CA ACF2™ for z/VM

Administrator Guide

r12

Second Edition

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.

Contents

Chapter 14: Maintaining Resource Rules with the Full-Screen Feature 299

Chapter 15: Maintaining Resource Rules with the ACF Command 335

Chapter 1: Introduction

As a security administrator, you are an important part of the security process at your site. You are responsible for protecting your company's resources used in its daily operations.

This guide provides information on assigning logonids to new users, writing rules to control access to data and resources, and creating special rules to control where a user can log on, when a user can log on, and so on. It is aimed at the beginning user and administrator who has limited knowledge of CA ACF2™ for VM (CA ACF2 for VM). You should read this guide and refer back to it as a source of important information.

This section contains the following topics:

Required Reading (see page 21)

Required Reading

Before you begin using CA ACF2 for VM, you should read the *General Information Guid*e. It contains introductory information on the functions and concepts of CA ACF2 for VM, including

- Structure and components of CA ACF2 for VM
- Logonids and UIDs
- Rules controlling access to data and resources
- CP command and diagnose control
- Backup and recovery features
- Reports and utilities available for administering CA ACF2 for VM

  Installation and maintenance material

Chapter 2: Getting Started

This chapter explains the basic concepts you need to know to use CA ACF2 for VM. It also explains how to get started using
CA ACF2 for VM. After you read this chapter, you will know:

- What a logonid is
- What a UID is
- What a password is
- What a password phrase is
- How to log onto an CA ACF2 for VM-secured system
- How to enter and change your password

You must understand these concepts to successfully administer an CA ACF2 for VM security system.

This section contains the following topics:

What is a Logonid?

The easiest way to understand a logonid is to think of it as a one- to eight-character string that identifies you to CA ACF2 for VM. You are assigned a logonid that is uniquely your own. No other user can have the same logonid. In VM, your logonid is the same as your user ID.

Your logonid is related to a logonid record that defines privileges that determine what you can and cannot do on the system. It also provides a history of what you have done on the system, the number of security and password violations you have committed, and so on. The CA ACF2 for VM Logonid database must contain a logonid record for each VM user ID. For additional information about the logonid record, see “About the Logonid Record”.

What is a UID?

The User Identification string (UID) is a combination of characters from fields in the logonid record, such as site code, department code, job function, and so on. Your site decides how long the UID should be and what its contents are. Because the UID is so critical in good rule writing, it is essential to familiarize yourself with your site's UID construction during implementation. Constructing a good UID makes rule writing easier because it lets you group users. For introductory information about the UID, see the *Implementation Planning Guid*e.

What is a Password?

A password is a secret word that only you and CA ACF2 for VM know. Your password tells CA ACF2 for VM that you really are who you say you are. The difference between a logonid and a password is that any number of people can know your logonid, but only you know your password. It is between one and eight characters long.

While both CA ACF2 for VM and CA ACF2 for VM for z/OS support the special characters @, #, and $, these characters are special characters to VM. The # character is the default LINEND character. When this character is encountered in an input line, the line splits at that point. The @ is the default CHARDEL character. When this character is encountered in an input line, the previous character is deleted as if the @ was a destructive backspace. Because of this, do not use these two characters in a password that VM uses. The only way that you can use them is to place the ESCAPE character (by default, ") before the @ or #. The ESCAPE character causes the @ and # to be treated as standard characters instead of special characters.

CA ACF2 for VM never displays your password in clear form. Passwords are one-way encrypted when you log on. If you forget your password at any time, no one can look it up for you. A security administrator must assign you a new password that CA ACF2 for VM forces you to change when you log on.

What is a Password Phrase?

A password phrase (supported on Version 5 Release 3.0 and above) is similar to a password, but is longer than 8-characters and can contain multiple words. A password phrase in VM can be between 9- and 200-characters long. It can be made up of upper and lower case characters, numeric digits, spaces, and selected special characters. A user can be defined with both a password and a password phrase. They are maintained separately and either can be used to logon to VM if the site or user is enabled to use password phrases. Passwords are stored in the logonid record and password phrases are stored in the PWPHRASE segment of the USER profile.

While both CA ACF2 for VM and CA ACF2 for z/OS support the special characters @, #, and $, these characters are special characters to VM. The # character is the default LINEND character. When this character is encountered in an input line, the line splits at that point. The @ is the default CHARDEL character. When this character is encountered in an input line, the previous character is deleted as if the @ was a destructive backspace. Because of this, *do not* use these two characters in a password phrase that VM uses. The only way to use them is to place the ESCAPE character (by default, ") before the @ or #. The ESCAPE character causes the @ and # to be treated as standard characters instead of special characters.

Important! CA ACF2 for VM never displays your password phrase in clear form. If you forget your password phrase at any time, no one can look it up for you. A security administrator must assign you a new password phrase that CA ACF2 for VM forces you to change when you log on.

Logging onto CA ACF2 for VM

The standard CA ACF2 for VM logon procedure is illustrated below.

```
Enter one of the following commands:

  LOGON userid          (Example: LOGON VMUSER1)

  DIAL userid           (Example: DIAL VMUSER2)
  MSG userid message    (Example: MSG VMUSER2 GOOD MORNING)
  LOGOFF

logon tlcams (press enter)

LOGON tlcams
ACFpgm244R Enter CA-ACF2 password

password (press enter)

A
ACFpgm137I TLCAMS last system access at 15.22 on 11/05/99 from GRAF419
DASD 190 LINKED R/O; R/W BY VMGEN; R/O BY 049 USERS
DASD 19D LINKED R/O; R/W BY VMGEN; R/O BY 037 USERS
LOGON AT 17:34:53 CST TUESDAY 11/09/99
Your site can have a different logon screen, but the essentials are the same.
Enter your logonid and password.
```

When you enter your password, you do not see it. CA ACF2 for VM never displays your password. This is so no one passing by your terminal or sitting next to you can see it. CA ACF2 for VM protects your password on the system. It is up to you to protect it outside of the system. Do not write it down or share it with others. The next section, Entering Your Password, provides more information about your CA ACF2 for VM password.

A

The ACFpgm137I message tells you the time, date, and location where your logonid last accessed the system. Always review this information to see if anyone else used your logonid without your knowledge. If you find unauthorized uses, change your password and contact your security administrator immediately.

Entering Your Password

CA ACF2 for VM demands that you use a password to ensure that you are the only person who can use your logonid. This password is one- to eight-alphanumeric characters. Your security administrator defines a minimum password length and decides how often you can or must change your password. Throughout the rest of the guide, we will assume that your site has defined a minimum password length of five characters.

Your password is your protection against unauthorized use of your logonid. You are accountable for what your logonid attempts to do. Keep your password to yourself and change it often.

Note: If you have been enabled to use a password phrase, when you are prompted for your CA ACF2 for VM password, you may enter your password phrase instead of your password. If password phrases have been enabled globally, or individually with the PWPALLOW privilege, any password entered that is over 8-characters will be validated as a password phrase. Password phrases are only supported on systems running z/VM Version 5 Release 3.0 and above.

Changing Your Password

Normally, you can change your password whenever you want, but your site may have defined a maximum time period between password changes. If you have exceeded this time period, CA ACF2 for VM does not let you log on until you specify a new password. To prevent users from changing their passwords too often, your site can also define a minimum time period between password changes. Your site may also assign a password to you instead of letting you change it. For easy explanation, our examples assume that you can change your own password.

To change your password, log onto CA ACF2 for VM (as explained earlier in this chapter). Enter your old password, a slash, and then your new password.

To change your password, follow this procedure:

```
LOGON TLCAMS
ACFpgm244R Enter CA-ACF2 password
old password/new password (press enter)

ACFpgm246R Enter new CA-ACF2 password again to verify

new password (press enter)

ACFpgm137I TLCAMS last system access at 17.34 on 11/09/99 from GRAF419
ACFpgm129I Password successfully altered
```

CA ACF2 for VM asks you to type your new password in again to verify your new password. This is to ensure that you really know what your new password is and your finger did not slip while you were typing it in the first time. The following message confirms that you have a new password:

```
ACFpgm129I Password successfully altered
```

Administering CA ACF2 for VM

After you have accessed the system, you can begin to maintain logonids and rules. There are several ways to perform these tasks:

- The full-screen feature
- The ACF command
- Local update procedures.

Subsequent chapters explain how to use the first two methods. The last method is not described in this guide. (Your site can set up local procedures that consist of panels and programs written to maintain records on the CA ACF2 for VM databases. Contact your local systems programmer for details.)

Chapter 3: Using the Full-Screen Feature

You can maintain CA ACF2 for VM logonid records, access rules, resource rules, and generate reports through the full-screen feature and through the ACF command. The full-screen feature is a panel-driven feature that lets you select the panel for the function you want to perform. You just fill in the appropriate fields, execute the screen, and the job is done. Help is available for each screen and for each field on the screen. This chapter explains how to use the full-screen feature to administer CA ACF2 for VM.

You can also administer CA ACF2 for VM with the ACF command and its subcommands. The chapter "Using the ACF Command" in this guide explains how to use the ACF command.

When you finish this chapter, you will know:

- How to access the full-screen feature
- How to get help in a full-screen panel
- How to get help for messages
- How to use line commands for the full-screen feature
- How to use the PF and PA keys.

This section contains the following topics:

Accessing the Full-Screen Feature

To use the full-screen feature, enter the following command from the CMS Ready prompt:

```
ACFFS
```

When you press Enter, CA ACF2 for VM displays the Primary Option Menu. From this menu, you can select the function you want to use, such as those to create and maintain logonids, access rules, resource rules, and generate reports.

As you use the full-screen feature, you see a series of screens. These screens let you select the administrative function you want to perform. CA ACF2 for VM never lets you perform functions that you do not have the authority to perform. Privileges that you cannot grant do not appear on the screens.

The second line in the sample screen below is an example of a COMMAND line. Valid commands are explained in Full-Screen Commands later in this chapter.

```
M9PA-1300     Change User Information (1.3)                    CA ACF2 for VM

COMMAND ===> ____________________________________________________________
```

The second line in the sample screen below is an example of an OPTION line. Option lines are displayed on menu screens. You can enter your selection for the function you want to perform on this line. For example, if you want to work with access rules, enter 2 (Data Access Control) on the OPTION line of the screen below.

```
M9PA-0000     CA ACF2 for VM Primary Option Menu                CA ACF2 for VM
OPTION ===> ____________________________________________________________
                            TIME 17:11

      0 Change options or PF keys

      1 User Identification

      2 Data Access Control

      3 Resource Control
```

Enter a panel number on the ACFFS command line to bypass the main menu and view the selected panel immediately. For example, entering the following command puts you directly into the Change User Information (1.3) panel:

```
ACFFS 1.3
```

The following sections provide information you should be aware of before you use the full-screen feature. Remember: before you can start using the full-screen feature, you must be allowed access to the panels and help files. Your security administrator has the details.

Getting Help

If you need help while using the full-screen feature, you can place the cursor anywhere on the screen at any time and press your HELP PF key (the default is PF1, but you can redefine your own PF keys). CA ACF2 for VM provides you with online assistance for that screen. You can also place the cursor on any field (or field title) and press your HELP PF key. CA ACF2 for VM displays additional information about that field. If you place the cursor on the OPTION or COMMAND lines and press your HELP PF key, you receive information on what kind of values are permitted.

You can also enter HELP on the COMMAND or OPTION line on the screen and move the cursor to a position on the screen (as described in the above paragraph), then press Enter. You do not need to use a PF key.

The HELP command also lets you obtain additional help that is not available through the HELP PF key. You can enter:

HElp Help

To obtain information about the HELP command

HElp Pfkeys

To obtain information about the PF key defaults

HElp INPUT

To obtain information about valid input on the command line

HElp Message msgid

To obtain information about message.

Message Help

There are three ways to obtain help for the last message the CA ACF2 for VM full-screen facility issued.

1. Move the cursor to the third line on the screen and press your HELP PF key
2. Enter HELP on the command line, move the cursor to the third line of the screen, press Enter
3. Enter HELP Message on the command line.

To receive help for any message, enter:

```
HElp Message msgid
```

The value of msgid is the data that appears before the message text. You can normally drop the pgrmid in 4 through 6 places in the message ID. For example, both of the following command will display help for message ACF929E:

```
HELP MES ACFMFS929E
HELP M ACF929E
```

To review help information about messages, you must be linked to and have access to the disk that contains the help files. The full-screen feature honors EMSG settings of ON, TEXT, and CODE, indicating the type of messages you see and how you see them. If you need information about the help files or message settings at your site, contact your systems programmer.

Field Information

For all displayed screens, field names followed by a **===>** (arrow) indicate input fields. You can enter values in these fields. Field names ending with a colon (:) are display-only fields. You cannot enter values in these fields.

You can customize the screens and PF keys shown in this guide. If your screens are different, your site has changed them. For instructions on customizing screens, see the *Systems Programmer Guide.*

You can define most commands to PF keys or issue them on the command line at the top of most screens. You can alter your PF key definitions to any of the commands shown in this section.

The full-screen feature supports all of the commands listed below. We explain how to define your PF keys in the Changing Your PF Keys section.

Each screen displays only the functions that you can use on that screen. Capital letters represent the shortest abbreviation of the command. For example, entering CL is the same as entering the entire word CLEAR.

Command	Description
" (ditto)	Indicates to keep the same value of the identical field from the previous panel. You can only enter this command on a field, not on the command line.
= (equal)	Indicates to use the equal sign to go directly to a particular screen. For example, entering =6.2 on the command line displays the 6.2 (Select CA ACF2 for VM Reports) screen.
ACF	Passes the indicated operands to the ACF module in line mode. If no operands followed the command, ACF line mode is entered.

Command	Description
BAckward	Scrolls backward in the list. This is only applicable in screens that contain lists.
CAncel	Terminates processing and returns back to one screen level. This command is the same as QUIT.
Clear	Clears all input fields on this screen and displays again what was overtyped.
CMS	Passes the indicated operands to CMS for processing. If no operands followed the command, CMS prompts for a command.
CP	Passes the listed command and operands to CP. If no operands followed the command, you are prompted for a CP command.
CURSOR	Moves the cursor from the current position to the command line, or from the command line to the last position. You can only use this command if you have defined one of your PF keys to this command.
DEfaults	If you redefined values for some fields, this command sets the values back to what they were when the product was first installed.
DItto	Fills in all input fields on this screen as they were in the last transaction.
ENd	Ends the current processing (EXECUTE implied), and returns to the previous screen.
EXecute	Indicates that current processing is complete. All accumulated data is passed to CA ACF2 for VM for processing if there are no errors.
FORMat	Displays a boiler plate format of the type of data that is expected in all fields. Underscores (__) indicate alphabetic fields; mm/dd/yy represent date fields; nnn illustrates numeric fields; and b indicates Y or N bit fields.
FOrward	Scrolls forward in the list. Applies only to screens that display lists.
HElp	Displays help information about fields, screens, and using ACFFS. To view more information, enter HELP HELP on the command line of a screen.
MVSVM	Switches the screen to the appropriate OS/390 or VM screen and reformats the rule data for an OS/390 or VM access rule. This command is available only for access rules screens.
NExt	Displays the next screen in sequence.
PREvious	Displays the previous screen.

Command	Description
PRInt	Sends a print image of the displayed screen to the user's virtual printer queue.
QUit	Stops current processing and returns to the previous screen. This command is the same as CANCEL.
REtrieve	Retrieves the last command entered on the command line of this screen but does not execute it. You can retrieve up to 12 commands.
RETUrn	Finishes current processing and returns to the primary menu. This command is the same as issuing multiple END commands.
SAVE	CA ACF2 for VM saves the parameter information in a parameter defaults file. It saves parameters common to all reports in a file named ACFRPTS DEFAULTS. CA ACF2 for VM saves parameters unique to a report in a file named fn DEFAULTS, where fn is the filename of the panel (for example, M9PA6210 DEFAULTS). CA ACF2 for VM tries to save the defaults on the same disk as the screen. If this is not possible, it saves the defaults on the user's A-disk.

Full-Screen PF Keys

Following are the default values for the Program Function keys displayed at the bottom of each screen. CA ACF2 for VM only displays the PF keys that apply to the specified screen. Wherever a PF key can have more than one value, we explain both values.

Your site may have redefined the values of these keys. You can also select option 0 (keys) to change the PF keys at every menu level. If you need help to find out the value of the PF keys at your site, ask your systems programmer about the new settings.

PF1 Help

To get additional information on the screen you are using, place the cursor on the heading line or any field and press PF1. A help screen appears that describes the various fields of the screen. To get information on valid command or option line entries, place the cursor on that line and press PF1. For the PRIVILEGE sections of screens, you can place the cursor on any character in the PRIVILEGE name, press PF1, and a brief description of the PRIVILEGE you selected appears.

PF2 Print

Prints the screen.

PF3 Quit

Cancels the work done and returns to the previous screen. No processing is done.

PF4 Return

Returns to the primary menu.

PF5 Execute

Processes (adds, changes, or deletes) the values you have previously entered on the screen.

PF6 MVS<->VM

Switches the screen to the appropriate OS/390 or VM screen (available only for access rule screens).

PF6 Format

Indicates the field length and expected format of the input fields. Those fields that need numeric values will be filled in with ns. Fields that need alphabetic values will be displayed with underscores (__). Fields that require a Y (yes) or N (no) value are displayed with B. (Available only for user ID maintenance screens.)

PF7 Backward

Scrolls the screen toward the top.

PF8 Forward

Scrolls the screen toward the bottom.

PF9 Director

Switches between CA ACF2 for VM and CA Director. When you are in the User Identification or Access Rule Maintenance screens and switch to CA Director., CA ACF2 for VM passes the logonid that you specified on the screen to CA Director.. If the screen has no logonid field or you have not specified a logonid, the logonid of the person using the screen passes to CA Director. If your site is not running CA Director., this key will not be defined.

PF10 Save

Redefines default values for a screen. The parameter values you specified on the screen are saved as the default parameter values in a file called M9PAnumb DEFAULTS fm, where numb is the panel number (located in the first line of the screen on the left, for example, M9PA6000), and fm is the filemode where the DEFAULTS file resides. If the DEFAULTS file is accessed read only, CA ACF2 for VM places the file on the A-disk. If you are converting from Release 3.2 to 3.3, you must erase the old DEFAULTS file from the A-disk and create a new DEFAULTS file to activate the new format. To do this, save all your default values for each screen. Issuing DEFAULTS on the command line sets the report parameters back to the original default values shipped with the CA ACF2 for VM full-screen feature.

PF10 Previous

Displays the previous logical screen, but does no processing. For example, if the Add User Privileges screen is displayed and you press PF10, the Add Local Information screen is displayed.

PF11 Next

Displays the next screen, but does no processing. For example, if the Add User Privileges screen is displayed and you press PF11, the Add Subsystem Privileges screen is displayed.

PF12 Retrieve

Displays the last command executed in the command or option lines. You can repeatedly press PF12 to retrieve the last 12 commands.

PA Keys

There are also two Program Attention (PA) keys. Their functions follow:

PA1

Puts you into CP read mode.

PA2

Deletes blanks in the fields and lets you insert values.

How to Use the Primary Option Menu

To begin using the full-screen feature, enter the following from CMS:

```
ACFFS
```

The CA ACF2 for VM Primary Option Menu appears. It shows you the options you can use with the CA ACF2 for VM for VM full-screen feature. Enter the number of the function you want to perform on the option line to begin processing.

```
M9PA-0000     CA ACF2 for VM Primary Option Menu         CA ACF2 for VM
OPTION ===> _____________________________________________________________
                                                              TIME 17:11
      0 Change options or PF keys

      1 User Identification

      2 Data Access Control

      3 Resource Control

      4 Source, Shift, Scope, and Zone Maintenance (not available)

      5 System Options - Control VMO records (and SHOW commands)

      6 Audit Reports

PF1=Help     2=Print     3=Quit         4=Return     5=        6=
PF7=         8=          9=Director     10=          11=       12=Retrieve
```

OPTION ===>

Enter one of the following digits for the type of function you want to perform:

0 Change options or PF keys

Displays the current values of the PF key settings and lets you change them.

1 User Identification

Select this option if you want to process logonids (define a new user, change an existing logonid, and so on).

2 Data Access Control

Select this option to process access rules.

3 Resource Control

Select this option to process resource rule sets.

4 Source, Shift, Scope, and Zone Maintenance

Not available at this time; for future use.

5 System Options - Control VMO records (and SHOW commands)

Select this option to porcess Control VMO records. Under this setting you can add, display, change, and delete VMO records.

Also provided under this option is the ability to display the output of any of the SHOW commands, including SHOW ALL, in a scrollable window that includes the ability to locate text within the show output.

6 Audit Reports

Select this option to run CA ACF2 for VM reports.

When you select an option from this menu, CA ACF2 for VM displays the panel you use to maintain logonids or rules. We describe these panels later in this guide.

For more information about using report screens see the *Reports and Utilities Guide*.

Changing Your PF Keys

To change your PF keys, select option 0, Change options or PF Keys, from the CA ACF2 for VM Primary Option Menu. CA ACF2 for VM displays the following screen:

```
M9PA-0010     Change Options or PFkeys                    CA ACF2 for VM
OPTION ===> ____________________________________________________________
                                                            TIME 17:12
 Options:
  Yes ==> _        Time (12/24) ==> _
  No  ==> _        Language     ==> _____

                Description   Command
  PF1  ==>      Help          HELP
  PF2  ==>      Print         PRINT
  PF3  ==>      Quit          QUIT
  PF4  ==>      Return        RETURN
  PF5  ==>      __________    __________
  PF6  ==>      __________    __________
  PF7  ==>      Backward      BACKWARD
  PF8  ==>      Forward       FORWARD
  PF9  ==>      Director      DIRECTOR
  PF10 ==>      Save          SAVE
  PF11 ==>      __________    __________
  PF12 ==>      Retrieve      RETRIEVE


PF1=Help      2=Print      3=Quit         4=Return      5=         6=
PF7=          8=           9=Director     10=           11=        12=Retrieve
```

Yes

Indicate how you want yes values displayed. Enter a 1 (binary) or Y (alpha) in this field.

No

Indicate how you want no values displayed. Enter 0 (binary) or N (alpha) in this field.

Time(12/24)

Indicate whether you want the time reported in a 12-hour (enter 12) or 24-hour (enter 24) clock.

Language

Indicate the five-character code for the language you want to use for screens. We supply the following valid options:

AMENG

American English

UCENG

Uppercase English.

Your site may have added additional language support. Contact your security administrator for this information.

To change the values for the PF keys, enter the new value to be displayed on the screen for the PF key in the Description column. Enter the actual command names (in caps) in the Command column.

Chapter 4: Using the ACF Command

As you learned in the "Getting Started" chapter, there are three ways to maintain logonids and rules. The previous chapter explained using the full-screen feature with CA ACF2 for VM. This chapter shows you how to use the ACF command and its subcommands to perform the same functions. With the ACF command, you enter subcommands that tell CA ACF2 for VM to perform functions instead of filling in fields on screens.

