDs that currently use a particular one • 160
POSIX information • 244
required control statements • 159
special comments for • 233
specify account numbers for user IDs in • 235
using • 160
directory space, finding for CP directory allocation • 73
directory, moving • 64
disabling user IDs by changing logon passwords • 152
disaster recovery • 77
distribution codes, changing for user IDs • 165
DMSJNE routine • 216
DMSPARMS configuration file • 216
DRCT minidisk
backing up • 78, 79, 80
changing size of • 62
dummy minidisks • 74
making volume changes • 61
duplicate directory statements • 161
dynamic reconfiguration
meaning • 36
reclaiming workspace • 65, 72

E

EDIT command, creating user IDs with • 145
editing directory profiles • 160, 161
enabling user IDs by changing logon passwords • 152
ending VM:Secure or VM:Director:abnormally • 22
ending VM:Secure or VM:Director:effects of immediate end • 21
ending VM:Secure or VM:Director:how to correct effects of immediate end • 21
end-user access to ESM-controlled VM:Director commands • 269
enrolled file pool comments • 237
ESM
command authorization • 253
configuration statement syntax • 265
rules and resource names • 265
security impact • 266
examples, conventions • 26
expanding directory entries with directory profile contents • 161
expiring passwords
automatically • 150
user ID directory entries required for automatic expiration • 150
EXTENT record, order in DASD CONFIG file • 53
extents
enlarging CP directory • 72
filling up free disk space within • 181

F

file pools • 205, 213
list of file pools user IDs can use • 237
specify in skeleton files which ones user IDs can use • 236
file space overallocation, limits • 202, 203
FILEPOOL EXEC, requirement for using • 17
filling up free disk space within extents • 181
format conventions • 26
formatting
authorization rules • 254
directory extents • 69
fragmented disk space, correcting • 181
free space, finding for CP directory allocation • 73

G

GENERAL skeleton file • 165
general users, responsibilities • 22

giving VM:Director access to ESM-controlled resources • 267
GRANT AUTHORITY authorization • 110
GRANT AUTHORITY command • 216
granting authorization to
directory managers • 256
general users • 255
grouping commands and user IDs in lists • 117

H

help
commands • 25
messages • 33
screens and menus • 23
HOLD minidisk
backing up • 82

I

IDLIMIT record • 230
IGNORE record, order in DASD CONFIG file • 53
inactive user IDs
definition • 235
deleting • 156
querying • 123
INCLUDE statements
adding to directory entries to include directory profiles • 161
deleting from directory entries • 163
replacing in directory entries • 162
initialization process
activities during • 241
POSIX activities during • 245
IPLable tapes, how to create • 79
IPLDISK minidisk
backing up • 81
IPLDISK procedure, automating password expiration • 150
IPLSTMT option, *ENROLL special comment • 236

L

last-edit comments • 235
limiting authorizations • 116
link passwords, verifying changes made to • 224
LINK statement in skeleton files • 165
Linux images, cloning • 173
LIST record continuation character • 117
listing skeleton files that can be used by directory managers • 169
lists
authority lists • 118
querying authorizations of list members • 124
user ID lists • 118
using in other lists • 119
LOCAL minidisk
backing up • 82
location, querying for SFS file pools • 207
locks
listing which ones are currently active • 21
querying • 21
logon password, using with NOPASS authorization • 129
LOGONBY authorization, DIAGPCHK authorization required to check for • 129
logons
disabling for user IDs by changing their passwords • 152
enabling for user IDs by changing their passwords • 152
re-enabling for user IDs by changing their passwords • 152

M

machine mode, changing default • 242
macros
authorization required to trace • 221
tracing execution • 221
usable by SFS • 220
MAINT command
moving minidisks • 178
putting user IDs on hold with • 154
reactivating user IDs with • 152
removing user IDs with • 156
MAINTMAN authorization • 108
MANAGE command
moving minidisks • 177
putting user IDs on hold with • 153
reactivating user IDs with • 155
removing user IDs with • 156
MANAGER record • 231
MANAGERS file
adding directory managers to • 43
allowing use of a skeleton file • 169
defining managers • 231
managing the system administrator • 48
overview • 20
skeleton files • 232

special comments for • 233
user ID limit • 230
withholding use of a skeleton file • 170
MANAGERS file records, specific
DEFSIZE • 228
DEVTYPE • 229
IDLIMIT • 230
MANAGER • 231
SKELETON • 232
managing the system administrator • 48
MDISK statements, allowed in skeleton files • 165
MDSKSCAN command
sample report • 174
scanning minidisks for content and usage • 174
menus, help with • 23
messages
help with • 33
migrating minidisk files to SFS, description • 217
MINIDISK user exit • 224
minidisks
600-6FF range • 129
access modes, verifying • 224
address restrictions • 172
analyzing for space usage and requirements • 174
AUDT, changing size of • 62
authorization required to • 129
changing • 178
creating • 67, 74, 173
default device type for directory managers • 230
default sizes • 228
defragmenting • 181
deleting • 181
DRCT • 61, 62
files, migrating to SFS • 217
migrating SFS files • 217
moving • 177, 178
required for SFS management • 199
restrictions on assigning • 172
transferring ownership • 178
mirror image backup • 79
MIXED startup option • 20
MOVERO authorization • 129
moving minidisks
from one user ID to another • 178
interactively • 173
through programming applications • 177
to defragment • 181
to free up space for CP directory • 73
using programming applications • 177
MULTIPLE command
putting user IDs on hold with • 153
reactivating user IDs with • 155
removing user IDs with • 156

N

nickname resolution • 216
NOFORMAT
DELETE authorization • 129
MOVE authorization • 129
NOFORMAT ADD authorization • 129
NOLINK record, order in DASD CONFIG file • 53
NOLOG
authorization • 129
changing password from • 152
command, disabling a user ID with • 152
purpose of password • 152
setting password to • 152
NOPASS authorization • 107, 129
notational conventions • 26
notations
examples • 26
syntax • 26

O

object directory
increasing size of • 71
moving • 64
operating system requirements • 17
OVER in GRANT AUTHORITY and REVOKE AUTHORITY authorizations • 113
OVERRIDE NEWUSER authorization • 129

P

page numbers, starting and ending for CP directory changes • 73
pages of space, calculating number to initialize the directory • 71
PASSWORD user exit • 266
passwords • 152
authorization to waive • 107
changes • 238
changing to • 152
checking • 129, 149
directory entry • 39
expiring automatically, user ID directory entry requirements • 150

history comments • 238
history-how to track for a user ID through a directory entry • 238
NOPASS authorization • 129
skipping verification • 129
verifying • 129, 224
pattern matching • 122
for command authorizations • 122
in authorizations • 122
performance considerations
delegating long-running tasks • 41
effect of directory profiles • 39
effect of skeleton files • 38
POSIX
changing • 244
maintaining group names and group IDs • 243
VMSECURE POSIX and VMDIRECT POSIX file • 245
POSIX GID • 243
POSIX UID • 243
POSIX, reserved user ID • 143
POSIX:directory statements supported by VM:Secure and VM:Director • 244
POSIX:objects supported by VM:Secure and VM:Director • 243
POSIXGLIST directory statement • 244
POSIXINFO directory statement • 244
POSIXOPT directory statement • 244
processes, querying active • 21
PROFILE control statements, where allowed • 159
PROFILE EXEC
comments • 238
SET LDRTBLS statement • 17
specifying • 238
prototypes
for creating user IDs • 145
using to create directory profiles • 160
PUBLIC keyword user ID • 114
putting
several user IDs on hold at the same time • 154
user IDs on hold • 153

Q

QUERY command, selecting user IDs for expiration with • 150
querying
active processes • 21
authorizations • 123
inactive user IDs • 151
lock on objects • 21
locks on objects • 21
POSIX information • 244
SFS file pool information • 207
user IDs that use a particular directory profile • 160

R

RACROUTE resource names • 257
reactivating user IDs • 155
records, user–entered • 26
re-enabling user IDs by changing logon passwords • 152
removing
directory managers • 46
file pools • 205, 213
SFS administrator authority • 208
replacing INCLUDE statements in directory entries • 162
requirements, operating system • 17
reserved user IDs • 143
reserved virtual addresses • 172
RESID record • 247
responses • 26
restricting authorizations • 116
resynchronizing configuration of SFS • 207
reviewing SFS manager authorities
allocable file pools • 213
allocable user storage groups • 215
REVOKE AUTHORITY authorization • 110
REVOKE AUTHORITY command • 216
RPC server for Systems Management API • 247
Rules Facility
DIAGPCHK authorization • 129
overview • 20
rules, formatting authorization • 254
running VM:Secure and VM:Director disconnected • 18

S

sample exit program, VMXEXIT9 XASSEMBL • 267
SAVEDIR startup option • 20
screens
help with • 23
how to move through • 23
security
basic features • 16
impact • 266