When you finish this chapter, you will know:

- How to activate the ACF command
- How to get help in an ACF command session
- How to get message help
- How to set the correct mode for various ACF processing
- What the various ACF subcommands do
- How to end an ACF session
- How to display various system settings.

This section contains the following topics:

Activating the ACF Command

You can enter the ACF command from CMS or from an exec. The screen below illustrates how to issue the ACF command from CMS:

```
R; T=1.23/2.34 12:00:00
acf

ACF
```

The system responds with ACF, indicating that you can use ACF subcommands to maintain logonids and rules.

Getting Help Under the ACF Command

When CA ACF2 for VM is active, you can issue the HELP subcommand to get help at any time. The syntax for this subcommand is:

```
         [ subcommand    ]
HELP     [ info-request  ]  [options]
         [ MENU          ]
```

We describe operands for the HELP subcommand below:

Subcommand

Displays the help file for the ACF subcommand you specified.

info-request

Requests help for a function of the ACF command. For example, entering HELP ACFRULES displays information about the syntax of a rule record. Entering HELP FIELDS displays information on logonid record fields used when updating or adding a logonid record. Valid CA ACF2 for VM info-requests are:

ACFRULES

Setting-dependent

FIELDS

LID or ACF settings

RECORDS

Infostorage settings

MODELS

Model setting

APPLIDS

Any setting

MODES

Displays valid command settings

RULES

Any setting (displays menu of rules and fields).

MENU

Displays the master ACF help menu. To display the menu for the current mode of the ACF command, enter the following:

HELP

There are no operands for this command.

Options

Specifies the same options as CMS help. CA ACF2 for VM passes these options to CMS help. Valid options are:

```
{DESCript }  {FORMat  } { PARMs  }
{FUnction }  {SYntax  } { OPERands}

{ERRors  }   {OPTions}   {NOTes}
{Message }
```

We describe these CMS HELP options below:

DESCript

Displays a general description of the specified HELP file.

FORMat

Displays the format section (the syntax) of the command you requested HELP for.

PARMs

Displays the parameter section (the operands) for the command you requested HELP for.

ERRors

Displays the error message and response sections.

OPTions

Displays the options section (a list of available options with a brief description).

NOTes

Displays the notes and example sections.

As you can see, many options that appear on the previous page are not described here. That is because they are OS/390 equivalents for the CMS HELP option. The table below shows which OS/390 HELP options have CMS equivalents:

OS/390 HELP Option	Equivalent CMS HELP Option
FUnction	DESCript
SYntax	FORMat
OPERands	PARMs
Message	ERRors

You can enter four options (DESCript, FORMat, PARMs, and ERRors) with the OS/390 ACF HELP subcommand. They are translated to the equivalent CMS HELP option before CMS HELP is invoked.

Getting Message Help

To get help with CA ACF2 for VM messages, use the following syntax:

```
       [ Message msgid ]
HELP   [ MSG msgid     ]
       [ msgid         ]
```

We describe valid operands for getting help with CA ACF2 for VM messages through the HELP command below.

Message msgid|MSG msgid|msgid

CA ACF2 for VM displays information on the specified message. The value of msgid can be any IBM or CA ACF2 for VM message number that has help available.

ACF SET Subcommand

After entering the ACF command, you must establish an ACF setting that tells CA ACF2 for VM what type of processing you want to do. Issue the ACF SET subcommand and an operand to establish the setting.

The SET subcommand determines the particular type of ACF record or rule set you can process through the ACF subcommands.

It also determines the type of display you see when you use the TRIVIA|NOTRIVIA (controls the fields displayed) and TERSE|VERBOSE (controls how much of a logonid record is displayed) operands. Through the MODE|NOMODE operands, the SET subcommand determines if certain ACF command responses appear as a question mark (?) or the name of the setting. Enter the SET subcommand with one of the following operands:

```
    { Acf                       }
    { applidclass(applidtype)   }
    { CMdlim                    }
    { Control(Vmo|ACFServe)     }
    { Diaglim                   }
    { Entry(type)               }
    { Force|NOForce             }
    { Lid                       }
SEt { MODE|NOMode               }
    { MODEL                     }
    { Resource(type)            }
    { Rule                      }
    { SCope(SCP)                }
    { SHift(SFT|ZON)            }
    { SYSid(sysid)              }
    { Terse|Verbose             }
    { TRivia|NOTrivia           }
```

When you issue the SET subcommand, you must use at least one of the above operands. When you enter the ACF command without entering the SET subcommand, the following default values are in effect: ACF, FORCE, MODE, VERBOSE, and TRIVIA.

For example, to establish the RULE setting to process access rule sets, enter:

```
acf
ACF
set rule
RULE
```

The system responds with **RULE** to tell you that you can use the RULE setting.

For some settings, you must be more specific about the type of CA ACF2 for VM record you want to process. For example, to process entry records, you must specify the setting and a three-character type code:

```
acf
ACF
set entry(src)
ENTRY(SRC)
```

CA ACF2 for VM responds with **ENTRY SRC**, indicating that you can process source entry records.

All of the valid operands you can use with the ACF SET subcommand are described below.

Acf

The default setting. You are in this setting when you enter ACF. Use this setting to process logonid records and access rule sets. It is a combination of the LID and RULE settings, that are described below.

applidclass(applidtype)

Specifies that you will be updating structured infostorage records.
CA ACF2 for VM supports and uses the CONTROL(VMO) and CONTROL(ACFSERVE) settings. You can define additional applications through the VMO APPLDEF record.

Cmdlim

Specifies that you want to limit access to CP commands.

Control(Vmo|ACFServe)

Specifies that you will be processing VMO records or ACFSERVE privilege records. (You can also use SER as an abbreviation for ACFSERVE.)

Diaglim

Specifies that you want to limit the authority to issue diagnose instructions.

Entry(type)

Specifies that you want to process entry records. Entry records specify which terminal a user must use to access the system. The parameter code represents the type of entry record to be processed. Valid types of entry records include:

SRC

Indicates source entry records

SGP

Indicates source group entry record

usr

Indicates a site-defined entry record.

Force|NOForce

Specifies if you want to force storing of the rule set on the database. During the STORE process, the FORCE setting (the default) replaces the rule set if it already existed on the database. The NOFORCE setting stores rules only if they did not exist on the database before.

Lid

Specifies that you want to process logonids.

MODE|NOMode

Sets how CA ACF2 for VM prompts you for input. The MODE setting (the default) prompts with the current record setting. For example, if you issued the SET RULE command, CA ACF2 for VM displays RULE every time you press **Enter** to remind you what setting you are in. The NOMODE setting issues a question mark (?) to indicate that CA ACF2 for VM is active and waiting for a command or response.

MODEL

Indicates that you want to process CP command syntax models.

Resource(type)

Specifies that you want to process resource rules. The type represents the resource type to process. CA ACF2 for VM-supplied type codes are ACT, ALG, DIA, GRP, IUC, and VMC. You can also define your own resources to protect.

Rule

Specifies that you want to process access rule sets.

SCope(SCP)

Specifies that you want to define scope records. These records can limit the authority given to powerful users, such as security administrators.

SHift(SFT|ZON)

Specifies that you want to process SHIFT (SFT) and ZONE (ZON) records. These records define when a user can successfully access the system.

SYSid(sysid)

Sets the session default SYSID for the CONTROL(VMO) and CONTROL(ACFSERVE) settings.

Terse|Verbose

Using TERSE displays a condensed version of any CA ACF2 for VM record, for example, a logonid record or rule set. The fields of the @HEADER macro of the FDR determine the TERSE output. See the @HEADER macro description in the *Installation Guide*. The VERBOSE (default) setting displays the complete record.

The following examples illustrate the TERSE and VERBOSE operands of the SET subcommand.

Ann Smith's logonid is listed under the TERSE setting:

```
TLCAMS     TLCTNYTLCAMS ANN SMITH     EXT 321
```

The same logonid listed under the VERBOSE setting looks like this:

```
TLCAMS          TLCTNYTLCAMS ANN SMITH  EXT 321
PRIVILEGES      AUDIT
ACCESS          ACC-CNT(223) ACC-DAT(02/21/98)
                ACC-SRCE(GRAF-247) ACC-TIME(08:32)
PASSWORD        PSWD-DAT(02/23/98) PSWD-TOD(01/09/98-07:34)
                PSWD-VIO(4)
STATISTICS      UPD-TOD(03/19/97-06:57)
RESTRICTIONS    PREFIX(TLCAMS)
```

TRivia|NOTrivia

TRIVIA (the default) displays all fields of a logonid record or structured infostorage record that a user is authorized to display. NOTRIVIA does not display those fields that have the LIMIT flag specified in the @CFDE macro.

The screen below shows Ann Smith's logonid listed under the NOTRIVIA setting:

```
TLCAMS          TLCTNYTLCAMS ANN SMITH EXT 321
PRIVILEGES      AUDIT
RESTRICTIONS    PREFIX(TLCAMS)
```

The same logonid listed under the TRIVIA setting looks like this:

```
TLCAMS          TLCTNYTLCAMS ANN SMITH EXT 321
PRIVILEGES      AUDIT
ACCESS          ACC-CNT(223) ACC-DAT(02/21/98)
                ACC-SRCE(GRAF-247) ACC-TIME(08:32)
PASSWORD        PSWD-DAT(02/23/98) PSWD-TOD(01/09/98-07:34)
                PSWD-VIO(4)
STATISTICS      UPD-TOD(03/19/98-06:57)
RESTRICTIONS    PREFIX(TLCAMS)
```

Most of the remaining chapters in this guide explain how to use the ACF subcommands for processing the different kinds of CA ACF2 for VM records. The chart below can help you locate information contained in this guide and direct you to other guides, if necessary.

Setting	Use	Source
ACF	Logonid records	
	Access rules	
CMDLIM	Command limiting rules	*Command and Diagnose Limiting Guide.*
CONTROL	System option records	

Setting	Use	Source
DIAGLIM	Diagnose limiting rules	*Command and Diagnose Limiting Guide.*
ENTRY	Entry records	
FORCE|NOFORCE	Session setting	
LID	Logonid records	
MODE|NOMODE	Session setting	
MODEL	Syntax models	*Command and Diagnose Limiting Guide.*
RESOURCE	Resource rules	
RULE	Access rules	
SCOPE	Scope records	
SHIFT	Shift|Zone records	
TERSE|VERBOSE	Session setting	
TRIVIA|NOTRIVIA	Logonid field display	

ACF Subcommands

After you enter the ACF command and establish the appropriate command setting, you can issue ACF subcommands. These subcommands let you do the actual processing of CA ACF2 for VM records and rule sets with the setting you established. The ACF subcommands are shown below.

You can abbreviate these subcommands to characters that uniquely identify each command. Uppercase letters indicate the minimum abbreviation for each subcommand.

ACFServe
Change
CMS
Compile
CP
DEComp
DELete
End
EXEc
Help
Insert
List
Quit
SEt (or seT)
Show
Store
Test
XEdit

Some subcommands are not valid in all settings. The END, QUIT, HELP, SET, and SHOW subcommands are common to all settings. The functions of all the ACF subcommands are described below.

ACFServe

This subcommand and its ACFSERVE command operands are passed to CA ACF2 for VM CP code. Refer to for more information.

Change

Under the LID setting, this subcommand modifies the fields of the logonid record; under the SCOPE setting, it modifies scope records; under the SHIFT setting, it modifies shift records, and under the ENTRY setting, it modifies entry lists defining input sources and input source groups.

CMS

Specifies that when you enter a CMS command on the command line, the CMS command is executed. After the CMS command completes processing, CA ACF2 for VM returns you to the ACF command mode you were in when you issued the CMS command. This subcommand lets you perform ACF command and CMS command functions in the same session.

Compile

Under the ACF, RULE, RESOURCE, CMDLIM, and DIAGLIM settings, this subcommand builds the specified rule sets.

CP

Issues a command to CP. This subcommand recognizes CP commands.

DEComp

Under the ACF, RULE, CMDLIM, DIAGLIM, RESOURCE, and MODEL setting, this subcommand decompiles a rule set that you had previously compiled or stored, and displays it on the screen.

DELete

Deletes individual records. In most CA ACF2 for VM settings, you can also delete several records.

End (or Quit)

Terminates the ACF command session and returns you to the CMS environment where you issued the ACF command.

EXEc

Runs an exec in an ACF command setting.

Help

Accesses the CA ACF2 for VM help facility. Provides information on CA ACF2 for VM and ACF subcommands. You can enter HELP to access the help facility directly from CMS or any ACF setting.

Insert

Adds new records to the Logonid and Infostorage databases. You can use an existing logonid record as a model to add new logonid records. You can add entry records, scope records, and shift and zone records with this subcommand. This subcommand is valid in all ACF settings except RULE, CMDLIM, DIAGLIM, and RESOURCE.

List

Displays individual records or a range of records in the CA ACF2 for VM databases. When processing rules, LIST is a synonym for the DECOMP subcommand.

SEt (or T)

Tells CA ACF2 for VM what type of ACF command processing you want to do. You must always issue an operand following this subcommand.

Show

Displays information about your system. CA ACF2 for VM only displays information that you have the authority to see. This chapter contains more information on the SHOW subcommand on subsequent pages.

STore

Stores a compiled rule set. This subcommand lets you store compiled access rules, command and diagnose limiting rules, and resource rules. It is valid under the ACF, RULE, CMDLIM, DIAGLIM, and RESOURCE settings.

TEst

Verifies how CA ACF2 for VM interprets rules. This subcommand is valid under the ACF, RULE, CMDLIM, DIAGLIM, and RESOURCE settings.

XEdit

Issues the CMS XEDIT command to let you create new files and perform basic editing tasks.

Additional factors, besides the ACF command setting that you establish, determine what ACF subcommands you can use and under what conditions you can use them. They are:

- Your logonid privileges
- Your site's system options
- The scope of your authority.

Displaying CA ACF2 for VM System Settings

The SHOW subcommand displays information about your CA ACF2 for VM system. There is no default. The type of information displayed depends on the level of authority you have. Enter the SHOW subcommand and one operand as follows:

```
     { ACF2            }
     { ACFServe        }
     { ACTive          }
     { ALL             }
     { APpldef         }
     { Backup          }
     { CLASMAP         }
     { CMdlim          }
     { CMSsec          }
     { DATE            }
     { Ddsns           }
     { DIaglim         }
     { Fields          }
     { Fields(recname) }
     { Linux<linuxopts>}
SHow { MAINT           }
     { MODE            }
     { NOAuto          }
     { NOSort          }
     { Programs        }
     { RESWord         }
     { SAFDEF          }
     { SECsys          }
     { SMF             }
     { SRF             }
     { STate           }
     { SYstems         }
     { UID             }
     { Zerofld s       }
```

Listed below are all the possible system settings for VM environments.

ACF2

Displays the same information as SHOW ALL.

ACFServe

Displays the fields of ACFSERVE privilege records.

ACTive

Displays the status of the CA ACF2 for VM system intercepts and exits. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

ALL

Displays the equivalent of entering all the SHOW commands. This operand displays all the information about the system you are authorized to see. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see all of this information.) This displays the same information as SHOW ACF2.

APpldef

Displays information about site-defined structured infostorage records.

Backup

Displays the backup parameters in effect at the time. This information includes when backups are taken, the filenames of the backup copies, and the address of the minidisk where the backup copy will be located. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

CMdlim

Displays information about the CP command limiting parameters in effect. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

CLASMAP

Displays information about internal and external CLASMAPS.

CMSsec

Indicates whether CMS file protection is installed. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

DATE

Displays the date format used in CA ACF2 for VM. This information is available to all users.

Ddsns

Displays which databases are currently in use. Also, it also displays the names of all CMS, VSE/VSAM, and backup files defined to CA ACF2 for VM. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

Diaglim

Displays information about the diagnose limiting parameters in effect. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

Fields

Displays the logonid field names that you have authority to display or change. This information is available to all users.

Fields(recname)

Displays fields for a specified record under a structured infostorage setting, such as CONTROL(VMO), or CONTROL(ACFSERVE). You must be able to list the record.

Linux

Displays Linux machine definitions.

Linux<All|Machine|Users|Users(*nn-mm*)||Groups|Groups(*nn-mm*)|Duplicates>

Displays Linux definitions according to the requested selection criteria, as defined below. You can specify multiple operands, for example:

```
SHOW LINUX MACHINE USERS

SHOW LINUX USERS GROUPS(1-100)

SHOW LINUX USERS(1000-2000) DUPLICATES
```

ALL

Displays all Linux definitions. If ALL is specified, the only other operand that can be used is DUPLICATES.

Machine

Displays Linux Machine definitions. This is the default if no other operands are specified.

Users

Displays Linux User definitions.

Users(*nn-mm*)

Displays Linux User definitions with a UID value that falls in the range specified by (nn-mm).

Users(*nn*)

A single value may also be supplied to only display Linux User definitions that match the single value.

Groups

Displays Linux Group definitions.

Groups(*nn-mm*)

Displays Linux Group definitions with a GID value that falls in the range specified by (nn-mm).

Groups(nn)

A single value may also be supplied to only display Linux Group definitions that match the single value.

Duplicates

Displays only the Linux User and Group definitions that have duplicate UID or GID values. This option can only be used with USERS and/or GROUPS. When it is specified, Machine definitions are not displayed. The MACHINE operand can not be specified with this option.

MAINT

Displays the logonids that are defined as maintenance logonids. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

MODE

Displays the current ACF command record setting. This information is available to all users.

NOAuto

Tells you if FORCEID users must supply their VM directory password before logging on if the CA ACF2 for VM service machine is unavailable. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

NOSort

Indicates the availability of $NOSORT control statements in CA ACF2 for VM rules. This information is available to all users.

Programs

Displays the same information as SHOW MAINT.

RESWord

Displays the current password reserved word prefixes.

SAFDEF

Displays information about internal and external SAFDEFS.

SECsys

Displays the shared systems that bypass diagnose limiting validation. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

SMF

Displays combined record numbers (if you specify this option in the @SMF macro of the ACFFDR) and various configuration options used for SMF recording. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

SRF

Indicates which guest machines are potentially authorized to use the System Request Facility (SRF). This machine also needs the SRF logonid privilege to be fully authorized. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

STate

Displays access rule options, password options, password phrase options, UID definition, resource types and classes, and data set rule interpretation parameters currently in effect. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

SYstem

Displays various SRVMOPTS parameters, command limiting parameters, IUCV, APPC/VM, and VMCF validation parameters, FORCEID, AUTOLOG, and NOAUTOU lists, the National Language support parameters, and the CMS file ID translate characters currently in effect. Some of these parameters are SMF-related. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

UID

Displays the user identification string definition used at your site. This information is available to all users.

Zeroflds

Displays the fields of the logonid record that are not copied to another logonid through the INSERT USING subcommand. (Only users with the ACCOUNT, AUDIT, or SECURITY privilege see this information.)

The values displayed when you issue the SHOW command are site-dependent and vary based on the options that are active at your site. Samples of all the SHOW subcommands are shown in the following sections.

Displaying ACFSERVE Command Privileges

To display the fields of the ACFSERVE privilege records currently in effect for your system, enter the following command:

```
SHOW ACFSERVE
```

```
ACF

show acfserve
 -- ACFSERVE COMMAND PRIVILEGES IN EFFECT --

 ACFSERVE RECORD ID                PRIVILEGES
```

```
ARCHIVE.SMF                 AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(ACCOUNT SECURITY)
BACKUP                      AUTLOGIC(AND) LOG   SCOPED
                            AUTH(SECURITY)
CKPT.DATABASE               AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(AUDIT SECURITY)
CKPT.SMF                    AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(AUDIT SECURITY)
DELETE.RESOURCE             AUTLOGIC(AND) LOG   SCOPED
                            AUTH(ACCOUNT SECURITY)
DISABLE.SYNC                AUTLOGIC(AND) LOG   NOSCOPED
                            AUTH(ACCOUNT SECURITY)
ENABLE.SYNC                 AUTLOGIC(AND) LOG   NOSCOPED
                            AUTH(ACCOUNT SECURITY)
QUERY.BATCH                 AUTLOGIC(AND) NOLOG SCOPED
                            AUTH()
QUERY.DATABASE              AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(AUDIT SECURITY)
QUERY.DDSN                  AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(AUDIT SECURITY)
QUERY.GROUPID               AUTLOGIC(AND) NOLOG SCOPED
                            AUTH()
QUERY.SECTRACE              AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(AUDIT SECURITY)
QUERY.SMF                   AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(AUDIT SECURITY)
QUERY.SOURCE                AUTLOGIC(AND) NOLOG SCOPED
                            AUTH()
QUERY.STATUS                AUTLOGIC(AND) NOLOG SCOPED
                            AUTH()
QUERY.SYNC                  AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(AUDIT SECURITY)
RELOAD.COMMAND              AUTLOGIC(AND) LOG   SCOPED
                            AUTH(SECURITY)
RELOAD.CONTROL.ACFSERVE     AUTLOGIC(AND) LOG   NOSCOPED
                            AUTH(SECURITY)
RELOAD.CONTROL.VMO          AUTLOGIC(AND) LOG   NOSCOPED
                            AUTH(SECURITY)
RELOAD.DIAGNOSE             AUTLOGIC(AND) LOG   SCOPED
                            AUTH(SECURITY)
RELOAD.FDR                  AUTLOGIC(AND) LOG   NOSCOPED
                            AUTH(ACCOUNT SECURITY)
RELOAD.LMP                  AUTLOGIC(AND) LOG   NOSCOPED
                            AUTH(ACCOUNT SECURITY)
RELOAD.PROFILE              AUTLOGIC(AND) LOG   SCOPED
                            AUTH(SECURITY)
RELOAD.RESOURCE             AUTLOGIC(AND) LOG   SCOPED
                            AUTH(ACCOUNT SECURITY)
RELOAD.RULE                 AUTLOGIC(AND) LOG   SCOPED
                            AUTH(SECURITY)
RELOAD.SHIFT                AUTLOGIC(AND) LOG   SCOPED
                            AUTH(SECURITY)
RELOAD.XREF                 AUTLOGIC(AND) LOG   SCOPED
                            AUTH(SECURITY)
RESET                       AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(ACCOUNT SECURITY)
RESTART                     AUTLOGIC(AND) LOG   NOSCOPED
                            AUTH(SECURITY)
SECTRACE.DELETE             AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(ACCOUNT SECURITY)
SECTRACE.DISABLE            AUTLOGIC(OR)  LOG   SCOPED
                            AUTH(ACCOUNT SECURITY)
```

```
SECTRACE.ENABLE                 AUTLOGIC(OR)  LOG   SCOPED
                                AUTH(ACCOUNT SECURITY)
SECTRACE.MODIFY                 AUTLOGIC(OR)  LOG   SCOPED
                                AUTH(ACCOUNT SECURITY)
SECTRACE.SET                    AUTLOGIC(OR)  LOG   SCOPED
                                AUTH(ACCOUNT SECURITY)
SET.SYSID                       AUTLOGIC(AND) LOG   NOSCOPED
                                AUTH(SECURITY)
SWITCH.SMF                      AUTLOGIC(OR)  LOG   SCOPED
                                AUTH(AUDIT SECURITY)
```

Displaying All System Options

The SHOW ALL subcommand is the same as entering all the SHOW subcommands or the SHOW ACF2 subcommand.

```
SHOW ALL
```

All SHOW subcommand information displays.

Displaying Intercepts and Exits

To display CA ACF2 for VM intercepts that received control since the last IPL (indicated by YES), which optional intercepts are enabled, and the status of local exits, enter:

```
SHOW ACTIVE
```

```
ACF
show active

 -- ACF2 VM/XA CONTROL ROUTINES THAT HAVE BEEN INVOKED --
 *RPI CONNECT      (YES) *RPI QUIESCE     (NO)  *RPI RESUME       (NO)
 *RPI SEND         (NO)  *RPI SEVER       (NO)  ACI DIAGNOSE      (NO)
 ALTUSER DIAGNOSE  (NO)  AUTOLOG COMMAND  (NO)  COMMAND VALIDATN  (YES)
 CPFORMAT COMMAND  (NO)  DASD DETACH      (YES) DASD REDEFINE     (NO)
 DATA SPACE PERMT  (NO)  DIAGNOSE INSTRUC (YES) DISPLAY/DUMP CMD  (NO)
 DIRCTRY UPDATE    (NO)  IPL COMMAND      (YES) IUCV-APPC CONNCT  (YES)
 IUCV-APPC SEVER   (NO)  LDEV DESTRUCTION (NO)  LOGOFF PROCESS    (YES)
 LOGON COMMAND     (YES) LOGON VMDBK SWAP (YES) LOGON VERIFICATN  (YES)
 PASSWORD READS    (YES) PASSWORD VALIDAT (YES) REAL DASD ATTACH  (NO)
 SET ACCOUNT CMD   (NO)  SHUTDOWN PROCESS (NO)  STORE HOST CMD    (NO)
 TERMINAL DIAL     (NO)  TERMINAL DROP    (NO)  TRACE COMMAND     (NO)
 UNIT RECORD LOG   (NO)  UNIT RECORD STRT (NO)  VIRT DASD CREATE  (YES)
 VMCF INITIATION   (NO)  VMDUMP COMMAND   (NO)  XAUTOLOG COMMAND  (YES)

 -- ACF2 VM OPTIONAL INTERCEPT STATUS --
 LOGON TRAP            (DISABLED)   IUCV-APPC VALIDATION  (ENABLED)
 VMCF VALIDATION       (ENABLED)    UNIT RECORD LOGGING   (DISABLED)
 DIAG84 ACF2 PASSWORDS (ENABLED)    DIRECTORY IPL         (ENABLED)
 DATA SPACE VALIDATION (ENABLED)    VM ACCT VALIDATION    (DISABLED)

 -- LOCAL EXITS SPECIFIED ON THIS SYSTEM --
 ACT PRE-VALIDATE=  *NO EXIT*      ACT POST-VALIDATE= *NO EXIT*
 DSN PRE-VALIDATE=  *NO EXIT*      DSN POST-VALIDATE= *NO EXIT*
 DSN VIOLATION=     *NO EXIT*      LDEV INTERFACE=    *NO EXIT*
 LGN USER AUTH=     *NO EXIT*      LGN PRE-VALIDATE=  *NO EXIT*
 LGN POST-VALIDATE= *NO EXIT*      NEW PASSWORD=      *NO EXIT*
 USER ENCRYPTION=   *NO EXIT*      EXPIRED PASSWORD=  *NO EXIT*
```

Displaying Backup Information

To display information about when backups are taken and where the backup copies are located, enter:

```
SHOW BACKUP
```

```
ACF
show backup

-- BACKUP PARAMETERS IN EFFECT --

AUTO BACKUP TIME=(00:01)
      NOTIFY=(OPERATOR)
      DDSNID=BACKUP
      MDISK=195
      AUTOLOG=NONE
```

Displaying Resource Classes

To display internal (CA ACF2 for VM-defined) and external (site-defined) CLASMAP records, enter:

```
SHOW CLASMAP
```

```
ACF
```

```
show clasmap

 -- INTERNAL CLASMAP DEFINITIONS --
MUSASS     RESOURCE     TYPE  ENTITY
 ID        CLASS        CODE  LENGTH
======     ========     ====  ======
CICS       FILE         CFC     8
CICS       PROGRAM      CPC     8
CICS       TRANS        CKC     4
CICS       TRANDATA     CTD     8
CICS       TEMPSTRG     CTS     8
CICS       DL/I         CPB     8
-          PROGRAM      PGM     8
-          UNVRPRT      UNR     0
-          UNVPGM       UNP     8
-          ACAPPL       ACA     0
-          ACDIALOG     ACD     0
-          DIRECTRY     SAF   153
-          FILE         SAF   171
-          SFSCMD       SAF   171
 -- EXTERNAL CLASMAP DEFINITIONS --
MUSASS     RESOURCE     TYPE  ENTITY
 ID         CLASS       CODE  LENGTH
========   ========     ====  ======
********   TSTPROD1     TPI     0
```

Displaying Command Limiting Parameters

To display current command limiting parameters, enter:

```
SHOW CMDLIM
```

These parameters include the mode for command limiting validation, command limited CP commands, and the type of command limiting models that define the valid syntax for each of the CP commands.

```
ACF

show cmdlim
-- COMMAND LIMITING PARAMETERS IN EFFECT --

MODE = RULE, ABORT, ABORT

SYNERR = PREVENT     NOSPOOL = PREVENT

VALIDATION IS ACTIVE FOR ALL CP COMMANDS EXCEPT THE FOLLOWING:
ADSTOP          ATTN          BEGIN         CLOSE
COMMANDS        DCP           DISCONN       DISPLAY
DVIRT           DUMP          ECHO          EXTERNAL
INDICATE        LOCATE        LOCK          LOGOFF
LINK            NOTREADY      PER           QUERY
READY           REQUEST       RESET         REWIND
SCREEN          SEND          SLEEP         STORE
SYSTEM          TERMINAL      TRACE

MDL TYPE = ZVM
```

Displaying CMS Security Information

To see if CMS file protection is active at your site, enter:

```
SHOW CMSSEC
```

```
ACF

show cmssec
 -- CMS SECURITY PARAMETERS --

 CMS SECURITY IS ENABLED
```

Displaying Date Format

To display the date format currently in effect for your system, enter:

```
SHOW DATE
```

```
ACF

show date
DATE FORMAT=MM/DD/YY
```

Displaying CA ACF2 for VM Database Names

To display the CMS filenames for the Rule, Logonid, and Infostorage databases, enter:

```
SHOW DDSNS
```

SHOW DDSNS also lists all groups of databases that your site can use.

The group of databases used is specified at CA ACF2 for VM startup time. If no database startup option is specified, CA ACF2 for VM uses the first set of databases defined in the list. For additional information on how these groups are defined, refer to the @DDSN macro information in the *Installation Guide.*

```
ACF

show ddsns
 -- ACF2 DYNAMIC DATASET NAMES SPECIFIED --
 DDSN DEFAULTED AT STARTUP.  DSNS IN USE ARE:
         RULES   =VMVSAM.ACF.RULES
         LOGONIDS=VMVSAM.ACF.LIDS
         INFOSTG =VMVSAM.ACF.INFO

 DDSN LISTS DEFINED IN FDR ARE:
 PRIMARY RULES    =VMVSAM.ACF.RULES
          LOGONIDS=VMVSAM.ACF.LIDS
          INFOSTG =VMVSAM.ACF.INFO

 ALT      RULES   =RULES
          LOGONIDS=LIDS
          INFOSTG =INFO
```

Display Diagnose Instruction Information

To display the mode for diagnose limiting validation, diagnose codes that are diagnose limited, and the type of diagnose limiting models defining valid syntax for each code, enter:

```
SHOW DIAGLIM
```

```
ACF

show diaglim
 -- DIAGNOSE LIMITING PARAMETERS IN EFFECT --

 MODE = RULE, ABORT, ABORT

 VALIDATION IS ACTIVE FOR THE FOLLOWING DIAGNOSE CODES:
 ACF                  104

 MDL TYPE = z/VM
```

Displaying Linux Machine and User Definitions

To display the Linux Machine and User definitions:

```
ACF

show linux all
-- LINUX DEFINITIONS --


-- LINUX MACHINES --

 MACHINE NAME              IP ADDRESS                              ACTIVE
 ------------------------  --------------------------------------  ------
 Linuxtest0001             1.2.3.4                                 YES

 Linuxtest0002             1.2.3.5                                 YES


 -- LINUX USERS FOR MACHINE TEST0001 --

  UID             NAME       GROUP      APPLICATION NAME
  -------------   --------   --------   ---------------------------------------
  111,111,111     USER01     TESTGRP1   longname4user01



 -- LINUX USERS FOR MACHINE TEST0002 --

  UID             NAME       GROUP      APPLICATION NAME
  -------------   --------   --------   ---------------------------------------
  222,222,222     USER02     TESTGRP2   longname4user02



 -- LINUX USERS (NO SPECIFIC MACHINE) --

  UID             NAME       GROUP      APPLICATION NAME
  -------------   --------   --------   ---------------------------------------
  201             VMTEST01   TESTGRP3   vmtest01

  205             VMTEST05   TESTGRP4   vmtest05

 -- LINUX GROUPS --

  GID             NAME
  -------------   --------
  101             TESTGRP1
  102             TESTGRP2
  103             TESTGRP3
  104             TESTGRP4
  2,123,123,123   BIGGROUP
```

Displaying Logonid Fields

To display all the logonid field names that you can display or modify, enter:

```
SHOW FIELDS
```

An asterisk (*) next to a field name indicates that you can modify that field.

```
ACF

show fields
 -- IDENTIFICATION --
  LID        *NAME       *PASSWORD  *PHONE       UID
 -- CANCEL/SUSPEND --

 *CANCEL     CSDATE      CSWHO      *MON-LOG   *MONITOR   *PP-TRC    *PP-TRCV
 *SUSPEND    *TRACE      *TSO-TRC

 -- PRIVILEGES --
```

```
*ACCOUNT   *ACTIVE    *AUDIT     *AUTOALL   *AUTODUMP  *AUTONOPW  *AUTOONLY
*BDT       *CICS      *CMD-PROP  *CONSULT   *DG84DIR   *DIALBYP   *DSNSCOPE
*DUMPAUTH  *EXPIRE    *GRP-OPT   *GRPLOGON  *IDMS      *IMS       *JOB
*JOBFROM   *LDEV      *LDS       *LEADER    *LIDSCOPE  *LOGSHIFT  *MAINT
*MUSASS    *NO-INH    *NO-OMVS   *NO-SMC    *NO-STORE  *NOMAXVIO  *NON-CNCL
*NOSPOOL   *PGM       *PPGM      *PRIV-CTL  *PROGRAM   *READALL   *REFRESH
*RESTRICT  *RSRCVLD   *RULEVLD   *SCPLIST   *SECURITY  *SRF       *STC
*SUBAUTH   *SYNCNODE  *SYNERR    *SYSPEXCL  *TAPE-BLP  *TAPE-LBL  *TDISKVLD
*TSO       *UIDSCOPE  *USER      *VAX       *VLDRSTCT  *VLDVMACT  *VM
*VMD4AUTH  *VMD4FSEC  *VMD4RSET  *VMD4TARG  *VMESM     *VMSAF     *VMSFS
*VMXA      *VSESRF
-- ACCESS --
 ACC-CNT    ACC-DATE   ACC-SRCE   ACC-TIME   GRP-USER
-- PASSWORD --
*KERB-VIO   KERBCURV   LIDTEMP   *LIDZMAX   *LIDZMIN   *MAXDAYS   *MINDAYS
*PSWD-DAT  *PSWD-EXP  *PSWD-INV   PSWD-MIX   PSWD-SRC   PSWD-TIM   PSWD-TOD
*PSWD-VIO  *PSWD-XTR
-- TSO --
*ACCTPRIV  *ALLCMDS   *ATTR2     *CHAR      *CMD-LONG  *CONSOLE   *DFT-DEST
*DFT-PFX   *DFT-SOUT  *DFT-SUBC  *DFT-SUBH  *DFT-SUBM  *INTERCOM  *JCL
*LGN-ACCT  *LGN-DEST  *LGN-MSG   *LGN-PERF  *LGN-PROC  *LGN-RCVR  *LGN-SIZE
*LGN-TIME  *LGN-UNIT  *LINE      *MAIL      *MODE      *MOUNT     *MSGID
*NOTICES   *OPERATOR  *PAUSE     *PMT-ACCT  *PMT-PROC  *PROMPT    *RECOVER
*TSOACCT   *TSOCMDS   *TSOFSCRN  *TSOPERF   *TSOPROC   *TSORBA    *TSORGN
*TSOSIZE   *TSOTIME   *TSOUNIT   *UNICNTR   *VLD-ACCT  *VLD-PROC  *WTP
-- STATISTICS --
*SEC-VIO    UPD-TOD
-- CICS --
*ACF2CICS  *CICSCL    *CICSID    *CICSKEY   *CICSKEYX  *CICSOPT   *CICSPRI
*CICSRSL   *IDLE      *MULTSIGN
-- IMS --

-- IDMS --
*IDMSPROF  *IDMSPRVS
-- MUSASS --
*MUSDLID   *MUSID     *MUSIDINF  *MUSOPT    *MUSPGM    *MUSUPDT   *NO-STATS
-- RESTRICTIONS --
*AUTHSUP1  *AUTHSUP2  *AUTHSUP3  *AUTHSUP4  *AUTHSUP5  *AUTHSUP6  *AUTHSUP7
*AUTHSUP8  *GROUP     *PREFIX    *SHIFT     *SOURCE    *VMACCT    *VMIDLEMN
*VMIDLEOP  *ZONE
-- DFP --
*SMSINFO
```

Many of the fields displayed above do not apply to CA ACF2 for VM for VM. They are supported in VM to maintain compatibility with other operating systems when used in a database synchronization environment.

Displaying Structured Infostorage Record Fields

To display the names of the fields in a structured infostorage record, enter:

```
SHOW FIELDS (recname)
```

The following example shows how you can display fields in a specified VMO record. An asterisk (*) indicates single-valued fields that you can change. A plus sign (+) indicates multi-valued fields you can change.

```
ACF
SET CONTROL(VMO)
SHOW FIELDS(BACKUP)

--BACKUP--
*MDISK  *TIME
```

Displaying Maintenance IDs

To display maintenance logonids that bypass normal CA ACF2 for VM data access validations, enter:

```
SHOW MAINT

ACF
show maint
-- NO MAINTENANCE LOGONIDS --
```

Displaying CA ACF2 for VM Mode

To see the record setting of the ACF command, enter:

```
SHOW MODE
```

```
ACF

show mode
 MODE: ACF
```

Displaying Validation When CA ACF2 for VM Is Not Active

To display how validation is being handled when the CA ACF2 for VM service machine is unavailable, enter:

```
SHOW NOAUTO
```

If NOAUTO=NOPASS is set, passwords are not required by FORCEIDs.

```
ACF

show noauto
 -- NOAUTO FORCEID LOGON MODE --

 NO LOGON PASSWORDS ARE REQUIRED
```

If you specify NOAUTO=DIRPASS, FORCEID users must enter their valid VM directory password while logging on.

```
ACF

show noauto
 -- NOAUTO FORCEID LOGON MODE --

 DIRECTORY LOGON PASSWORDS ARE REQUIRED
```

Displaying Rule Sorting Default

To determine if $NOSORT control statements are supported in a specific rule set, enter:

```
SHOW NOSORT
```

```
ACF

show nosort
NOSORT=NO
```

Displaying Maintenance IDs

To display logonids that bypass normal CA ACF2 for VM data access validation, enter:

```
SHOW PROGRAMS
```

```
ACF

show programs
 -- MAINTENANCE LOGONIDS --
 $ARCHIVE
 $ASMBMON
 $BACKUP
 $DASDMNT
 $DEFRAG
 $MIGRATE
```

Displaying Password Reserved Word Prefixes

```
ACF

show resword
-- RESERVED WORD PREFIX LIST --
 APPL     APR      ASDF     AUG      BASIC    CADAM    DEC      DEMO
 FEB      FOCUS    GAME     IBM      JAN      JUL      JUN      LOG
 MAR      MAY      NET      NEW      NOV      OCT      PASS     ROS
 SEP      SIGN     SYS      TEST     TSO      VALID    VTAM     XXX
 1234
```

Displaying SAFDEF Record Information

To display the values in a SAFDEF record, enter:

```
SHOW SAFDEF
```

```
ACF

show safdef
SFSFILEA  JOBNAME=********  USERID=********   PROGRAM=********   RB=********
           RETCODE=0        SAFDEF=INTERNAL   MODE=IGNORE        SUBSYS=ACF2
```

Displaying Bypassed Diagnose Validation

To display the shared systems that bypass diagnose validation, enter:

```
SHOW SECSYS
```

You can receive four possible responses (we illustrate one below). This option can improve the performance of these systems.

```
ACF

show secsys
 -- SECURE SHARED SYSTEM PARAMETERS --

 ALL SHARED SYSTEMS BYPASSING DIAGNOSE LIMITING
```

Displaying CA ACF2 for VM SMF Options

To display event record numbers for the various SMF record types and the SMF processing options in effect, enter:

```
SHOW SMF
```

CA ACF2 for VM displays both precombined and combined record entries.

```
ACF

show smf

-- SMF PARAMETERS IN EFFECT --

 ACF2 COMBINED SMF RECORD = 230

 PRE-COMBINED SMF RECORD NUMBERS:
 PASSWORD=220            DATASET VIO=221       LID JOURNAL=222
 RULE JOURNAL=223        INFO JOURNAL=226      RESOURCE VIO=227
 ACFSERVE COMMAND=229    CMDLIM/DIAGLIM=231    DirMaint JOURNAL=232

 ACF2 SMF CONTROL:
 DUMP DESTINATION ID          = ISODUMP  SWITCH ACTION          = DUMP
 AUTOMATIC SWITCH TIME        = 00:00

 ACF2 SMF CPU IDENTIFIER      = 4381

 SMF MINIDISK(S)              = 201, 202, 203

 CHECKPOINT RECORD COUNT      = 020    CHECKPOINT TIMER INTERVAL = 015 MIN
 DUMP PUNCH INTERVAL          = 001    MAX QUEUED MSF WRITES     = 010

 SMF I/O IS ASYNCHRONOUS (PRIORITY=NO)
```

Displaying SRF Users

To display the potential users who can use the System Request Facility on behalf of other users, enter:

```
SHOW SRF
```

The display output shows which users are authorized and the mode that the user is in. This machine also needs the SRF privilege to be fully authorized.

```
ACF

show srf
 --- SHOW SRF ENTRIES ---

 VMUSER ID OF SRF = TLCMTM
 ----------------
 MODE = ABORT     NRULE = NONE   NMODE = NONE

 VMUSER ID OF SRF = TLCEMX
 ----------------
 MODE = WARN      NRULE = NONE   NMODE = NONE
```

Displaying System Options

To display information about the CA ACF2 for VM databases, CA ACF2 for VM mode, current ACFFDR, access rule and password options, UID, who can decompile rules, tape volume pseudo data set name and protected tape volumes, resource rule cache and types, resident type lists, SSFTYPE definitions, and data set rule options, enter:

```
SHOW STATE
```

```
ACF

show state
 RUNNING CA ACF2 for VM VM r8.0;  WITH MODE = RULE, LOG, WARN

 USING FDR ASSEMBLY:  11.30  04/18/05

 OPTIONS IN EFFECT:
 CONTROL=DECENTRALIZED      %CHANGE=ALLOWED            CPUTIME=LOCAL
 DATE FORMAT=MM/DD/YY       MAX VIO PER SESSION=10     NOSORT=NO
 VMCHK=**NULL**             MAXLID=512                 NOTIFY=YES
 IDLEOP=OFF                 IDLEMN=0                   RULELONG=ENABLED

 PASSWORD OPTIONS IN EFFECT:
 LOGON RETRY COUNT=1        MIN PSWD LENGTH=1          MAX PSWD ATTEMPTS=2
 PSWD ALTER=YES             PSWD FORCE=YES             PSWD WARN DAYS=1
 PSWD HISTORY=NO            PSWD RESERVE WORD=NO       PSWD EQ LID=ALLOW
 PSWD ALL NUMERIC=ALLOW     PSWD CMD CHANGE=ALLOW      PSWD VOWEL CHAR=ALLOW
 REQ ALPHABET CHAR=NO       REQ NUMERIC CHAR=NO        REQ NAT/USR CHAR=NO
 REPEAT PAIR CHAR=0         PSWD CHARACTER LIST=NONE
 GLOBAL MINDAYS=0           GLOBAL MAXDAYS=0           SKIP TEMP PSWD AGING=NO
 EXTENDED PSWD HIST=NO      EXTENDED PSWD HIST #=0     MIXED CASE PASSWORDS=NO

 UID STRING = LID

 DECOMP AUTHORITY = SECURITY, AUDIT

 INFO LIST AUTHORITY = SECURITY, AUDIT

 VOLUME PSEUDO DSN = @VOLSER.VOLUME

 -- VOLSER PROTECTED VOLUMES --
 ******
```

Information on this screen includes the defined resource types, the size of the resource rule cache, resident resource types, the resource class translation values for CAISSF (CA Standard Security Facility), and parameters that affect the construction of the pseudo data set name used in rule interpretation.

```
ACF2 RESOURCE TYPES (VMO RESCLASS):
DIAL:     DIA             AUTOLOG:  ALG             IUCV:     IUC
ACCOUNT:  ACT             VMCF:     VMC             GRPLOGON: GRP
DSPACE:   DSP             POSIXGRP: PGR

RESOURCE RULE CACHE SIZE = 64K

RESIDENT RESOURCE TYPES:
ACT, ALG, DIA, DSP, GRP, IUC, PGR, VMC

SSF TYPE AND CLASS LIST:
ACTAP                     ACA
ACTDI                     ACD
ACTLS                     ACL
ACTME                     ACM
ACTPA                     ACP
ACTQU                     ACQ
ACTRP                     ACR
CACMD                     CAC
CADOC                     CAD
CALIBMEM                  LBM
JOBNAME                   JOB
PANEL                     PAN
RECIPID                   REC
SCHEDULE                  SCH
STATION                   STA
SUBMIT                    SUB
UNVEDIT                   UNE
UNVPGM                    UNP
UNVRPRT                   UNR
VMANAPPL                  VMA
VTRMNODE                  VTR

-- DATASET RULE INTERPRET PARAMETERS --
ATTACH KEY RULEID = SYSTEM
NUMBER OF DIGITS IN AN ATTACH ADDRESS  = 4
NUMBER OF DIGITS IN A MINIDISK ADDRESS = 4
```