Servant Facility
backing up components of • 80
disabling • 87
enabling • 87
how to set up • 85
in Systems Management API • 247
issuing GRANT AUTHORITY and REVOKE AUTHORITY commands • 216
maximum number of worker user IDs allowed • 84
names of worker user IDs • 84
overview • 154
requirements for setting up • 85
status of • 87
service virtual machine
starting • 18
SET LDRTBLS statement for VM:Secure and VM:Director PROFILE EXEC • 17
setting up the Servant Facility • 85
SFS
file spaces, granting and revoking access • 216
improving system performance when using • 220
macros • 220
migrating files to or from minidisks • 217
SFS administrator
adding • 208
authorization • 128
functions of • 184
overview • 20
purpose • 183
removing • 208
SFS enrollment comments • 236
SFS file pools
adding • 209, 213
changing • 213
granting access • 216
querying status of • 207
removing • 213
reviewing for SFS managers • 213
revoking access • 216
SFS macros, identifying to VM:Secure or VM:Director • 220
SFS managers
adding • 209
assigning • 209
authority required to become SFS managers • 184
authorization • 209
changing • 212
definition • 209
functions • 184
removing • 215
source of allowed allocations • 184
who can become an SFS manager • 209
SFS user storage groups
adding • 214
allocation limits when automatically configured • 188
changing • 203, 215
interaction of user storage group allocation and SFS managers allocation limit • 196
removing from SFS managers • 215
reviewing for SFS managers • 215
space overallocation • 195
skeleton files
allow directory managers to use • 169
allowed contents • 165
assigning a directory manager • 232
changing • 168
CP directory statements • 165
creating • 168
deciding which ones you need • 165
definition of • 165
deleting • 169
GENERAL • 165
listing those used by a directory manager • 169
overview • 165
preventing directory managers from using • 170
special comments • 233
allowed in • 165
specifying
a PROFILE EXEC for user IDs through • 238
account numbers for user IDs in • 235
password history tracking for a user ID through • 238
SFS allocations for user IDs in • 236
system-supplied • 165
SOURCE startup option • 20
space overallocation in SFS user storage groups
purpose • 195
range allowed • 195
special comments
account numbers specified in *AC= special comments and ACCOUNT user exits • 235
account numbers user IDs can use • 235
allowed in skeleton files • 165
enrolling a user and *SP= • 236
file pools user IDs are enrolled in • 237

last-edit tracking • 235
overview • 233
password history tracking • 238
SFS enrollment specifications • 236
specify PROFILE EXEC • 238
starting VM:Secure • 18
startup options
ALTVOL • 20
AUDINV • 20
changing • 19
MIXED • 20
SAVEDIR • 20
SOURCE • 20
status, querying for SFS file pools • 207
SUBPOOL record
example • 54
order in DASD CONFIG file • 53
syntax conventions • 26
system administrator
authorization overview • 40
managing • 48
recommended authorization • 125
responsibilities • 22
system administrator, default name • 26
system performance, ways to improve
VM/ESA systems • 219
when using SFS • 220
SYSTEM, reserved user ID • 143
SYSTEMMP, reserved user ID • 143
systems management API support
overview • 247
RESID record • 247
VSMSERVE • 247
systems management API support:VM:Secure
authorizations • 251

T

terminating VM:Secure or VM:Director:abnormally • 22
terminating VM:Secure or VM:Director:how to correct effects of immediate termination • 21
terminating VM:Secure or VM:Director:immediately • 21
TRACE command, purpose • 221
tracing macros
authorization required to trace • 221
overview • 221

U

user exits
CHKPASS (VMXEXIT2) • 266
CHNGPASS(VMXEXITJ) • 267
MINIDISK • 224
user ID lists
general information • 117
purpose • 105, 118
querying authorizations of • 124
using in other user ID lists • 119
user IDs
authorization needed to create • 108, 148
creating
based on control file contents • 148
based on other user IDs • 145
minidisks for • 172
minidisks for existing user IDs • 173
minidisks for interactively • 173
several at the same time • 148
through programming applications • 145
deactivating • 153
deleting • 156
based on control file contents • 157
interactively • 157
through programming applications • 156
directory managers' requirements for creating • 148
invalid names • 143
listing those that use particular profile • 160
maximum directory manager can create • 230
maximum workers allowed through the Servant Facility • 84
moving minidisks for • 177
interactively • 178
through programming applications • 177
names of workers created through the Servant Facility • 84
password expiration • 149
putting on hold • 153
by a command • 153
several at a time • 154
using a screen menu • 153
querying inactive • 151
reactivating
by a command • 155
overview • 154
several at a time • 155
using a screen menu • 155

restrict how many a manager can own • 230
system administrator, default • 26
using with
OVERRIDE NEWUSER authorization • 129
SURROGAT authorization • 129
valid names • 143
withholding authorizations to use specific commands • 116
USER statement in skeleton files • 165
user word, specifying in the skeleton file • 165
USER, reserved user ID • 143
user–entered commands • 26
user–entered records • 26
utilities
authorizations required to use • 106
granting and withholding authorizations to use • 106

V

virtual address restrictions • 172
VM:Director:basic security features • 16
VM:Director:giving access to ESM-controlled resources • 267
VM:Director:giving end-users access to ESM-controlled commands • 269
VM:Director:managing minidisks for • 176
VM:Director:releasing to users • 22
VM:Director:running disconnected • 18
VM:Director:shutting down • 20
VM:Director:specifying page space in the PROFILE EXEC • 17
VM:Director:stopping • 20
VM:Manager, end-user screens and CHPW authorization • 129
VM:Secure:basic security features • 16
VM:Secure:managing minidisks for • 176
VM:Secure:releasing to users • 22
VM:Secure:running disconnected • 18
VM:Secure:shutting down • 20
VM:Secure:specifying page space in the PROFILE EXEC • 17
VM:Secure:stopping • 20
VM:Secure:virtual address restrictions • 172
VMDIRECT POSIX file • 245
VMSECURE POSIX file • 245
VMXEXIT2 user exit • 266
VMXEXIT9 XASSEMBL sample exit program • 267
VMXEXITJ user exit • 267
VMXSYS startup options • 19

W

WITHHOLD record • 116
withholding authorizations • 116