Displaying Service Machine Options

To display options the control program component of CA ACF2 for VM uses, including the name of the CA ACF2 for VM service machine, the number of command limiting and diagnose limiting cache entries, and the FORCEID, AUTOLOG, and NOAUTOU lists, enter:

```
SHOW SYSTEMS
```

```
ACF
 show systems

 -- SRVMOPTS PARAMETERS IN EFFECT --

 ACF2 SERVICE MACHINE = ACF2VM

 CMS SECURITY      = YES
 LINKRPI           = CALL       DIRMDPW             = RESPECT
 TRACEID           = 3F         NOAUTO              = NOPASS
 LOGONID TRAP      = YES        UNIT RECORD LOGGING = YES
 VM USER FIELD     = VMUSER1

 COMMAND/DIAG LIMITING CACHE ENTRIES =  50

 RPROMPT = YES               IPROMPT = YES   PRDISK = NONE

 -- ACF2 VM SYSID VALUES --

 ACFFDR SYSID  = *NONE*
 STARTUP SYSID = 2NDLEVEL  SET BY: *CP-SYSID*
 CURRENT SYSID = 2NDLEVEL  SET BY: *STARTUP*

 -- IUCV-APPC/VM-VMCF VALIDATION PARAMETERS --

 USERIDS/SYSTEM SERVICES SUBJECT TO VALIDATION:

 ALL USERIDS AND SYSTEM SERVICES

 -- FORCEID LIST --

 MAINT
 DIRM2

 -- NOAUTOU LIST --

 MAINT

 -- AUTOLOG LIST --

 AUTOLOG1

 -- NATIONAL LANGUAGE SUPPORT PARAMETERS --

 AMENG    DMKAF0
 UCENG    DMKAL0

 CMS FILEID TRANSLATE CHARACTERS:

 SOURCE    IMAGE
```

Displaying the UID Definition

To display how your site defines its UID string, enter:

```
SHOW UID
```

```
ACF

show uid
UID STRING = COMPANY,SITE,LEVEL,PROJECT,LID,IDNUM
```

Displaying Fields that Cannot Be Copied

To display the fields that will not be copied into new logonid records when created with the ACF INSERT USING subcommand, enter:

```
SHOW ZEROFLDS
```

PASSWORD and PSWD-TOD always appear because CA ACF2 for VM ensures that they are never copied.

```
ACF

 show zeroflds
 -- FIELD VALUES WHICH WILL NOT BE COPIED DURING 'INSERT USING' PROCESSING --
 PASSWORD   PSWD-TOD   ACC-CNT    ACC-DATE   ACC-SRCE   ACC-TIME
 ACCOUNT    ACCTPRIV   ACF2CICS   AUDIT      AUTHSUP1   AUTHSUP2
 AUTHSUP3   AUTHSUP4   AUTHSUP5   AUTHSUP6   AUTHSUP7   AUTHSUP8
 AUTOALL    AUTODUMP   AUTONOPW   AUTOONLY   BDT        CMD-PROP
 CONSOLE    CONSULT    DG84DIR    DIALBYP    GROUP      GRP-OPT
 GRP-USER   GRPLOGON   HOMENODE   JOBFROM    LEADER     LOGSHIFT
 MAINT      MOUNT      MUSASS     MUSUPDT    NAME       NO-SMC
 NOMAXVIO   NON-CNCL   NOSPOOL    OPERATOR   PHONE      PPGM
 PRIV-CTL   PRV-TOD1   PRV-TOD2   PRV-TOD3   PRV-TOD4   PRVPSWD1
 PRVPSWD2   PRVPSWD3   PRVPSWD4   PSWD-DAT   PSWD-INV   PSWD-SRC
 PSWD-TIM   PSWD-VIO   PSWD-XTR   PSWD-XTV   READALL    REFRESH
 RESTRICT   RSRCVLD    RULEVLD    SCPLIST    SEC-VIO    SECURITY
 SHIFT      SRF        SYNCNODE   SYNERR     TAPE-BLP   TDISKVLD
 TSORBA     UNICNTR    UPD-TOD    VLDVMACT   VMACCT     VMD4AUTH
 VMD4RSET   VMD4TARG   VMESM      VMSAF      VMSFS      VSESRF
 ZONE
```

Masking ACF Subcommands

You can use the LIKE and UID operands with the CHANGE, DECOMP, DELETE, and LIST subcommands of the ACF command under certain ACF command settings. When you use the LIKE operand, you must mask its value. When you use the UID operand, specify a portion of the UID, and CA ACF2 for VM automatically pads out the rest with standard masking characters.

Operand	ACF Subcommand Settings	Valid under ACF Subcommands
LIKE (must be masked)	LID	LIST CHANGE DELETE
	RULE	DECOMP
	ENTRY	LIST CHANGE DELETE
	SCOPE	LIST CHANGE DELETE
	SHIFT	LIST CHANGE DELETE
	CMDLIM	DECOMP LIST
	DIAGLIM	DECOMP LIST
	RESOURCE	DECOMP DELETE
UID (needs no masking)	LID	LIST CHANGE DELETE

Masking the LIKE Operand

The LIKE operand of the CHANGE, DECOMP, DELETE, and LIST subcommands processes multiple logonid records, access rules, and other CA ACF2 for VM records. You must mask the value of this operand with at least one masking character. CA ACF2 for VM supports the asterisk and the dash. Otherwise, CA ACF2 for VM displays the following message:

```
ACFpgm008E SUPPLIED MASK DOES NOT CONTAIN ANY ASTERISKS
```

Asterisks

Asterisks in LIKE operand masks are substituted on a one-to-one basis for any character, including blanks. If the asterisk is at the end of the value or a trailing asterisk, it can also match a null. When you use asterisks, no padding takes place.

```
LIST LIKE(PRG*)
```

This LIST subcommand under the LID setting lists PRG and PRG1, but does not include PRG12.

```
LIST LIKE(LID*)
```

The LIST subcommand under the SCOPE setting lists LID and LID1, but not LID12.

```
DECOMP LIKE(SYS*)
```

The DECOMP subcommand under the RULE setting decompiles SYS1 and SYS2, but not SYSABC.

Dashes

When you use the LIKE parameter, you can use a dash as the last position of the masked field. The field pads out to its maximum length.

```
LIST LIKE(PRG-)
```

This LIST subcommand under the LID setting might list the following logonid:

PRG

PRG1

PRG11

PRG111

The following LIST subcommand under the SCOPE setting might list all logonids:

```
LIST LIKE(-)
```

Unlike the asterisk that translates on a one-to-one basis, the dash at the end of the mask results in padding out the field. Imbedded dashes in a field or those that are leading characters of a field are treated **as literal** dashes.

Chapter 5: Using REGISTER

CA-REGISTER is a component of CA Common Infrastructure (CA-CIS) that lets your site automate the process of identifying users to multiple applications or system products. Rather than entering user information over and over again, CA-REGISTER provides your site with a single point registration facility. It lets you add and delete users to and from the CA ACF2 for VM databases, as well as from other products and applications.

CA-REGISTER functions independently of CA ACF2 for VM. It has a separate installation procedure as part of CA-CIS. CA-REGISTER is really a shell that provides an interface between you and the applications at your site. CA ACF2 for VM provides you with an application to be used by CA-REGISTER. You can provide other applications for CA-REGISTER.

You can use the REGISTER panels or the RGCMD command to add and delete users from the system. Both methods are easy to use.

This chapter explains how to:

- Access REGISTER
- Use the REGISTER PF keys
- Use the REGISTER screens to add and delete users
- Use the REGISTER screens to define applications to CA-REGISTER
- Use the RGCMD command to add and delete users
- Send data to other VM applications
- Override prototypes with the ZOOM function

This section contains the following topics:

Accessing CA-REGISTER

CA-REGISTER files reside on the CAIMAINT 291 disk. CA-REGISTER users should always have read-only access to this disk. To access REGISTER, type the following command at the VM Ready prompt and press Enter:

```
REGISTER
```

The REGISTER menu appears:

```
RGST-0000     Single Point Registration       CA-REGISTER
===>

  Userid  ________

  Option __


  Enter the UserID and the option, then press the ENTER key:

       1 Add user

       2 Delete user



PF1=Help      2=       3=End        4=Return      5=       6=
PF7=          8=       9=          10=           11=      12=Cursor
```

The options shown on this screen are defined below.

Userid

The user ID of the user you are defining or deleting. This ID can be up to eight characters long.

Option

The function you want to perform. The values you can enter are:

1

Add a user

2

Delete a user.

If you already know the user ID and the function you want to perform (ADD or DELETE), you can use the fast path method to can bypass this screen. For example, to delete the directory entry for user TLCAMS, enter:

```
REGISTER DELETE TLCAMS
```

REGISTER displays the screen that corresponds to the function you are performing.

Using the PF Keys

REGISTER has defined some PF keys to make it easy to use REGISTER. This table describes these PF keys and the default functions. Your site may have redefined these keys. If so, contact your local systems programmer for this information.

Key	Function	Description
PF1	Help	Displays Help information.
PF2	Send	Sends user information to the system for batch processing.
PF3	End	Clears the screen and redisplays the previous screen.
PF4	Return	Clears the screen and redisplays the first screen in the series.
PF6	Zoom	Lets you override prototypes.
PF12	Cursor	Toggles the cursor between the command line and screen prompts.

Adding a User in CA-REGISTER

If you selected Option 1 from the Single Point Registration screen (Add a User), or if you used the REGISTER command with the ADD option, you see this screen:

```
RGST-1000              Add a User                  CA-REGISTER
===>

  Userid ________ Group ________Initial Password ________
  Name _____________________________________ Phone ___ ___ ____
  Info ____________________________________________________

  Option: 1-Include, 2-Exclude

  Opt    Application    Prototype   Exit Name   Zoom Name
  --- ----------------  --------    --------    --------
   1  CA-ACF2           ________    ACF2EXIT    ACF2ZOOM
   1  SFS               ________    SFSEXIT     SFSZOOM
   1  VM:DIRECTOR       ________    ________    ________


PF1=Help      2=Submit       3=End        4=Return      5=          6=Zoom
PF7=          8=             9=          10=           11=         12=Cursor
```

The fields for this screen are defined below.

Userid

Specifies the ID of the user you are adding (one to eight characters).

Group

Specifies an optional value that identifies the group that the user belongs to. You can use this field to indicate the user's department or job function (one to eight characters).

Initial Password

Specifies the user's initial logon password (one to eight characters).

Name

Specifies the full name of the user that you are adding (1 to 40 characters).

Phone

Specifies the user's telephone number or extension. This information is optional (one to ten characters).

Info

Specifies information about the user, such as his department or title. This information is optional (1 to 60 characters).

Opt

Use options 1 or 2 on this field to indicate whether you want REGISTER to invoke the application exit. REGISTER invokes the exit when you indicate 1, but not when you indicate 2. The default is 1.

Application

Specifies the names of the applications that your site uses. These systems are site-specific. The screen you see can contain different system names.

Prototype

Specifies the model name that contains the basic outline for a user ID. You can type over this field with a different prototype name.

Exit Name

Specifies the exit name the REGISTER service machine executes.

Zoom Name

Specifies the zoom program name that lets you override the prototype values.

To process information you entered on this screen, press PF2. When you press PF2, REGISTER assigns the command a transaction number and sends it to batch for processing. You will receive a message with the transaction number. When processing is complete, REGISTER creates a file named REGISTER ft, where ft is your transaction number. REGISTER sends the file to your reader. REGISTER creates registration schemas for these systems. This means that REGISTER creates a directory entry (a logonid) for the new user.

You can also override the prototype for this user ID. Move your cursor to the prototype name and press PF6. REGISTER displays the default information from the prototype. Change any definitions you want and press PF2 to save your changes. Press PF3 to return to the Add a User screen. When you add the new user ID, it includes the overrides to the prototype. For more information, see the Overriding Prototypes section.

Deleting a User in CA-REGISTER

If you selected Option 2 from the Single Point Registration screen or if you used the DELETE option with the REGISTER command, you see this screen:

```
RGST-2000              Delete a User          REGISTER
===>


   Userid TLCAMS   Group _________

   Option: 1-Include, 2-Exclude

  Opt    Application    Prototype   Exit Name   Zoom Name
  --- ----------------   --------    --------    --------
   1  CA-ACF2            ________    ACF2EXIT    ACF2ZOOM
   1  SFS                ________    SFSEXIT     SFSZOOM
   1  VM:DIRECTOR        ________    ________    ________



PF1=Help      2=Submit        3=End        4=Return      5=          6=
PF7=          8=              9=          10=           11=         12=Cursor
```

The fields for this screen are defined below.

Userid

Specifies the name of the user you are removing (one to eight characters).

Group

Specifies the group that the user belongs to. You can use this field to indicate the user's department or job function (one to eight characters).

Opt

Use options 1 or 2 in this field to indicate whether you want to delete this user ID definition for the application. REGISTER deletes the user ID from the applications you indicate with 1, but not those you indicate with 2.
The default is 1.

Application

Displays the names of other VM software products your site uses. These systems are site-specific. The screen you see can contain different names.

Prototype

Specifies the model name that contains the information for a user ID.

Exit Name

Specifies the exit name the REGISTER service machine executes.

Zoom Name

Specifies the zoom program name that lets you override the prototype values. For example, you can indicate to CA ACF2 for VM whether or not rules associated with the ID should be deleted.

When you press PF2, REGISTER submits the transaction to remove this user's registration from all VM software that REGISTER maintains.

Defining Applications to CA-REGISTER

This section shows you how to define applications to CA-REGISTER. CA-REGISTER allows you to combine the following applications in VM:

- CA supplied enrollment applications
- Site-developed applications
- Other vendor applications
- REGISTER service machine

The CA ACF2 for VM interface with REGISTER consists of:

ACF2ZOOM CAIPANEL file

A program, from the REGISTER Zoom function, called during the addition or deletion of a user. ACF2ZOOM interfaces to the CA ACF2 for VM Define a User screen (M9PA-1110) and the CA ACF2 for VM Delete a User screen (M9PA-1410). The ACF2ZOOM CAIPANEL file resides on the CAIMAINT 291 disk.

ACF2EXIT EXEC file

A program that receives arguments from the REGISTER panels and the CA ACF2 for VM panels. It issues commands to add or delete a logonid record and its associated access rules and resource rules from the databases. This file resides on the CAIMAINT 291 disk.

The following steps provide information necessary to set up the connection between CA ACF2 for VM and CA-REGISTER:

1. Define the CA ACF2 for VM application entry in the REGISTER APPLS file. You can do this from the Define “REGISTER APPLS” File screen (S510-I002). This screen is part of the CA-REGISTER installation tasks that are invoked when installing CA-CIS from CAIMAINT.

CA-Activator displays this screen:

```
S510-I002     Define "REGISTER APPLS" File      REGISTER
===>

Options: 1=Add 2=Delete

                          Default  Application   Zoom
  Opt  Application Name    Model      Exit     Program
  ---  ----------------  --------  --------   --------
   _   CA-ACF2           ________  ACF2EXIT   ACF2ZOOM
   _   SFS               ________  SFSEXIT    SFSZOOM
   _   VM:DIRECTOR       ________  ________   ________


PF1=Help      2=Save       3=End        4=Return      5=       6=
PF7=Backward  8=Forward    9=          10=           11=      12=Cursor
```

Opt

Use option 1 or 2 in this field to indicate whether you want to add or delete this application name.

Application name

Specifies the names of the applications that your site uses.

Default model

Specifies the name of the model. Use this field with the INSERT LID subcommand.

Application exit

Specifies the name of the program that issues the commands to add or delete user IDs. For CA ACF2 for VM, it is the ACF2EXIT command.

Zoom Program

Specifies the name of the program that is called when you press the PF6 key (for the Zoom function). For CA ACF2 for VM, it is the ACF2ZOOM command.

After you finish editing, press PF2 to save the file on the A-disk. Each REGISTER user needs access to this file. These users can also have unique versions on their A-disk. Consult the CA-Register documentation for more information.

The REGISTER service machine issues the surrogation diagnoses (diagnose x'D4') to switch to the authority of the command issuer. To allow REGISTER to do a diagnose x'D4, you must assign the REGISTER service machine the following CA ACF2 for VM logonid privileges:

- VMD4TARG
- VMD4AUTH
- VMD4RSET

Using the RGCMD Command

You can also use the RGCMD command to perform REGISTER functions. The RGCMD command is a line command that lets you add or delete users and manage the REGISTER service machine. You do not receive an immediate response when you issue the RGCMD command. Instead, REGISTER assigns you a transaction number. When processing is complete, REGISTER sends you the console log that contains messages and all return codes. The file you receive is the REGISTER transaction number.

The syntax for the RGCMD command is:

```
      { ADD      } {USER(userid) GROUP(grpid) PHONE(number)
      {          } NAME(user) PARM(info) ZOOM(exitnam,zmdata)
      {          } PW(initpw)}
      {          }
      { DELete   } {USER(userid) GROUP(grpid) PHONE(number)
      {          } NAME(user) PARM(info) ZOOM(exitnam,zmdata)
      {          } PW(initpw)}
      {          }
      { PURge    } transnum
RGCMD {          }
      { Query    } {TRANs|ALL}
      {          }
      { SHUTdown } {TRANEND }
      {          } {IMMed   }
      {          }
      { ?        } [subcommand]
```

Parameter Descriptions

ADD

Defines a user to all applications defined in the REGISTER APPLS file the user has access to.

ALL

Affects all transactions.

DELete

Removes a user to all applications defined in the REGISTER APPLS file the user has access to.

GROUP(grpid)

Specifies an optional value that identifies the group that the user belongs to. Some systems, such as CA Director, use groups as the basis for user privileges. (One to eight characters.)

IMMed

Shuts down the REGISTER service machine immediately.

NAME(user)

Specifies the full name of the user (1 to 40 characters).

PARM(info)

Identifies parameters specific to selected applications that are passed to the application exit. Your systems programmer must code the application exit to scan the info line for data that may apply to it. (1 to 60 characters.)

PHONE(number)

Specifies the user's telephone number or extension. This information is optional. (1 to 10 characters.)

PURge

Removes the transaction from the queue.

PW(initpw)

Specifies the user's initial logon password. The application exits can use this password. (One to eight characters.)

Query

Displays information on the transaction in the REGISTER service machine.

SHUTdown

Terminates the REGISTER service machine.

TRANEND

Shuts down the REGISTER service machine when the current transaction ends. This is the default.

TRANs

Returns the status of transactions in REGISTER.

transnum

Specifies the four-digit REGISTER transaction number.

USER(userid)

Specifies the ID of the user you are adding or removing. (One to eight characters.)

ZOOM(exitnam,zmdata)

Specifies the exit name and data block that is passed to the REGISTER service machine. You can specify multiple entries, but you must specify them in pairs. The exit name is the name of the exit program. In a full-screen zoom, REGISTER obtains it from the application exit field in the REGISTER APPLS file.

This field overrides the REGISTER APPLS file. If you specify this parm, you must have a pair for each application exit to call in the REGISTER service machine, even if the zmdata is null. The order of the pairs affects the order that REGISTER calls the application exits.

You can only specify this parameter if a REXX program invokes RGCMD.

?

Displays the syntax of all of the RGCMD subcommands and operands. If you specify a subcommand with the ?, displays only that subcommand and syntax.

Overriding Prototype Values

As you have learned, when you define users to software systems, you create unique user IDs. For example, when your site installs CA ACF2 for VM, you create a unique logonid for each user. These IDs identify users to the system. Because many logonids are similar to each other, REGISTER lets you use prototypes when you create these IDs. Prototypes contain the basic outline for a user ID. Alter one or two values in the model to create a unique user ID.

The prototypes maintained in REGISTER are distinct from models used in other VM systems. The REGISTER prototypes contain specifications of default values that user IDs share. Because you can determine these defaults, you can also tailor these prototypes to your needs. REGISTER lists these prototypes on the Add a User screen. From this screen, you can then override the prototypes with the Zoom function.

The Zoom Function

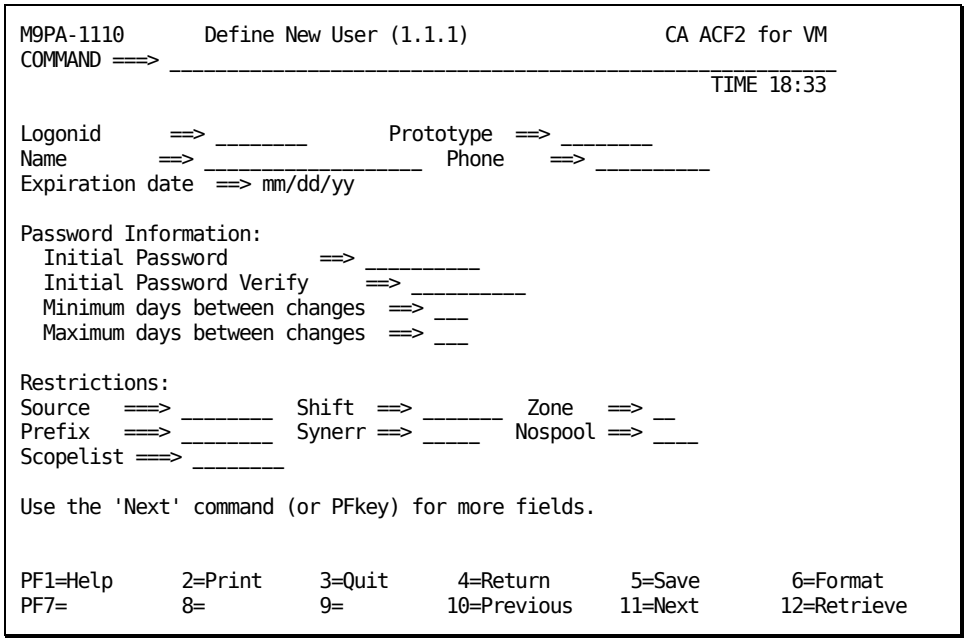
To override values in a REGISTER prototype, move your cursor to the prototype name and press PF6. This PF key toggles you into the system panels or programs to let you override the default prototype values. How you save your changes is product-specific. Refer to that product's documentation for information.

When you use REGISTER to register users to other VM software systems, each system must have an application exit that retrieves information from REGISTER. That is, the exit does all the work. All CA VM software products that require user registration provide such exits. Your systems programmer should write exits for your other VM software products that require user registration.

Defining Users through the Zoom Function

If you select the Zoom function from the REGISTER add panel, the CA ACF2 for VM Define New User screen is displayed. If the REGISTER add panel has text in the model field, that model's information is obtained from the CA ACF2 for VM Logonid database and displayed on the Define New User screen. A prototype in REGISTER is identical to the model in CA ACF2 for VM or the same as issuing the ACF INSERT USING(lid) subcommand.

From this point, you can select any of the Define New User screens with your NEXT and PREVIOUS PF keys. If you press your SAVE PF key, all collected information passes back to REGISTER, the REGISTER service machine, and the ACF2EXIT program. The CA ACF2 for VM Define New User screen is illustrated below.



For an explanation of the fields depicted on this screen, see the “Maintaining Logonids with the Full-Screen Feature” chapter.

Deleting Users through the Zoom Function

If you select the Zoom function from the REGISTER delete user screen, REGISTER displays the Delete User Verification screen. This screen displays user information so that you can verify the user ID you selected for deletion. This screen also displays any rules associated with this logonid, such as access rules, resource rules, and so on.

From this panel, enter a Y|N in the selection field to select whether to delete the associated rules. For example, suppose you were deleting user ID TLCAMS. When REGISTER displayed the information for TLCAMS, it found ACCESS, GRPLOGON, and DIAL rules. If you wanted to delete these rules, leave the Y as is, and press the SAVE PF key. If you did not want to delete these rules, enter an N in the selection field and press the SAVE PF key. The screen (if you selected Zoom) or an option in the ACF2EXIT program (if you did not select Zoom) controls the default for selecting automatic rule deletion. Consult the *Systems Programmer Guid*e for more details. The CA ACF2 for VM Delete User Verification screen is illustrated below.

```
M9PA-1410     Delete User Verification (1.4.1)       CA ACF2 for VM
COMMAND ===> _____________________________________________________
                                                        TIME 14:31

Identification:
 Logonid         :
 Name            :                       Phone  :
 Expiration Date :

Password Information:
 Expired         :                  Violations     :
 Minimum days    :                  Maximum days   :
 Date - time set :                  Violation date :

Delete Users Access Rule  ==> Y

Y to Delete Associated Resource Rules   N to bypass Deletion
 AUTOLOG  ALG    : N     DIAL     DIA   : Y     GRPLOGON  GRP   : Y
   IUCV   IUC    : N     VMCF     VMC   : N

Use 'Execute' to perform the delete.



PF1=Help      2=           3=Quit       4=Return       5=Save    6=
PF7=          8=           9=          10=Previous    11=Next    12=Retrieve
```

If you want to change the default settings for resources, (resource types you select as associations of a logonid on the Delete a User screen) locate the following section in the Delete User Verification panel:

```
/*RESOURCE
AUTOLOG ALG 1
DIAL     DIA 1
GRPLOGON GRP 1
IUCV     IUC 1
VMCF VMC 1
ERESOURCE*/
```

The RESOURCE name is the first token, the resource type is the second token. The third token is the default action. 1 is YES (delete rule); 0 is NO.

REGISTER Service Machine Exit for Adding Users

The REGISTER service machine requires exit code to handle the actual adding or deleting of user IDs. For CA ACF2 for VM, this is the ACF2EXIT EXEC.

Chapter 6: About the Logonid Record

Every logonid has certain privileges assigned to it. Most users may only have the basic privileges shown on the following pages, while more powerful users (department heads, for example) have more powerful logonid privileges assigned to them.

The CA ACF2 for VM Field Definition Record (ACFFDR) is an assembly of a module that defines the fields in the logonid record and specifies other VM system options. The @CFDE macro in the ACFFDR defines the name of each field in the logonid record. It also relates the CA ACF2 for VM privileges you need to display or change logonid record fields. For more information about the ACFFDR macros, see *Installation Guide*.

When you finish this chapter, you will know:

- About the different types of CA ACF2 for VM users and field types
- About the different types of logonids
- About the various logonid privileges that a user can have
- How to assign and limit the powers of various users
- How to mask logonids.

This section contains the following topics:

Types of Users

There are six types of CA ACF2 for VM users: ACCOUNT, AUDIT, CONSULT, LEADER, SECURITY, and USER. Each logonid for a specified user has certain privileges associated with it.

ACCOUNT

Indicates that this user is an account manager. An account manager can insert, delete, and change logonid records in the limits defined by his scope. Account managers usually establish, maintain, and delete logonid records. The ACCOUNT privilege grants no authority for writing rules or processing other CA ACF2 for VM records.

SECURITY

Indicates that this user is a security administrator and can access all data, protected programs, and resources. A security administrator can insert, change, list, and delete access and resource rules and list and change certain fields of logonid records. He cannot insert or delete logonid records unless he also has the ACCOUNT privilege. He can insert, change, list, and delete any infostorage records in his scope. You can use a scope record to protect any access granted by the SECURITY privilege.

AUDIT

Indicates that this user can list logonid records, access and resource rules, entry records, shift records, zone records, scope records, and VMO records. He cannot update or delete logonid records, or access any resources other than those authorized to him through rules.

CONSULT

Indicates that this user can display most fields of logonid records, and update only certain nonsecurity-related fields. The CONSULT privilege is usually given to individuals who assist other users on the system so they can answer questions about logonid record information.

LEADER

Indicates that this user has the same privileges as CONSULT. However, can only display and modify certain fields of other logonid records as defined by the SCPLIST field. (For detailed information on SCPLIST, see the Administrator Guide.)

USER

Indicates that this user can display their own logonid record. You can specify whether a user with only the USER privilege can write access rules for their own data with a system-wide option. A user can have more than one privilege. For example, he can have both SECURITY and ACCOUNT, which gives him all authorities associated with the SECURITY privilege and all authorities associated with the ACCOUNT privilege.

Types of Fields

Each logonid field also has an associated type code that tells CA ACF2 for VM how to process it. The TYPE operand of its corresponding @CFDE macro in the ACFFDR defines the type code for each field. The CA ACF2 for VM logonid record field types are defined below.

Binary field

Specifies a one, two, three, or four-byte binary. You can define these fields in the ACFFDR from one- to four-bytes long. This specification defines the maximum value that you can assign to the field. Only binary two- and four-byte fields can contain negative values.

Bit field

Specifies a field that contains binary code, indicating that the field is active or inactive (on or off).

Character field

Specifies a field of 1-255 bytes. Specify this type of field with the fieldname followed by the replacement characters in parentheses. CA ACF2 for VM assigns the character string you specify to the appropriate field in the record and fills it on the right with blanks. To blank out a field, use a null (()).

Date field

Specifies a packed decimal date field of four bytes. Specify this field with the fieldname followed by the Gregorian date in parentheses (mm/dd/yy, or yy/mm/dd, depending on the DATE operand in the OPTS VMO record). You can suppress the field listing if the value is zero.

Hexadecimal field

Specifies a field up to eight bytes long (16 hexadecimal digits). You can enter valid hexadecimal digits (0-9, A-F) for an even number of characters in parentheses. The field is left-justified and filled on the right with hexadecimal zeros. You can suppress the field listing if the value is zero.

Logonid Record Fields

This section contains all the CA ACF2 for VM-defined logonid fields that a user can have. The fields below are arranged according to group number. For an alphabetized list of all the CA ACF2 for VM logonid fields and their descriptions (including z/OS-defined fields), see the Logonid Record Field Descriptions section.

Logonid Record Field Groups	Fields
Identification (Group 0)	LOGONID NAME PASSWORD PHONE UID
Cancel/Suspend (Group 1)	CANCEL CSDATE CSWHO SUSPEND TRACE

Logonid Record Field Groups	Fields
Privileges (Group 2)	ACCOUNT AUDIT AUTOALL AUTONOPW AUTOONLY CONSULT DG84DIR DIALBYP DUMPAUTH EXPIRE GRPLOGON GRP-OPT LEADER LDEV LIDSCOPE LOGSHIFT NOSPOOL NO-STORE NON-CNCL PGM PSWD-UPP READALL RESTRICT RULEVLD VMD4FSEC SCPLIST SECURITY SRF STC SYNERR TDISKVLD USER VLDVMACT VM VMD4AUTH VMD4FSEC VMD4RSET VMD4TARG VMESM VMSAF VMSFS VMXA VSESRF

Logonid Record Field Groups	Fields
Access (Group 3)	ACC-CNT ACC-DATE ACC-SRCE ACC-TIME GRP-USER
Password (Group 4)	LIDTEMP LIDZMAX LIDZMIN MAXDAYS MINDAYS PSWD-DAT PSWD-EXP PSWD-MIX PSWD-TOD PSWD-VIO PSWDCVIO
TSO (Group 5)	MODE
Statistics (Group 6)	CRE-TOD SEC-VIO UPD-TOD
CICS (Group 7)	All the fields in this group are only active for z/OS sites. If you need additional information on these fields, refer to the CA ACF2 for VM *for z/OS CICS Support Guide*. CICSCL CICSID CICSPRI CICSRSL IDLE
IMS (Group 8)	This group contains any fields developed for the CA ACF2 for VM IMS security subsystem.
IDMS (Group 9)	All the fields in this group are active only for z/OS sites. For more nformation on these fields, see the CA ACF2 for VM *for z/OS Administrator Guide*.
MUSASS (Group 10)	The fields in this group are active only for OS/390 sites. See the *Administrator Guide* for information on these fields. MUSDLID MUSID MUSUPDT

Logonid Record Field Groups	Fields
Restrictions (Group 11)	PREFIX SHIFT SOURCE VMACCT VMIDLEMN VMIDLEOP ZONE
DFP (Group 12)	The fields in this group are active only for OS/390 sites. See the *Administrator Guid*e for information on these fields. DATACLAS MGMTCLAS STORCLAS

Logonid Record Field Descriptions

The logonid record fields listed below are arranged in alphabetical order. For a list of the logonid record fields according to group number, see the Logonid Record Fields section.

ACF2CICS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

ACC-CNT

Indicates the number of system accesses this logonid made since it was created. The maximum number is 999,999. This is a binary field.

ACC-DATE

Indicates the date of this user's last system access. The date is specified in the dd/mm/yy, mm/dd/yy, or yy/mm/dd format, depending on the DATE parameter in the OPTS VMO record. This is a date field.

ACC-SRCE

Indicates the address of the input device used to enter the system. This is a character field.

ACC-TIME

Indicates the time that this user last accessed the system. The format is hh.mm.ss. This is a binary field.

ACCOUNT

Indicates that this user is an account manager. With this privilege, an account manager can insert, delete, and change logonids as limited by the SCPLIST privilege. A user with the ACCOUNT only or SECURITY only privilege cannot list or change a logonid record for a user who has both ACCOUNT and SECURITY. This is a bit field.

ACCTPRIV

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

ACTIVE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

ALLCMDS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

ATTR2

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUDIT

Indicates that this user is an auditor. An auditor can inspect (but not modify) the parameters of CA ACF2 for VM. The SCPLIST privilege limits this privilege. This is a bit field.

AUTHSUP1

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTHSUP2

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTHSUP3

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTHSUP4

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTHSUP5

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTHSUP6

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTHSUP7

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTHSUP8

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTOALL

Indicates that this user can autolog any machine without entering a password. No autolog resource rules are required, but CA ACF2 for VM generates a logging record if no rules exist. This is a bit field.

AUTODUMP

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

AUTONOPW

Indicates that this logonid can be autologged without requiring a password. Autolog resource rules must exist. This is a bit field.

AUTOONLY

Indicates that this logonid can be autologged, but no one can log onto this ID. This is a bit field.

BDT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CANCEL

Indicates that this logonid has been canceled. This user cannot access the system. This is a bit field.

CHAR

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CICS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CICSCL

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CICSID

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CICSPRI

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CICSRSL

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CMD-LONG

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CMD-PROP

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CONSOLE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

CONSULT

Indicates that this user can display other logonid records, The SCPLIST privilege limits this privilege. This is a bit field.

CRE-TOD(*date-time*)

Indicates the date and time that a logonid record was created. CA ACF2 for VM displays the date in the format *mm/dd/yy, dd/mm/yy,* or *yy/mm/dd*, depending on the DATE field of the VMO OPTS record. Year designations of 70-99 assume a date in the 20th century (1970-1999); year designations of 00-42 assume a date in the 21st century (2000-2042). You can change the date, but you cannot change the time. You cannot specify a date past the current day's date. You must have the SECURITY or ACCOUNT privilege to alter this field. (Eight-byte binary field)

CSDATE

Indicates the date that the CANCEL or SUSPEND field has been set for this user. The format is in mm/dd/yy, dd/mm/yy, or yy/mm/dd, depending on the OPTS VMO record. This is a date field.

CSWHO

Indicates the logonid of the user who set the CANCEL or SUSPEND for this user.

DFT-DEST

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

DFT-PFX

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

DFT-SOUT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

DFT-SUBC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

DFT-SUBH

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

DFT-SUBM

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

DG84DIR

Indicates that this virtual machine has diagnose 84 passwords validated against the VM directory. If the virtual machine does not have this attribute, diagnose 84 passwords are validated against the Logonid database. This attribute is usually assigned to VM class B users that do directory management or system backup functions.

DIALBYP

Indicates that this logonid bypasses DIAL command validation for dials to this ID. When a user dials to a secured target machine with this privilege, standard DIAL validation and logging does not occur. If this privilege is not granted, the dialer is prompted for a logonid and password and standard DIAL validation occurs.

DSNSCOPE(*logonid mask*)

Specifies a logonid mask limiting the scope of SECURITY access. (Eight-byte character field)

Important! This is an old field left over from pre-3.1 release of CA ACF2 for VM. This field should not be used. To limit a user's administrative authority over the CA ACF2 for VM logonid, rule, and Infostorage databases, use the SCPLIST field in the logonid record with scope records to limit a user's administrative authority. For more information on how to create scope records, see the chapter "Processing Scope Records."

DUMPAUTH

Indicates that this user can use the CP DISPLAY, DUMP, PER, TRACE, and VMDUMP commands to display storage and trace programs, EXEC files, and XEDIT macros, even when an execute-only EXEC or MODULE is in storage. It also allows EXEC tracing when an execute-only EXEC or MODULE is in storage.

EXPIRE

Indicates the date that this logonid expires. On this date, the user cannot log on or submit jobs. This date must be in the format mm/dd/yy, dd/mm/yy, or yy/mm/dd, depending on how your site defined the format in the OPTS VMO record. You can remove this EXPIRE restriction with the ACF CHANGE subcommand. For example, change the user's logonid record to specify EXPIRE(0).

GROUP

Defines the user's primary group for the POSIX environment. Refer to the chapter “OpenExtensions VM Support” for more information.

GRP-OPT

Designates an ID as an optional group ID. A logonid with this attribute can be logged onto as the primary ID, or a group ID. To access a virtual machine with this attribute as a group ID, this privilege must be present and a group resource rule must exist. If the GRP-OPT and the GRPLOGON field are present, then the GRPLOGON attribute takes precedence. GRP-OPT requires using LOGON BY when logging on as a group user.

GRP-USER

Indicates the last user (logonid) to use the group virtual machine. If this machine is not a group machine, CA ACF2 for VM does not display this field.

GRPLOGON

Indicates that this logonid is a group virtual machine. This attribute is typically assigned to shared user IDs, such as MAINT. When a user accesses a virtual machine with the GRPLOGON attribute, he is prompted for his own logonid and password. Group logon resource rules control access to a group ID. This is a special access privilege. (For more information about the GRPLOGON privilege, see the “Logging onto CA-ACF2 Group Machines” chapter.

HOMENODE

This field is not active for VM sites. For additional information on this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

IDLE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

IDMS

This field is not active for VM sites. For more information on this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

IMS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

INTERCOM

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

JCL

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

JOB

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

JOBFROM

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

KERB-CUR

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

KERB-CURV

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

KERB-PRE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

KERB-VIO

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

KERB-PREV

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LDS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LDEV

Indicates that this user can create logical devices when using the IBM Passthru Virtual Machine (PVM) product. This privilege applies only when the optional CA ACF2 for VM intercept is in place.

LEADER

Indicates that this user can display and alter certain fields of other logonids. The SCPLIST privilege limits this privilege.

LGN-ACCT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LGN-DEST

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LGN-MSG

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LGN-PERF

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LGN-PROC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LGN-RCVR

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LGN-SIZE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LGN-TIME

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LGN-UNIT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LID

Indicates the key to the logonid record. In VM, this is usually also the user ID.

LIDSCOPE(*logonid mask*)

Specifies a logonid mask limiting the scope of SECURITY/ACCOUNT/LEADER access. (Eight-byte character field)

Important! This is an old field left over from pre-3.1 release of CA ACF2 for VM. This field should not be used. To limit a user's administrative authority over the CA ACF2 for VM logonid, rule, and Infostorage databases, use the SCPLIST field in the logonid record with scope records to limit a user's administrative authority. For more information on how to create scope records, see the chapter "Processing Scope Records."

LIDTEMP

Specifies that the current password is a temporary password. This bit will be set if the current password was set by a non-owner of the LOGONID (security administrator or account manager), and the password was immediately expired. This bit cannot be modified using the ACF command, and is not displayed by default. It is used for internal processing.

LIDZMAX

Specifies that a zero value for the MAXDAYS field in the LIDREC will override the global PSWDMAX value in the PSWD VMO record.

LIDZMIN

Specifies that a zero value for the MINDAYS field in the LIDREC will override the global PSWDMIN value in the VMO PSWD record.

LINE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

LOGSHIFT

Indicates that this user can access the system outside of the time period specified in the SHIFT field of his logonid record. All such system accesses are logged in SMF records and are listed in the ACFRPTPW report.

MAIL

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MAINT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MAXDAYS

Indicates the number of days (up to 255) that will elapse before this user is forced to change his password. If you set this field to zero, no limit is enforced.

MINDAYS

Indicates the number of days (up to 255) that must elapse before this user can change his password. This field prevents a user from immediately changing his password back to the previous password.

MODE

Indicates that the ACF command prompts the user with the current ACF command setting instead of the default **?**. This is a bit field.

MULTSIGN

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MON-LOG

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MONITOR

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MOUNT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MSGID

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MUSASS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MUSDLID

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MUSID

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MUSIDINF

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

MUSUPDT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

NAME

Indicates the user's name.

NO-INH

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

NO-OMVS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

NO-SMC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

NO-STATS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

NO-STORE

Indicates that this logonid cannot store rule sets (cannot make rule changes active) regardless of ownership (PREFIX values), SECURITY attribute, or delegation through %CHANGE.

NOMAXVIO

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

NON-CNCL

Indicates that this logonid has access to all data, but the PREFIX field or access rules would log all accesses that would not normally be allowed. The ACFRPTDS report shows the request was allowed because the user had the NON-CNCL attribute. CA ACF2 for VM never cancels a user with this privilege for security violations. The NON-CNCL privilege overrides RULEVLD.

NOSPOOL

Indicates how CA ACF2 for VM responds when a user with this privilege enters a command that results in the SPOOL FILE NOT FOUND condition.

Values for this field are PREVENT (CA ACF2 for VM rejects the command), PREVENT-LOG (CA ACF2 for VM rejects and logs the command), LOG

(CA ACF2 for VM passes the command to CP for normal syntax checking and generates a logging record), or ALLOW (CA ACF2 for VM passes the command to CP for normal syntax checking). If this field is null (or blank), NOSPOOL processing is passed to the command model COMMAND clause, then to the CMDLIM VMO record. The default value is a null.

NOTICES

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

OPERATOR

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PASSWORD

Specifies a 1- to 8-character password that is not displayed and is stored in encrypted format.

PAUSE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PGM

This field is not active for VM sites. For more information for this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PHONE

Specifies the user's telephone number or extension.

PMT-ACCT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PMT-PROC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PP-TRC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PP-TRCV

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PPGM

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PREFIX

Indicates the ID associated with the user's minidisks. The user's access to CMS files on a minidisk whose ID matches the PREFIX is always allowed. The PREFIX also identifies the access rule sets that the user can decompile and store. By default, PREFIX is the same as the user's logonid. You can mask this eight character field with asterisks (*), but not with dashes (-).

PRIV-CTL

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PROGRAM

This field is not active for VM sites. For more information for this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PROMPT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PRVPSWD1 through PRVPSWD4

These fields are not active for VM sites. For more information for this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PRV-TOD1 through PRV-TOD4

These fields are not active for VM sites. For more information for this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PSWD-DAT

Indicates the date of this user's last invalid password attempt.

PSWD-EXP

Indicates that the user's password has manually expired. This forces users to change their passwords. This is a bit field.

PSWD-INV

This field is not active for VM sites. For more information on this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PSWD-MIX

Indicates the current password is case sensitive. This means the PSWDMIXD setting in the PSWD VMO record was in effect when the current password was created or changed. This field is display only and cannot be changed.

PSWD-SRC

This field is not active for VM sites. For more information on this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PSWD-TIM

This field is not active for VM sites. For more information on this field, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PSWD-TOD

Indicates the date and time the password was last changed. This is a display-only field. You cannot change it.

PSWD-UPP|NOPSWD-UPP

Specifies that the new password is to be upper-case. PSWD-UPP does not affect the current password in any way. This field can be used in conjunction with the VMO PSWD record field PSWDMIXD. PSWDMIXD is the global specification that says passwords are case-sensitive. When PSWDMIXD is on, PSWD-UPP can be turned on to specify for this user that their new passwords will not be case-sensitive. PSWD-UPP should only be used as a means to exclude some users from having case-sensitive passwords. The default is NOPSWD-UPP. (Bit field)

PSWD-VIO

Indicates the number of password violations that occurred on PSWD-DAT. This is a binary field.

PSWD-XTR

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PSWD-XTV

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PSWDCVIO(*nn*)

Indicates the number of cumulative invalid password attempts for a user that occurred since the logonid record was created. The only time this field is physically set to zero (0) is when the CA ACF2 for VM security administrator resets the field. (Two-byte binary field)

PTICKET

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

PWP-DATE(*date*)

Specifies the date the user made the last invalid password phrase attempt. The date is displayed in the format mm/dd/yy, dd/mm/yy, or yy/mm/dd, depending on the DATE field of the VMO OPTS record. Year designations of 70-99 assume a date in the 20th century (1970-1999); year designations of 00-69 assume a date in the 21st century (2000-2069). (Four byte packed field)

PWP-VIO(*count*)

Specifies the number of password phrase violations that occurred on PWP-DATE. The PWP-VIO field is incremented by one for every password phrase violation incurred within the same date. Any password phrase violations incurred after the current value in PWP-DATE will cause the PWP-VIO count to be reset to 1 and the PWP-DATE field will be updated to reflect the current date. The only time the PWP-VIO field is physically set to zero (0) is when the CA ACF2 for VM security administrator resets the field. (2-byte binary).

PWPALLOW|NOPWPALLOW

Specifies whether a user can be authenticated using a password phrase even when the VMO PWPHRASE record does not specify ALLOW to enable all users to use password phrases. The default is NOPWPALLOW, the user can only use a password phrase if the VMO PWPHRASE record specifies ALLOW

READALL

Indicates that this logonid has read access to all data. This is similar to NON-CNCL, but grants read access only and enforces any existing rules for other types of accesses. This attribute is usually assigned to system backup products. The READALL privilege overrides RULEVLD.

RECOVER

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

REFRESH

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

RESTRICT

Indicates that a password cannot be supplied for logging onto a user ID with this privilege. This is compatible with the OS/390 setting. In VM, we suggest that you use the AUTOONLY setting.

Although you can use RESTRICT, you should consider the following:

- No user can log onto a user ID with the RESTRICT privilege active. CA ACF2 for VM aborts the logon and sends an error message.
- A user ID with RESTRICT can be autologged without a password from a user ID with AUTOALL.
- A user ID with RESTRICT can be autologged with a password if the AUTOLOG resource rules allow access.
- If a target user ID is GRPLOGON and RESTRICT, the group logon fails and CA ACF2 for VM sends an error message.
- If a group user ID has RESTRICT, CA ACF2 for VM prevents a group logon.
- CA ACF2 for VM allows a dial to a user ID with RESTRICT.

RSRCVLD

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

RSTDACC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

RULEVLD

Indicates that access rules must exist for all of this user's data accesses, even if the access is in a security administrator's SCPLIST or a user's SCPLIST. However, if this user defined a temporary disk, CA ACF2 for VM does not check temporary disks against rules unless TDISKVLD is in effect. NON-CNCL and READALL override RULEVLD. This is a bit field.

SCPLIST

Indicates the name of a scopelist record that limits this user from displaying or modifying CA ACF2 for VM records and rules. You must predefine scopelist records in the Infostorage database under the type code SCP. This field, in part, determines the user's limitations when displaying or modifying CA ACF2 for VM records and rules. If this field is null, no scoping is done. A user without a SCPLIST field is unscoped.

SEC-VIO

Indicates the total number of security violations that this user has. This is a binary field.

SECURITY

Indicates that this user is a security administrator. A security administrator can create and inspect access rules and update certain fields in logonid records. He can also access all data in the limits of his SCPLIST field. CA ACF2 for VM logs all accesses that his PREFIX field or access rules would not normally allow. He can also create and inspect records on the Infostorage database in the limits of his SCPLIST. This is a bit field.

SHIFT

Indicates the name of the shift record used for system entry validation. Shift records indicate time, days, or dates when a user can log on. If this field is null (or blank), CA ACF2 for VM does not validate the shift. You must predefine valid shift records in the Infostorage database. You cannot mask this field.

SOURCE

Indicates the name of the source or group record that limits the location where this user must log on. You must predefine valid source records to
CA ACF2 for VM entry lists under the type code SRC. If this field is null (or blank), CA ACF2 for VM does not check the source. You cannot mask this field.

SMSINFO(*recid*)

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

SRF

Indicates that this user can issue System Request Facility (SRF) requests to the CA ACF2 for VM service machine. These SRF requests can validate the accesses of other users and perform direct maintenance of the CA ACF2 for VM databases. To fully utilize SRF, the user must also have an @SRF definition in the ACFFDR. This is a bit field.

STC

A logonid with this attribute cannot log onto the VM system, but can be autologged if the AUTOONLY attribute is turned on for the logonid.

SUBAUTH

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

SUSPEND

Indicates that this logonid is suspended. This user cannot access the system. This is a bit field.

SYNCNODE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

SYNERR

Indicates how CA ACF2 for VM responds when a user with this field enters a command that results in a command syntax error. Values for this field are PREVENT (CA ACF2 for VM rejects the command), PREVENT-LOG (CA ACF2 for VM rejects and logs the command), LOG (CA ACF2 for VM passes the command to CP for normal syntax checking and generates a logging record), ALLOW (CA ACF2 for VM passes the command to CP for normal syntax checking). If this field is null, error processing is passed to the command model COMMAND clause, then to the CMDLIM VMO record. The default value is a null.

SYSPEXCL

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TAPE-BLP

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TAPE-LBL

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TDISKVLD

Indicates that access rules must exist for all data on temporary disks that this user accesses. This is a bit field. TDISKVLD is a method that lets you control which files a user can write to or read from his own T-disks to create a "padded cell" environment. For TDISKVLD to be effective, a user cannot change his own access rule. A special access rule syntax is required for files on T-disks. See the chapter “About access Rules” in this guide for more information.

TRACE

Traces and logs all data and resources this user references through access and resource validations. This is a bit field.

TSO

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSO-TRC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSOACCT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSOCMDS

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSOFSCRN

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSOPERF

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSOPROC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSORBA

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSORGN

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSOSIZE

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

TSOTIME

This field is not active for VM sites. For more information, see the*VM CA ACF2 for VM for z/OS Administrator Guid*e.

TSOUNIT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

UID

A pseudo-field concatenating selected information from the logonid record, including information from user-defined fields, such as company code, department, job function, and the logonid field. You cannot modify this field.

UIDSCOPE(*UID mask*)

Specifies a UID mask limiting logonid access. (24-byte character field)

Important! This is an old field left over from pre-3.1 release of CA ACF2 for VM. This field should not be used. To limit a user's administrative authority over the CA ACF2 for VM logonid, rule, and Infostorage databases, use the SCPLIST field in the logonid record with scope records to limit a user's administrative authority. For more information on how to create scope records, see the chapter "Processing Scope Records."

UNICNTR

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

UPD-TOD

Indicates the date and time that this logonid record was last updated.

USER

Indicates that this logonid belongs to a regular user. All logonids defined to CA ACF2 for VM are automatically users. This field is never displayed and no one should alter it. This is a bit field.

VLD-ACCT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

VLD-PROC

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

VLDRSTCT

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

VLDVMACT

Indicates that CA ACF2 for VM performs VM account validation under the LID account mode setting.

VM

Indicates that this user can log onto VM. A user must have this attribute to log on if the VMCHK field in the OPTS VMO record is set to VM. This field controls which systems a user can access in a multi-CPU environment when sharing databases. This is a bit field.

VMACCT

Indicates the default account number for a virtual machine.

VMD4AUTH

A user with this attribute can issue diagnose d4 to surrogate virtual machines with the VMD4TARG attribute. Use extreme caution when you assign this privilege. The VMD4TARG and VMD4AUTH privileges are very powerful. A typical class B user with both of these attributes could potentially surrogate itself to any user ID on the system and have access to anything on the system. In previous releases, this privilege was the VMBATMON privilege.

VMD4FSEC

This is the Diagnose D4 CMS File Level Security attribute. It indicates that CA ACF2 for VM should keep track of the surrogated id that is in use when minidisks are linked. This saved information is then used to validate CMS file accesses using the surrogated ID, even if the CMS file accesses are done after the surrogation is no longer in place. This only applies to CMS file accesses through standard CMS interfaces, not through services such as the *BLOCKIO System Service.

For example, FTPSERVE uses Diagnose D4 to link minidisks under the authority of the user requesting FTP services, but resets surrogation before actually transferring the CMS files. With the VMD4FSEC attribute on in the FTPSERVE logonid, CA ACF2 for VM will validate access to the CMS files on the minidisk using the authority of the user that was surrogated when the minidisk was linked.

VMD4RSET

Indicates that this user can be the target of the diagnose d4 reset after the logonid was surrogated to another ID. Use extreme caution when assigning this privilege. Never give this logonid attribute to a batch worker machine.

The combination of VMD4RSET, VMD4AUTH, and VMDTARG lets products like TCP/IP and VMBACKUP function properly. To track the use of the diagnose d4, you can write a diagnose limiting rule to log each time the diagnose d4 is issued. In previous releases of CA ACF2 for VM (releases 3.2 and below), this attribute was called VMRESET.

VMD4TARG

A user ID with this attribute can be the target of diagnose d4 (the alternate user diagnose). Use extreme caution when you assign this attribute. In previous releases, this privilege was the VMBATCH privilege.

VMESM

Indicates that this server can use the CA ACF2 for VM security interface.

VMIDLEMN

Specifies the number of minutes (from 1 to 240) that this user can be idle on the system before idle terminal processing begins. This value overrides the system-wide IDLEMN value defined in the OPTS VMO record.

VMIDLEOP

Specifies the type of idle terminal processing to perform when this user exceeds the idle time limit. This value overrides the system-wide IDLEOP value defined in the OPTS VMO record. Values for this field are:

OFF

Disables idle terminal processing for this user

DISC

Forces disconnection from the system when this user exceeds the idle terminal limit

LOGOFF

Forces this user off the system when he exceeds the idle terminal limit

NOLOGOFF

Prompts the user for his password when he exceeds the idle terminal limit. Incorrect passwords are counted as a password violation. The user can also disconnect from the system at this prompt. Similar to REPROMPT, but the option to logoff is not allowed.

REPROMPT

Prompts the user for his password when he exceeds the idle terminal limit. Incorrect passwords are counted as password violations. The user can also logoff or disconnect from the system at this prompt.

VMSAF

Indicates that this logonid can use VM interfaces to validate CA ACF2 for VM passwords. Supported interfaces are diagnose code x'A0' subfunction 4 and an APPC connect with password. This is a bit field.

Diagnose A0 subfunction 4 lets users validate passwords from their own unique applications.

APPC connect with password allows APPC/VM VTAM support (AVS) service machines to validate CA ACF2 for VM passwords.

VMSFS

Indicates that this SFS server can use the CA ACF2 for VM security interface.

VMXA

Indicates that this user can access the system if VMCHK (VMXA) is defined in the OPTS VMO record. During DIAL and GRPLOGON processing, CA ACF2 for VM bypasses the VMCHK authorization. See the VM attribute above for more information.

VSESRF

Indicates that this logonid (for a CA ACF2 for VM VSE controlled system) can issue System Request Facility (SRF) requests to the service machine. SRF requests validate the accesses of users and perform direct maintenance of the CA ACF2 for VM databases. This field works with the ACFFDR @SRF definition.

WTP

This field is not active for VM sites. For more information, see the *CA ACF2 for VM for z/OS Administrator Guid*e.

ZONE

Indicates the time zone where this user normally accesses the system. You must have predefined time zones of three characters in the Infostorage database. If this field is null or blank, CA ACF2 for VM does not check the zone.

Masking Logonids

Masking lets you specify multiple logonids at one time. For example, the logonid mask PAY*** represents all logonids that begin with the letters PAY and end with any zero to three characters.

You can use masking to process multiple logonid records with a single ACF subcommand. You can also use logonid masks to process other CA ACF2 for VM records and to create CA ACF2 for VM rules.

There are two different masking symbols that you can use in logonids, the asterisk and the dash.

Asterisks

A logonid mask that contains asterisks represents all valid logonids that begin with the specified characters and end with any number of characters that do not exceed the number of asterisks. These characters can include blanks.

Mask	Matches	Does Not Match
PAY***	PAY PAYDS PAYDSJ	PA PBYKL PAYKLST

Dashes

A logonid mask that contains a dash represents all valid logonids that begin with the specified characters. The total length of a valid logonid must be eight or less characters.

Mask	Matches	Does Not Match
PAY-	PAY PAYDS PAYDSJ PAYDS12J	PA PBYKL PAVKLSTS

Chapter 7: Maintaining Logonids with the Full-Screen Feature

This chapter explains how you can maintain logonid records through the CA ACF2 for VM full-screen feature.

After you finish this chapter, you will know how to:

- Define a new user to CA ACF2 for VM
- Display logonid information
- Change a logonid
- Delete a logonid from the system.

We show the screens in this chapter as we ship them with CA ACF2 for VM. You can customize, delete, or reorder the screens as your site requires. (See the *Installation Guide* for detailed instructions on how to modify your screens.) The screen display can also vary depending on your CA ACF2 for VM privileges. The LIST and ALTER specifications of the @CFDE macro in the ACFFDR govern the fields.

This section contains the following topics:

Processing Logonids

To begin processing logonid records, enter option 1, User Identification, from the Primary Option Menu. CA ACF2 for VM displays the User Identification screen shown below.

```
M9PA-1000      User Identification (1)               CA ACF2 for VM
OPTION ===> ____________________________________________________
                                                        TIME 16:57
      0 Change options or PF keys

      1 Define New User

      2 Display User Information

      3 Change User Information

      4 Delete a User

      Option may be followed by ".userid" to select
      the user to which the option is to apply.




PF1=Help       2=Print       3=Quit       4=Return      5=        6=
PF7=           8=            9=           10=           11=       12=Retrieve
```

The options shown on this screen are defined below.

0 Change options or PF keys

Displays current values of PF keys for logonid maintenance and lets you change them.

1 Define New User

Lets you create logonids for new users.

2 Display User Information

Lets you see logonid record information for a specified user.

3 Change User Information

Lets you change logonid record information if you have the necessary privileges in your logonid record.

4 Delete a User

Lets you remove a logonid from the Logonid database.

Changing Options or PF Keys

You see this screen if you selected option 0, Change Options or PF keys, from the User Identification (1) screen. Use this screen to change values of your PF keys for **logonids** only.

```
M9PA-0010     Change Options or PFkeys                    <eacf2>
OPTION ===> ____________________________________________________________
                                                          TIME 17:12
 Options:
  Yes ==> Y        Time (12/24) ==> _
  No  ==> N        Language   ==> _____

             Description   Command
  PF1  ==>  Help          HELP
  PF2  ==>  Print         PRINT
  PF3  ==>  Quit          QUIT
  PF4  ==>  Return        RETURN
  PF5  ==>  Execute       EXECUTE
  PF6  ==>  MVS<->VM      MVSVM
  PF7  ==>  Backward      BACKWARD
  PF8  ==>  Forward       FORWARD
  PF9  ==>  Director      DIRECTOR
  PF10 ==>  Previous      PREVIOUS
  PF11 ==>  Next          NEXT
  PF12 ==>  Retrieve      RETRIEVE


PF1=Help       2=Print       3=Quit       4=Return     5=        6=
PF7=           8=            9=           10=Save      11=       12=Retrieve
```

Following is a brief description of the fields on this screen.

Yes

Indicates how you want yes values displayed. Enter 1 (to indicate binary) or Y (to indicate alpha) in this field.

No

Indicates how you want no values displayed. Enter 0 (to indicate binary) or N (to indicate alpha) in this field.

Time(12/24)

Enter whether you want the time reported in a 12-hour (enter 12) or 24-hour (enter 24) clock.

Language

Enter the five-character code for the language you want to use for screens. CA supplies the following valid options:

AMENG

American English

UCENG

Upper Case English.

Your site may have added additional language support. Contact your security administrator or systems programmer for this information.

To change the values for PF keys, enter the value to be displayed on the screen for the PF key in the Description column. Enter the actual command names (in caps) in the Command column.

Defining a Logonid

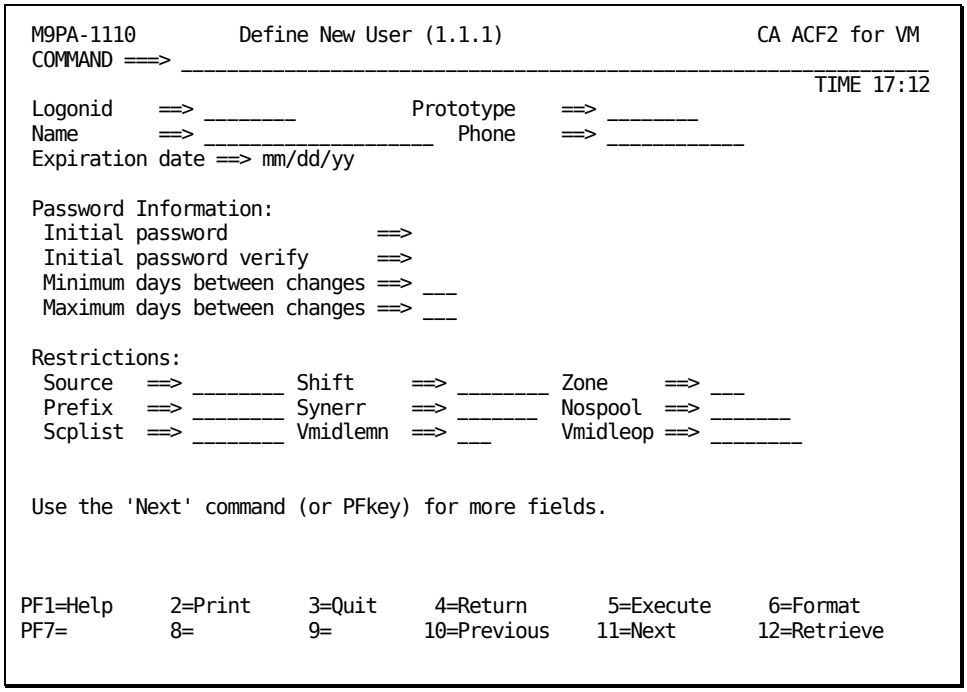
After you install and test CA ACF2 for VM at your site, you must define logonids to the system. Only logonids defined to CA ACF2 for VM can access the system. A site assigns a primary individual (usually an account manager) to perform this task. All logonid records defined to CA ACF2 for VM reside on the Logonid database. This section explains how you can define logonids to CA ACF2 for VM through the full-screen feature.

When you finish this section, you will know how to:

- Create a logonid for a new user
- Assign privileges to the new user's logonid
- Assign privileges to a user that are unique to your site.

Defining a User

You see this screen if you selected option 1, Define New User, from the User Identification (1) screen. Use this screen to define new users to the system. In most cases, you can use a prototype to define users from this single screen. To use this short cut, press EXECUTE (or the PF key that you have defined to execute this screen) after you have filled this screen in completely.



Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid of the user.

Name

Specifies the name of the user.

Expiration Date

Specifies the date that this logonid expires. This date must be in the format your site defined in the OPTS VMO record (mm/dd/yy, dd/mm/yy, or yy/mm/dd).

Prototype

Enter the logonid to use as a model for this logonid. This option provides a short cut in creating logonids. Use a logonid that has privileges similar to the one you want to create. When you type a logonid in this field and press Enter, CA ACF2 for VM retrieves the prototype data for you. CA ACF2 for VM never copies the PASSWORD and SECURITY fields.

Phone

Specifies the user's telephone number.

Initial Password

Enter the initial password (up to eight characters).

Initial Password Verify

Enter the required password just as you did in the Initial Password ===> field. CA ACF2 for VM asks you to reenter the password to be sure that no typing error occurred the first time you entered it. If you do not enter the password exactly the same both times, CA ACF2 for VM asks you to enter the password twice again. CA ACF2 for VM never displays either password field for security reasons. If anyone is near your terminal when you enter your password, he cannot see it.

Minimum days between changes

Specifies the number of days (up to 255) that must elapse before you can change your password (MINDAYS).

Maximum days between changes

Specifies the number of days (up to 255) that can elapse before you are forced to change your password (MAXDAYS).

Restrictions Fields

For information about these restrictions, see the “About the Logonid Record” chapter. If you are not authorized to grant any of these restrictions, CA ACF2 for VM protects the fields and you cannot enter data in them.

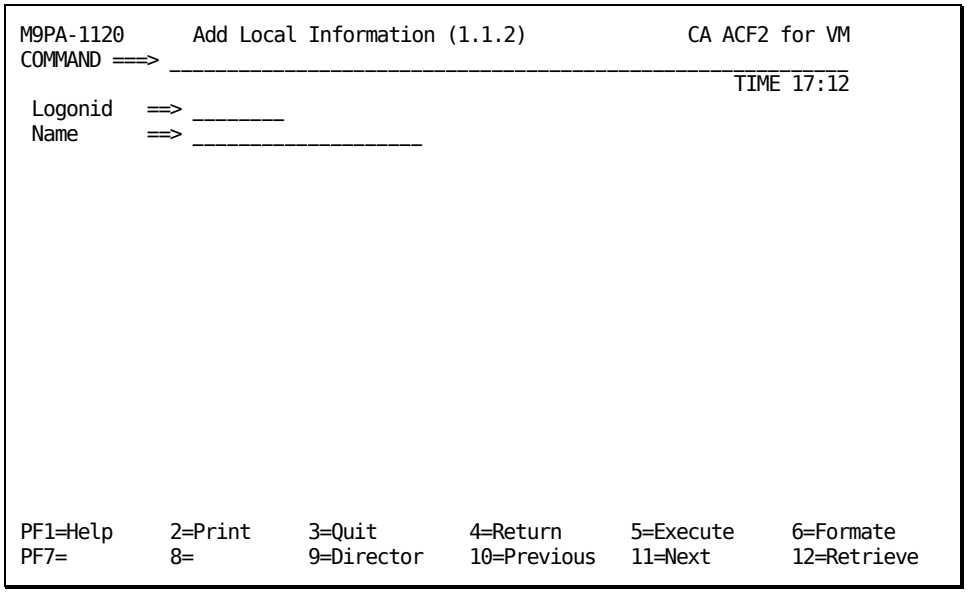
When you have completed all screens defining the user to the Logonid database, press your EXECUTE PF key to process the information. CA ACF2 for VM displays the following message in line three of the screen when you have successfully added a new user to the Logonid database:

```
ACFpgm710I LOGONID lid INSERTED
```

To add user privileges or local information, press PF11 (NEXT) and CA ACF2 for VM displays the next screen.

Adding Local Information

To see this screen, press PF11 after you have viewed the Define a User screen. Your site formats this screen. Contact your local systems programmer for details.



Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid of the user

Name

Specifies the name of the user.

Adding User Privileges

To see this screen, press PF11 after you have viewed the Add Local Information screen. Use this screen to define privileges for the new user.

```
M9PA-1130        Add User Privileges (1.1.3)              CA ACF2 for VM
COMMAND ===> _____________________________________________________________
                                                                TIME 15:39
 Logonid    ==> ________
 Name       ==> ____________________

 Privileges: (Enter "y" or "1" to select)

ACCOUNT   ==> _ AUDIT    ==> _ AUTOALL  ==> _ AUTONOPW ==> _ AUTOONLY ==> _
CANCEL    ==> _ CICS     ==> _ CONSULT  ==> _ DIALBYP  ==> _ DUMPAUTH ==> _
GRPLOGON  ==> _ LEADER   ==> _ LOGSHIFT ==> _ MUSASS   ==> _ NO-STORE ==> _
NON-CNCL  ==> _ PSWD-EXP ==> _ READALL  ==> _ RULEVLD  ==> _ SECURITY ==> _
SRF       ==> _ SUSPEND  ==> _ TDISKVLD ==> _ TRACE    ==> _ VLDVMACT ==> _
VM        ==> _ VMD4AUTH ==> _ VMD4RSET ==> _ VMD4TARG ==> _ VMXA     ==> _
VSESRF    ==> _




PF1=Help     2=Print     3=Quit       4=Return     5=Execute    6=Format
PF7=         8=          9=Director   10=Previous   11=Next      12=Retrieve
```

Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid of the user.

Name

Specifies the name of the user.

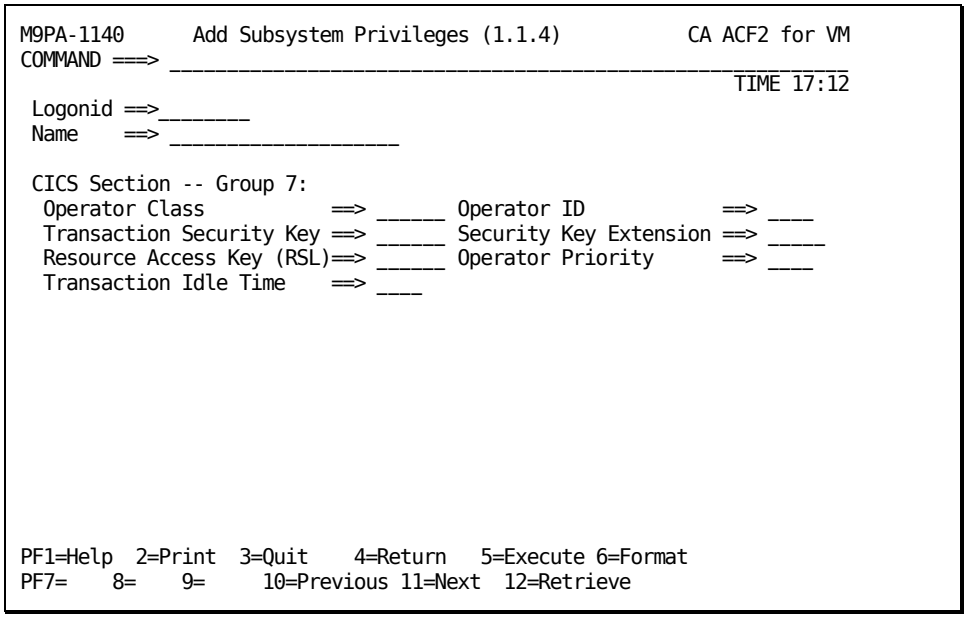
Privileges Fields

To define privileges for the user, enter Y or 1 next to the privileges to be allowed. To enter the definitions into the Logonid database, press your EXECUTE PF key.

For information about these privileges, see “About the Logonid Record”. CA ACF2 for VM does not display privileges that you are not authorized to grant. If the user has been given privileges that you are not authorized to grant, contact your security administrator.

Adding Subsystem Privileges

To see this screen, press PF11 after you have viewed the Add User Privileges screen. Use this screen to add subsystem privileges.



Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid of the user.

Name

Specifies the name of the user.

CICS Section Fields

For information about these privileges, see “About the Logonid Record."

Displaying a Logonid

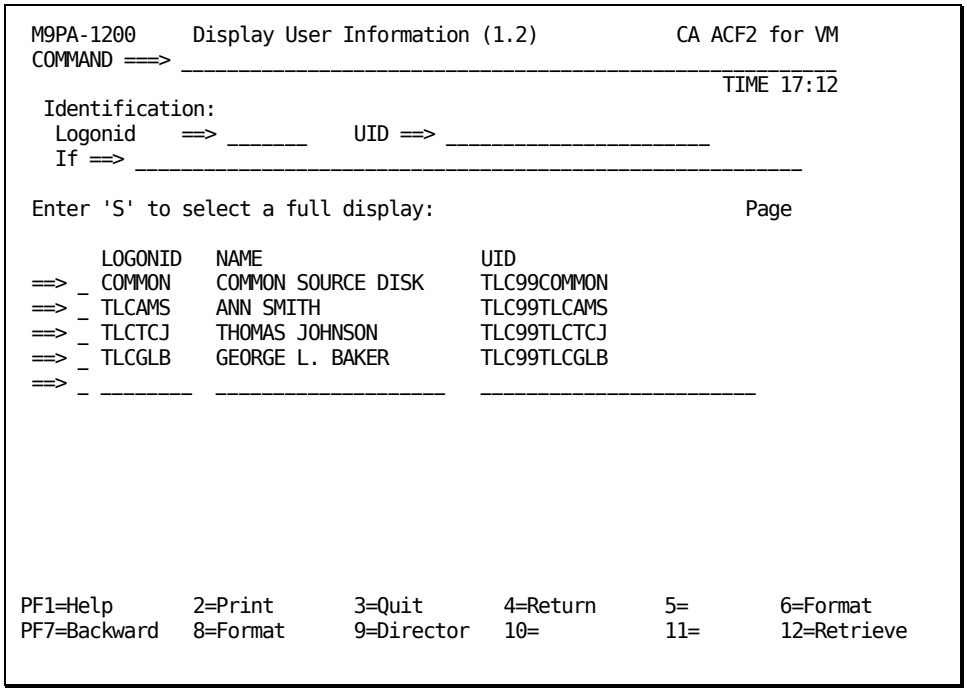
You can use the screens illustrated in the following sections to display logonid information of other users. You can display this information only if you have the necessary privileges defined in your logonid record. If your logonid has no special privileges assigned, you can only display information about your own logonid.

When you finish this section, you will know how to display:

- Information about your logonid (and those of other users if your logonid has special privileges)
- The user and UID associated with a logonid
- Restrictions imposed on a logonid
- Information that is unique to your site about a logonid
- Logonid privileges assigned to a specific user
- Any subsystem privileges assigned to a specific user
- Access information for a specific user, such as when he last accessed the system and where he made the access.

Displaying User Information

You see this screen if you selected option 2, Display User Information from the User Identification (1) screen. You can also use the fast path method to bypass intermediate screens. Fast path is a shortcut that lets you select options from the Primary Option Menu screen and proceed directly to the screen you choose. In this example, you can select 1.2 from the Primary Option Menu and go directly to the Display User Information screen. Use this screen to display logonid information of other users. The selection criteria for this search is the logonid or the UID.



Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid of the user.

UID

Specifies the user identification string of the user you want information about.

Enter the logonid or UID of the user you want information about. You can mask fields on this screen. If you do use masking on this screen, CA ACF2 for VM shows you a list of logonids that match the specified mask. To search for specific privileges, specify a value in the If field. If you use the If field, CA ACF2 for VM displays a list of only those logonids that match the If criteria. To see more information about any of these logonids, enter an S before the logonid. CA ACF2 for VM shows you detailed information about each logonid on another screen.

Displaying User Identification

When you select a logonid from the Display User Information screen shown on the previous page and press Enter, you see the Display User Identification screen. This screen shows you more detailed information about the logonid you selected on the previous screen.

```
M9PA-1210      Display User Identification (1.2.1)        CA ACF2 for VM
COMMAND ===>
_____________________________________________________________________________
                                                                    TIME 17:12
Identification:
 Logonid          : ________
 Name             : ____________________  Phone  : ____________
 Expiration Date  : ________

Password Information:
 Expired          : _                Violations      : _____
 Minumum days     : ___              Maximum days    : ___
 Date - time set  : _____________    Violation date  : 00/00/00

Restrictions:
 Source    : ________ Shift     : ________ Zone    : ___
 Prefix    : ________ Synerr    : ________ Nospool : _______
 Scplist   : ________ Vmidlemn  : ___      Vmidleop  : ________




PF1=Help       2=Print       3=Quit       4=Return      5=        6=
PF7=           8=            9=           10=Previous   11=Next   12=Retrieve
```

The fields on this screen provide three types of information:

- Identification

- Password information
- Restrictions.

Identification Fields

Logonid

Specifies the logonid of the user.

Name

Specifies the name of the user.

Phone

Specifies the user's telephone number.

Expiration Date

Specifies the date that this logonid expires. This date is displayed in the format your site defined in the OPTS VMO record (mm/dd/yy, dd/mm/yy, or yy/mm/dd).

Password Information Fields

Expired

Indicates the date the password expired.

Violations

Indicates the number of password violations for this logonid.

Minimum days

Indicates the number of days (up to 255) that must elapse before this user can change his password.

Maximum days

Indicates the number of days (up to 255) that must elapse before this user is forced to change his password.

Date-time set

Indicates the date and time this user last changed his password.

Violation date

Indicates the date the last password violation occurred.

Restriction Fields

For information about these fields, see “About the Logonid Record."

Displaying Local Information

To see this screen, press PF11 after you have viewed the Display User Identification screen. Your site determines what information appears on this screen. Contact your systems programmer for details.

```
M9PA-1220     Display Local Information (1.2.2)          CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 17:12
 Logonid  : ________
 Name     : ____________________









PF1=Help       2=Print       3=Quit       4=Return      5=        6=
PF7=           8=            9=Director   10=Previous   11=Next   12=Retrieve
```

Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid of the user.

Name

Specifies the user's name.

Displaying User Privileges

To see this screen, press PF11 after viewing the Display Local Information screen. This screen displays all the privileges that are granted to a specified user, such as whether he has the AUTOLOG or SECURITY privilege.

```
M9PA-1230      Display User Privileges (1.2.3)          CA ACF2 for VM
COMMAND ===> _____________________________________________________________
                                                              TIME 17:09
 Logonid  : ________
 Name     : ____________________________

 Cancel/Suspend:
  By Userid  : ________    On Date  : ______

 Privileges:

ACCOUNT   : _ AUDIT    : _ AUTOALL   : _ AUTONOPW : _ AUTOONLY : _
CANCEL    : _ CICS     : _ CONSULT   : _ DIALBYP  : _ DUMPAUTH : _
GRPLOGON  : _ LEADER   : _ LOGSHIFT  : _ MUSASS   : _ NO-STORE : _
NON-CNCL  : _ PSWD-EXP : _ READALL   : _ RULEVLD  : _ SECURITY : _
SRF       : _ SUSPEND  : _ TDISKVLD  : _ TRACE    : _ VLDVMACT : _
VM        : _ VMD4AUTH : _ VMD4RSET  : _ VMD4TARG : _ VMXA     : _
VSESRF    : _



PF1=Help       2=Print       3=Quit       4=Return      5=        6=
PF7=           8=            9=Director   10=Previous   11=Next   12=Retrieve
```

Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid of the user.

Name

Specifies the name of the user.

Cancel/Suspend Fields

By Userid

Specifies the logonid of the user that turned on the CANCEL or SUSPEND bits for this user (eight characters).

On Date

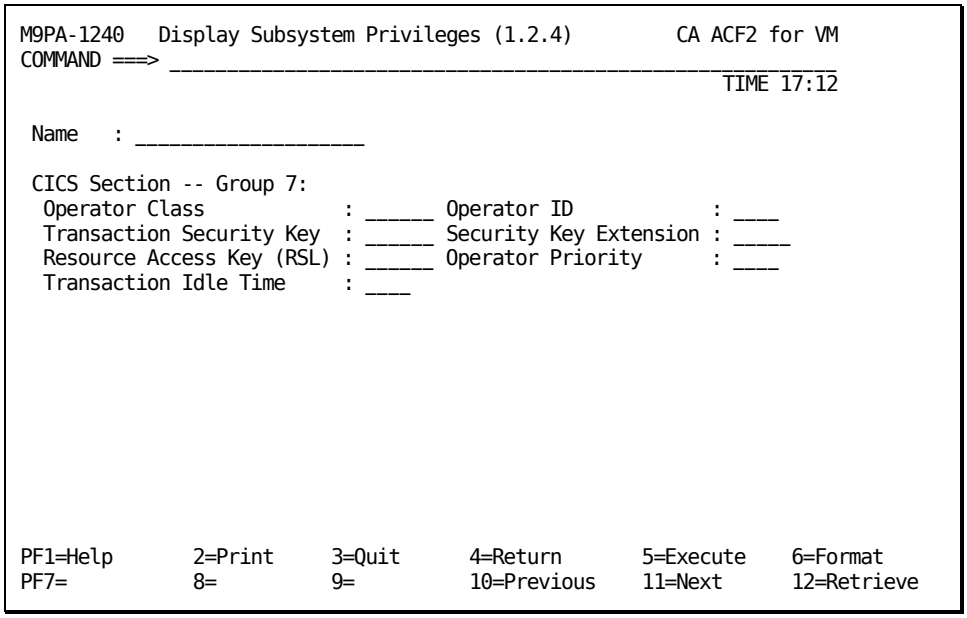
Specifies the date that the CANCEL or SUSPEND field was set for this user (mm/dd/yy, dd/mm/yy, or yy/mm/dd, depending on the OPTS VMO record).

Privileges Fields

For information about these privileges, see “About the Logonid Record."

Displaying Subsystem Privileges

To see this screen, press PF11 after viewing the Display User Privileges screen. This screen displays subsystem privileges for the specified user.



Following is a brief explanation of the fields on this screen.

Name

Specifies the name of the user.

CICS Section Fields

For information about these fields, see the appropriate CA ACF2 for VM for z/OS and OS/390 subsystem guide, such as the *CICS Support Guide* or the *IMS Support Guide*.

Displaying Access Information

To see this screen, press PF11 after viewing the Display Subsystem Privileges screen.

This screen displays information on accesses a specific logonid made, such as the date the user last accessed the system and where the user made the access.

```
M9PA-1250     Display Access Information (1.2.5)          CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 17:12
 Logonid   : ________
 Name      : ____________________

 Access Information:
  Count       : ____
  Last date   : ____
  Last time   : ____
  From source : ________
  Group user  : ______

 Statistics:
  Security Violations : ___
  LID Last Update TOD : ________




PF1=Help       2=Print       3=Quit       4=Return      5=        6=
PF7=           8=            9=Director   10=Previous   11=Next   12=Retrieve
```

Below is a brief explanation of the information shown above.

Logonid

Specifies the logonid of the user.

Name

Specifies the name of the user.

Access Information Fields

Count

Specifies the number of accesses this user made.

Last date

Specifies the date this user last accessed the system.

Last time

Specifies the time this user last accessed the system.

From source

Specifies the physical location of this user's last access.

Group user

Corresponds to the GRP-USER attribute in the logonid record. It indicates which users can log onto a group ID using another logonid and password and displays the logonid of the last user to access the group machine. Each time another user logs on, the previous logonid is erased.

Statistics Fields

Security violations

Specifies the number of security violations this user incurred to date.

LID last update TOD

Specifies the date and time when a field in the logonid record was changed. (This changes every time you alter a field.)

Changing a Logonid

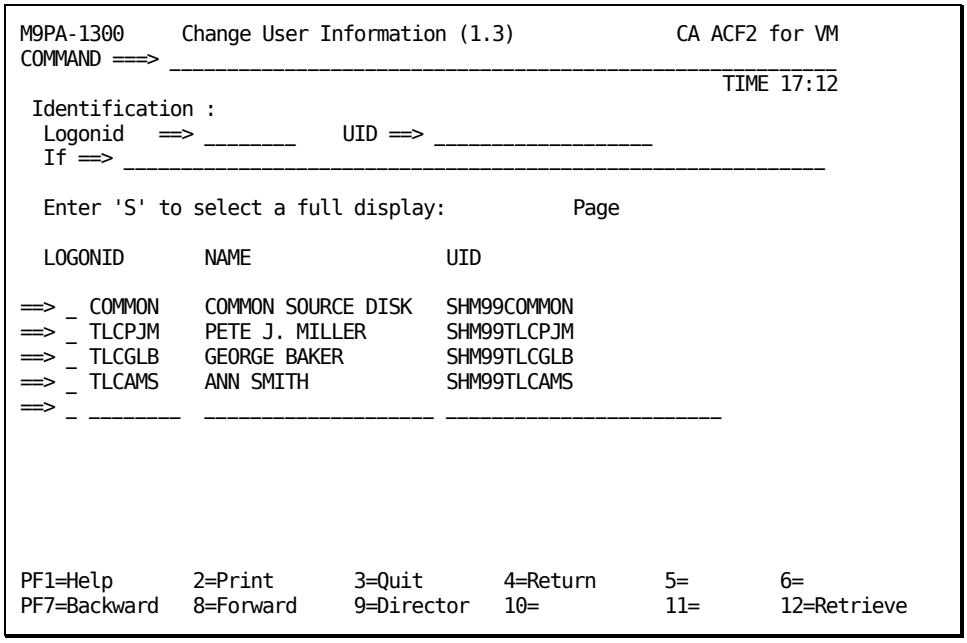
This section explains how to change information about specific users. The privileges assigned to your logonid determine the type of information you can change. Your privileges can also limit the users you can change information for. If you are not authorized to make necessary changes, contact your security administrator. If you are the security administrator, contact your systems programmer for this information.

When you finish this section, you will know how to:

- Change password information or restrictions for a specific logonid
- Change logonid information unique to your site
- Change logonid privileges for a specific user
- Cancel or suspend a logonid
- Specify the subsystems a user can use.

Changing User Information

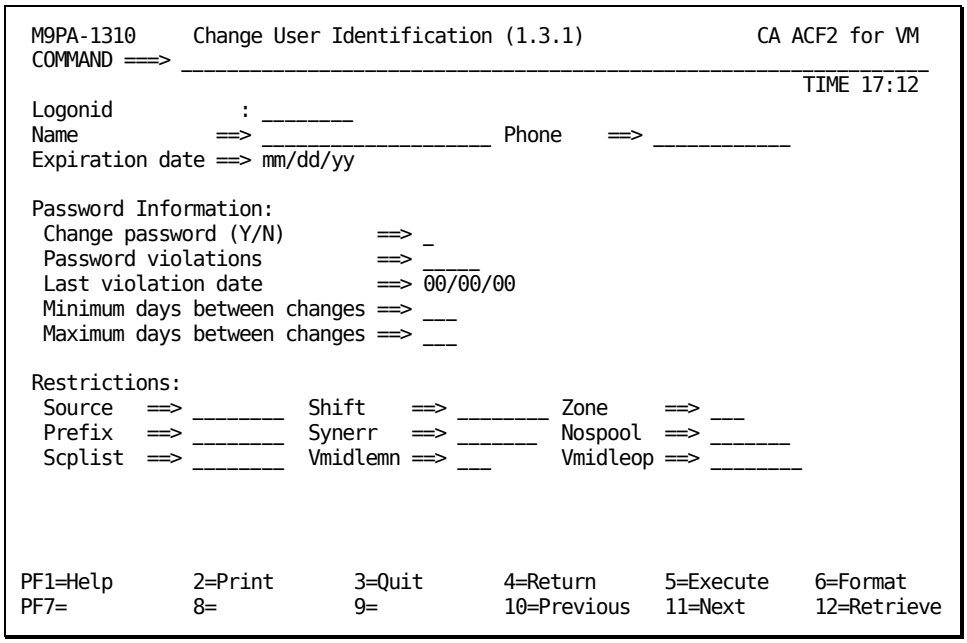
You see this screen if you selected option 3, Change User Information, from the User Identification (1) screen. You can also use the fast path method and select 1.3 from the Primary Option Menu to go directly to this screen. Use this screen to change information for a specified user.



Enter the logonid or UID of the user whose logonid information you want to change. You can mask the logonid and UID. To change logonids with specific privileges, specify a value in the IF field. In other words, if you specify a logonid, or UID, and IF statement, CA ACF2 for VM displays only those logonids that match both criteria. If the logonid is not masked, CA ACF2 for VM ignores all UID and IF arguments. Press Enter and CA ACF2 for VM displays another screen where you can change values associated with this logonid.

Changing a User's Identification

To see this screen, press PF11 after you have viewed the Change User Information screen. Use this screen to change a user's identification information, password, or restrictions.



The information displayed was picked up from the logonid you entered on the Change User Information screen. You can type over the values on the screen to change any of the displayed information. If the change you make is shorter than the old value, erase the rest of the field.

Following is a brief explanation of the fields on this screen.

Logonid

Specifies the user's logonid.

Name

Specifies the name of the user.

Phone

Specifies the user's telephone number.

Expiration Date

Specifies the date that this logonid expires. This date must be in the format your site defined in the OPTS VMO record (mm/dd/yy, dd/mm/yy, or yy/mm/dd).

Password Information Field

Change password (Y/N)

For security reasons, CA ACF2 for VM never displays this field value. To give the user a new password, type Y. CA ACF2 for VM prompts you for a new password value.

Password violations

Displays the number of password violations for this logonid. You can change this number if the user forgot his password and could not access the system because he reached the password violation limit defined in the PSWD VMO record.

Last violation date

Specifies the date the last password violation occurred.

Minimum days between changes

Specifies the number of days (up to 255) that must elapse before this user can change his password (same as MINDAYS).

Maximum days between changes

The number of days (up to 255) that must elapse before this user can change his password (same as MAXDAYS).

Restrictions Fields

For information about these privileges, see “About the Logonid Record.".

To make further changes to this user's privileges, press PF11 (NEXT) and press Enter. You see another screen you can use for input. When you have completed all the screens necessary to change the user's logonid record, press your EXECUTE PF key to change the information in the user's logonid record.

CA ACF2 for VM displays the following message in line three of the screen when you have changed a logonid:

```
ACFpgm704I LOGONID lid CHANGED
```

Changing a User's Local Information

To see this screen, press PF11 after viewing the Change User Identification screen. Your site formats this screen.

```
M9PA-1320      Change Local Information (1.3.2)          CA ACF2 for VM
COMMAND ===> _____________________________________________________________
                                                              TIME 17:12
 Logonid     : ________
 Name     ==> ____________________




PF1=Help       2=Print    3=Quit       4=Return       5=Execute    6=Format
PF7=           8=         9=Director   10=Previous     11=Next      12=Retrieve
```

To change the user's information, complete all the screens necessary and press your EXECUTE PF key. Your changes are sent to the user's logonid record.

Changing a User's Privileges

To see this screen, press PF11 after viewing the Change Local Information screen. Use this screen to change existing privileges for the logonid defined on the Change User Identification screen.

```
 M9PA-1330      Change User Privileges (1.3.3)          CA ACF2 for VM
COMMAND ===> _____________________________________________________________
                                                              TIME 11:29
 Logonid  : ________
 Name   ==> ____________________

 Cancel/Suspend:
    By Userid  :         On Date  :

 Privileges: (Enter "y" or "1" to select)

 ACCOUNT  ==> _ AUDIT    ==> _ AUTOALL  ==> _ AUTONOPW ==> _ AUTOONLY ==> _
 CANCEL   ==> _ CICS     ==> _ CONSULT  ==> _ DIALBYP  ==> _ DUMPAUTH ==> _
 GRPLOGON ==> _ LEADER   ==> _ LOGSHIFT ==> _ MUSASS   ==> _ NO-STORE ==> _
 NON-CNCL ==> _ PSWD-EXP ==> _ READALL  ==> _ RULEVLD  ==> _ SECURITY ==> _
 SRF      ==> _ SUSPEND  ==> _ TDISKVLD ==> _ TRACE    ==> _ VLDVMACT ==> _
 VM       ==> _ VMD4AUTH ==> _ VMD4RSET ==> _ VMD4TARG ==> _ VMXA     ==> _
 VSESRF   ==> _


PF1=Help       2=Print    3=Quit       4=Return     5=Execute    6=Format
PF7=           8=         9=Director   10=Previous   11=Next      12=Retrieve
```

Following is a brief explanation for the fields on this screen.

Logonid

Specifies the logonid of the user.

Name

Specifies the name of the user. (You can change this information.)

Cancel/Suspend Fields

By Userid

Displays the logonid of the user that set the CANCEL, SUSPEND, or MONITOR field for this user (eight characters).

On Date

Displays the date the CANCEL or SUSPEND field that was set for this user (mm/dd/yy, dd/mm/yy, or yy/mm/dd, depending on the OPTS VMO record).

Fields you can change are indicated with an arrow (==>) after the privilege name. You are not authorized to change the fields where the privilege name ends with a colon (:).

To define privileges for a user, enter Y or 1 for those privileges you are adding for this user. To remove a privilege, enter N or 0 after the privilege name. You cannot grant privileges that you are not authorized to grant. If the user needs privileges that you do not have authorization to grant, contact your security administrator. If you are the security administrator, contact your systems programmer for further instructions.

Privileges Fields

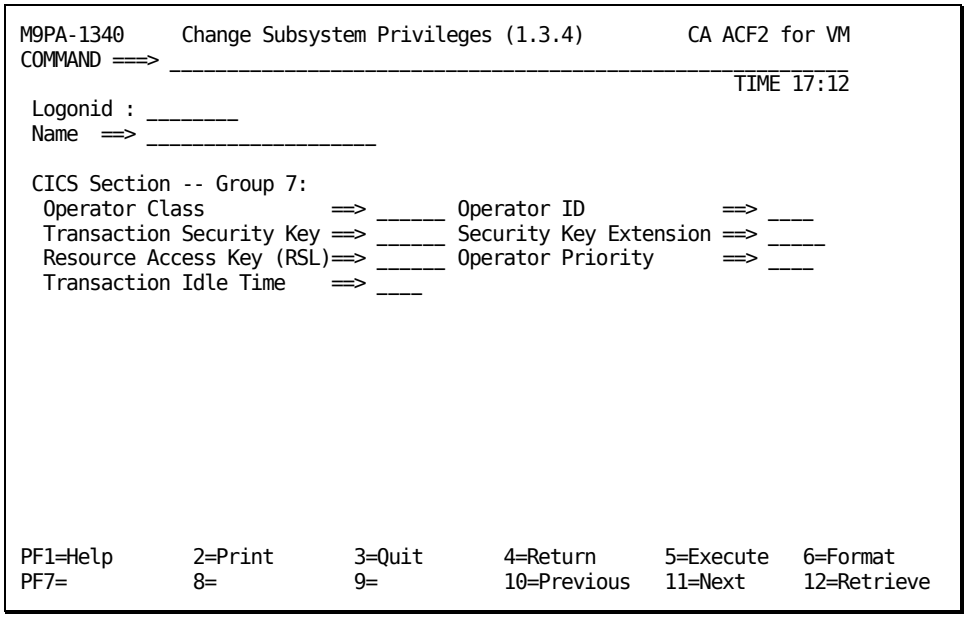
For information about these privileges, see “About the Logonid Record."

When you have completed all the screens necessary to change the user's privileges, press your designated EXECUTE PF key to change these privileges in the user's logonid record. CA ACF2 for VM displays the following message in line three of the screen when you have changed the user's logonid privileges:

```
ACFpgm704I LOGONID lid CHANGED
```

Changing Subsystem Privileges

To see this screen, press PF11 after viewing the Change User Privileges screen. Use this screen to change the subsystems a user can use.



Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid of the user.

Name

Specifies the name of the user.

CICS Section Fields

For information about these privileges, see “About the Logonid Record.”

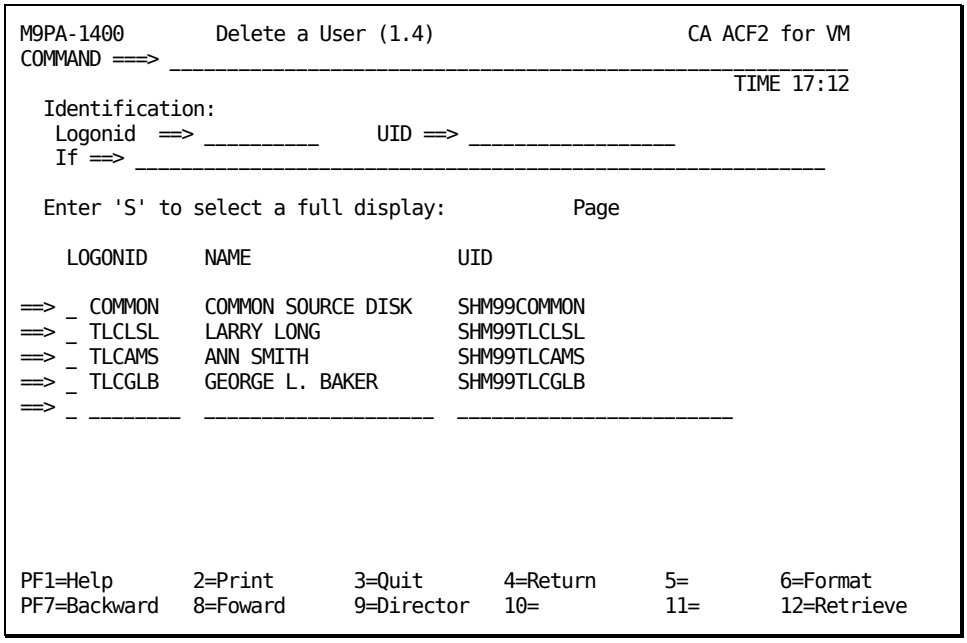
Deleting a Logonid

This section explains how to delete logonid records. Be very careful when deleting logonid records. If you delete one accidentally, you must redefine it.

When you have finished this section, you will be able to use the full-screen feature to delete a logonid record.

Deleting a User

You see this screen if you selected option 4, Delete a User, from the User Identification (1) screen. You can also select 1.4 from the Primary Option Menu and go directly to this screen. Use this screen to delete a user's logonid from the system. You must have the necessary authority to delete a logonid.



Following is a brief explanation of the fields on this screen.

Logonid

Specifies the logonid to delete.

UID

Specifies the UID of the user to delete.

If

Enter search criteria for a list of users to delete. For information about valid fields, see “About the Logonid Record.”

If you use masking or the If statements, CA ACF2 for VM displays a list of users that match the specified mask or search criteria. Enter an S (for SELECT) before the logonid of the user you want to delete. Press Enter and you see another screen that requires confirmation before CA ACF2 for VM deletes the logonid.

Confirming a User Deletion

To see this screen, press Enter after you have completed the Delete a User screen. This screen asks you to be sure that you really want to delete the specified logonid from the Logonid database. After you delete it, it is gone from the database.

```
M9PA-1410     Delete User Verification (1.4.1)          CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 17:12
 Identification:
  Logonid         : ________________
  Name            : ____________________  Phone : _____________
  Expiration Date : ________

 Password Information:
  Expired:      : __      Violations          : ____
  Minimum days  : ____     Maximum days       : ____
  Date/Time Set : ___________ Violation Date : 00/00/00

Delete Users Access Rule   ===> Y

Y to delete associated Resource Rules   N to bypass Deletion
  AUTOLOG  ALG ==> Y  DIAL  DIA ==> Y  GRPLOGON  GRP ==> Y
  IUCV     IUC ==> Y  VMCF  VMC ==> Y



PF1=Help       2=Print     3=Quit        4=Return      5=Execute    6=
PF7=           8=          9=Director    10=Previous   11=Next      12=Retrieve
```

To delete a logonid, carefully check the information displayed. If it is correct, press your designated EXECUTE PF key to delete the logonid.

Identification Fields

Logonid

Specifies the logonid to be deleted.

Name

Specifies the name of the user whose logonid to be deleted.

Phone

Specifies the user's telephone number.

Expiration Date

Specifies the date this logonid expires. This date is in the format your site defined in the OPTS VMO record (mm/dd/yy, dd/mm/yy, or yy/mm/dd).

Password Information Fields

Expired

Indicates if the password has expired.

Violations

Indicates the number of password violations this logonid has to date.

Minimum days

Indicates the number of days (up to 255) that must elapse before this user can change his password.

Maximum days

Indicates the number of days (up to 255) that must elapse before this user must change his password.

Date-time set

Indicates the date and time this password was last changed.

Violation date

Indicates the date the last password violation occurred.

CA ACF2 for VM displays the following message in line three of the screen when you have deleted the logonid and its associated rules:

```
ACFpgm714I LOGONID lid DELETED
```

Delete Users Access Rule Field

This field is where you indicate whether to delete any access rules associated with the specific logonid you are deleting. Specify a Y or 1 in this field to delete this user's access rules with his logonid. Specify an N or 0 to retain any existing access rules. You can enter only one character in this field. Valid values are 1, Y, y, 0, N, or n. The default is Y (for Yes). If no rules exist, this field defaults to No, and you cannot modify the field.

Y to Delete Associated Resource Rules N to Bypass Deletion

This field is where you indicate whether to delete any resource rules associated with the specific logonid you are deleting. As with access rules, specify a Y or 1 in this field to delete this user's resource rules with his logonid. Specify an N or 0 to retain any existing resource rules. You can enter only one character in this field. Valid values are 1, Y, y, 0, N, or n. The default is Y (for Yes). If no rules exist, this field defaults to No, and you cannot modify the field.

Chapter 8: Maintaining Logonids with the ACF Command

This chapter contains information on how to process logonids with the ACF command. The ACF command lets you process logonids through line commands instead of full-screen panels.

When you finish this chapter, you will know how to use the ACF command and its subcommands to:

- Display user information
- Add a logonid to the Logonid database
- Change logonid information
- Change the PASSWORD field
- Remove a logonid from the database.

However, before you can process logonids, you must have the necessary CA ACF2 for VM privileges defined in your logonid record. The information on the following page describes these privileges.

This section contains the following topics:

Required CA ACF2 for VM Privileges

ACCOUNT, LEADER, and SECURITY users can change selected logonid records in the limits specified through their SCPLIST field. A user is unrestricted if he has no SCPLIST field specified in his logonid record.

Here is some important information about security privileges and scope lists:

ACCOUNT

This user can insert, list, delete, and change logonid records in his SCPLIST. An ACCOUNT user without the SECURITY privilege cannot change or list the logonid record of a user with both the ACCOUNT and SECURITY privileges. A user with both of these privileges is more powerful than a user with only one. An ACCOUNT user normally maintains logonid records, but has no authority to write access rules.

LEADER

This user can change selected fields of logonid records in his SCPLIST. This differs from the ACCOUNT privilege that lets a user change any logonid record field in his scope.

SECURITY

This user is a security administrator. He can list and change fields of logonid records for users in his SCPLIST. The security administrator cannot create (INSERT) new logonid records unless he also has the ACCOUNT privilege. An ACCOUNT user is responsible for maintaining logonid records, while a SECURITY user is responsible for writing and maintaining access rules and all records residing on the Infostorage database.

SCPLIST

This field limits the logonid records a privileged user (a user with ACCOUNT, LEADER, or SECURITY) can access. For example, the Identification and Privileges portions of a logonid record for Ann Smith are:

```
TLCAMS        AAPMGRGTLCAMS ANN SMITH EXT.458
PRIVILEGES    ACCOUNT SCPLIST(ACCTMGR)
```

Ann has the ACCOUNT privilege so she can process logonid records.The SCPLIST field in her logonid record indicates that she is limited to creating and maintaining only the logonid records defined in the scope record named ACCTMGR on the Infostorage database.

Beginning Logonid Record Processing

Before you can begin to process logonids, you must issue the appropriate ACF subcommand to tell CA ACF2 for VM what kind of function you want it to perform.

To begin logonid record processing, enter the following:

```
SEt Lid
```

CA ACF2 for VM responds with LID to indicate that the system is ready for you to begin logonid record processing.

Using ACF LIST to Display User Information

You can display logonid information with the ACF LIST subcommand. If you have the correct privileges (specified in the SCPLIST field of your logonid record), you can display other logonid records. You can always display your own logonid record. The following syntax shows how you can display logonid information with the LIST command:

```
      {  *                 }
      {  logonid           }
List  {  Like(lidmask)     }
      {  Uid(uidmask)      }
      {  If(field,...,field) }
```

*****

Tells CA ACF2 for VM to display the last referenced logonid you worked with. Your logonid is the default when you first enter the ACF command.
CA ACF2 for VM uses it until you enter another logonid.

Logonid

Tells CA ACF2 for VM to display a specific logonid record.

Like(lidmask)

Tells CA ACF2 for VM to display all of the logonid records that match the masking criteria. For example, entering LIST LIKE(SA—) displays all logonids that begin with SA.

Uid(uidmask)

Tells CA ACF2 for VM to display all of the UIDs that match the masking criteria. For example, entering UID(TLC—) displays all UIDs that begin with TLC.

If(field,...,field)

Tells CA ACF2 for VM to display logonids that contain the privileges indicated by the fields that you list in the parentheses. For example, LIST IF(SECURITY, AUTOALL) displays all users with the SECURITY and AUTOLOG privileges.

To display logonid records without a specific bit field, prefix the field with NO. For example, LIST IF(AUDIT,NOVM) displays all users with the AUDIT privilege, but without the VM privilege.

You can specify only the bit fields of the logonid record as defined by

CA ACF2 for VM or your site. If you specify multiple fields in one IF parameter, CA ACF2 for VM treats them as AND fields. All specified fields must match those in the record.

For information on CA ACF2 for VM defined fields, see the “About the Loginid Record” chapter.

You can specify two or more selection parameters, (IF, LIKE, and UID) in the same LIST subcommand. If you specify two or more selection parameters, displayed records match all of the specified criteria. You cannot specify a selection parameter with the logonid parameter.

You can use the LIST subcommand to display the fields in your logonid:

```
LIST mylogonid
```

Or you can specify LIST *. LIST * displays the last logonid referenced in your ACF session, not necessarily your own. If you had just finished listing someone else's logonid, entering LIST * redisplays that person's logonid, not yours. (To display another user's logonid record, you must have the appropriate
CA ACF2 for VM privileges.) If you did not reference any other user's logonid record before entering LIST * in your current session, CA ACF2 for VM displays your own logonid record by default.

Users with special privileges (such as AUDIT or ACCOUNT) can use the LIST subcommand to see a variety of logonid records. Enter LIST LIKE(PRG**) to display these records. This example shows how to use the LIKE parameter of the LIST subcommand with asterisks (*) as masking characters. Assuming that all employees of the programming department are assigned logonids with the leading characters PRG, this subcommand tells CA ACF2 for VM to select and display any logonid beginning with PRG, and extending up to a total of five characters (including blanks). The two asterisks account for two extra characters after PRG. For example, CA ACF2 for VM would select PRG1 and PRG10, but not PRG100 because it is six characters.

You can also use a dash (—) for masking at the end of the defined characters. It is identical to padding the rest of the positions out with asterisks. Entering LIST LIKE(PRG-) is the same as entering LIST LIKE(PRG*****). CA ACF2 for VM displays PRG100, PRG1, and PRG10. For more information about masking in ACF subcommands, see the “Masking ACF Subcommands” chapter.

Using the LIKE and UID operands with a mask lists groups of selected logonid records. The IF operand selects a logonid record only if the bit fields you selected match the values in the record. For example, entering LIST IF(AUDIT) displays only those logonid records that have the AUDIT privilege bit on. The IF operand also selects multiple combinations of privilege types. For example, LIST IF(SECURITY,ACCOUNT) displays all users who have both the SECURITY and ACCOUNT privilege, as shown in the following example of a listing.

This listing is under the VERBOSE setting:

```
acf
ACF
set verbose
ACF
list if(security,account)

TLCAMS        TAPMGRGTLCAMS   ANN SMITH    EXT.458
PRIVILEGES    ACCOUNT SECURITY
ACCESS        ACC-CNT(0) ACC-DATE(0) ACC-TIME(0)
PASSWORD      PSWD-TOD(01/04/98-12:01)
TSO           DFT-PFX(TAPMGR)
STATISTICS    UPD-TOD(01/04/98-12:01)
RESTRICTIONS  PREFIX(TLCAMS)

TLCMAS        TGPMGRGTLCMAS   MARK SCOTT   EXT.584
PRIVILEGES    ACCOUNT SECURITY
ACCESS        ACC-CNT(0) ACC-DATE(0) ACC-TIME(0)
PASSWORD      PSWD-TOD(02/23/98-12:01)
TSO           DFT-PFX(TGPMGR)
STATISTICS    UPD-TOD(02/21/98-03:12)
RESTRICTIONS  PREFIX(TLCMAS)
```

This listing is under the TERSE setting:

```
acf
ACF
set terse
ACF
list if(security,account)

TLCAMS        TAPMGRGTLCAMS   ANN SMITH    EXT.458
TLCMAS        TGPMGRGTLCMAS   MARK SCOTT   EXT.584
```

This listing is under the NOTRIVIA setting:

```
acf
ACF
set notrivia
ACF
list if(security,account)

TLCAMS        TAPMGRGTLCAMS   ANN SMITH    EXT.458
PRIVILEGES    ACCOUNT SECURITY
RESTRICTIONS  PREFIX(TLCAMS)

TLCMAS        TGPMGRGTLCMAS   MARK SCOTT   EXT.584
PRIVILEGES    ACCOUNT SECURITY
RESTRICTIONS  PREFIX(TLCMAS)
```

This listing is under the TRIVIA setting:

```
acf
ACF
set trivia
ACF
list if(security,account)

TLCAMS          TAPMGRGTLCAMS   ANN SMITH   EXT.458
PRIVILEGES      ACCOUNT SECURITY
ACCESS          ACC-CNT(0) ACC-DATE(0) ACC-TIME(0)
PASSWORD        PSWD-TOD(01/04/98-12:01)
TSO             DFT-PFX(TAPMGR)
STATISTICS      UPD-TOD(01/04/98-12:01)
RESTRICTIONS    PREFIX(TLCAMS)

TLCMAS          TGPMGRGTLCMAS   MARK SCOTT   EXT.584
PRIVILEGES      ACCOUNT SECURITY
ACCESS          ACC-CNT(0) ACC-DATE(0) ACC-TIME(0)
PASSWORD        PSWD-TOD(02/23/98-12:01)
TSO             DFT-PFX(TGPMGR)
STATISTICS      UPD-TOD(02/21/98-03:12)
RESTRICTIONS    PREFIX(TLCMAS)
```

Using ACF INSERT to Create Logonids

Use the ACF INSERT subcommand to create or modify logonid records. If you have the required privileges (specified in the SCPLIST field of your logonid record), you can create or change other logonid records. The syntax for the INSERT subcommand is:

```
Insert [Using(protolid)]{*       }  PASsword(password)
                        {logonid } {field,...,field}
```

Using(protolid)

Tells CA ACF2 for VM that you want to use an existing logonid as a model for the new logonid you are creating.

*****

Tells CA ACF2 for VM to use the last logonid you referenced. Your logonid is the default when you first enter the ACF command. CA ACF2 for VM uses it until you specify another logonid.

PASsword

Specifies the password associated with the logonid you are creating. This field is required. If you do not specify the PASSWORD field, the command fails with an error message. If you do not specify a password value, CA ACF2 for VM issues the standard password prompt and reverification request. However, if you specify the RESTRICT field when creating a logonid, you cannot specify an initial password. If you specify the STC field when you create a logonid, you do not need to specify an initial password (unless you also assigned the AUTOONLY privilege to the new logonid).

Logonid

Tells CA ACF2 for VM to use one individual logonid record.

{field,...,field}

Tells CA ACF2 for VM what logonid record fields you are adding to the logonid record. For a list of valid fields, see the“About the Loginid Record” chapter.

Using ACF CHANGE to Change User Information

You can change logonid information with the ACF CHANGE subcommand. If you have the correct privileges (specified in the SCPLIST field of your logonid record), you can change other logonid records. The syntax for the ACF CHANGE subcommand is:

```
        {*             }
        {logonid       }
CHange  {Like(lidmask) }  field,...,field
        {Uid(uidmask)  }  IF(field,...,field)
```

*****

Tells CA ACF2 for VM to use the last logonid you referenced. Your logonid is the default when you first enter the ACF command. CA ACF2 for VM uses it until you enter another logonid.

logonid

Tells CA ACF2 for VM which logonid record to change.

Like(lidmask)

Tells CA ACF2 for VM which group of logonid records to change. These logonid records match the mask you provide.

Uid(uidmask)

Tells CA ACF2 for VM to change the logonids that match the UID you specified.

field,...,field

Tells CA ACF2 for VM which fields to change, specified by their field names. For a list of valid field names, see the “About the Loginid Record” chapter.

IF(field,...,field)

Tells CA ACF2 for VM to change logonids that have the privileges that you list in parentheses. For example, CHANGE IF(SECURITY,ACCOUNT) changes all users with the SECURITY and ACCOUNT privileges. To change logonid records without a specific bit field, prefix the field name with NO.

You can specify only the bit fields of the logonid record as defined by CA ACF2 for VM or your site. For information about CA ACF2 for VM defined fields, see the “About the Loginid Record” chapter.

You can also specify the IF parameter with the LID or UID parameter.

The LIKE parameter lets you change a group of logonid records at one time if they match the specified mask. Using the UID operand and mask changes all logonid records with UIDs that match the mask. Your ability to change logonid records can be scoped.

To illustrate the CHANGE subcommand, we will first list Ann Smith's (TLCAMS) logonid record:

```
ACF

list TLCAMS

TLCAMS          AAPMGRGUSER09 ANN SMITH EXT.458
PRIVILEGES      ACCOUNT SCPLIST(ACCTMGR)
ACCESS          ACC-CNT(0) ACC-DATE(0) ACC-TIME(0)
PASSWORD        PSWD-TOD(01/04/98-12:01)
STATISTICS      UPD-TOD(01/04/98-12:01)
RESTRICTIONS    PREFIX(TLCAMS)
```

The account manager wants to change Ann's logonid record to remove the ACCOUNT privilege and add the AUDIT privilege. Ann will have the AUDIT privilege and can display the logonid records and access rules of other users.

```
ACF

change TLCAMS audit noaccount

TLCAMS          AAPMGRGUSER09 ANN SMITH EXT.458
PRIVILEGES      AUDIT SCPLIST(ACCTMGR)
ACCESS          ACC-CNT(0) ACC-DATE(0) ACC-TIME(0)
PASSWORD        PSWD-TOD(01/04/98-12:01)
STATISTICS      UPD-TOD(01/04/98-12:01)
RESTRICTIONS    PREFIX(TLCAMS)
```

The AUDIT privilege appears in the Privileges section of this logonid record and the account manager has removed the ACCOUNT attribute in the Privileges section. The AUDIT and ACCOUNT fields are bit or attribute fields of the logonid record. To add an attribute field, enter the field name in the CHANGE or INSERT subcommand (as AUDIT in the previous example). To remove an attribute field, enter the prefix NO and the field name (as in NOACCOUNT).

To add fields that have a value, enter the field name followed by the value of the field in parentheses, such as SCPLIST(ACCTMGR). To delete fields that have a value, enter the field name followed by left and right parentheses, such as SCPLIST(). The only exception to this rule is a change made to the PASSWORD field, explained in the next section.

The account manager issued this CHANGE subcommand done under the default of SET VERBOSE. If he had entered the SET TERSE subcommand before changing this logonid record, CA ACF2 for VM would have replied:

```
Logonid TLCAMS changed
```

Using ACF CHANGE to Change the PASSWORD Field

Use the ACF CHANGE subcommand to change the PASSWORD field of a logonid record. Changing the PASSWORD field in this way is useful if you forget your password. A user with the SECURITY privilege must issue the CHANGE subcommand through one of the following formats:

- To change your own password or, if you have the SECURITY privilege, change another user's password, enter:

  ```
  CHANGE logonid PASSWORD
  ```

- If you have the SECURITY privilege and need to change the password for one or more users at one time, use:

  ```
  CHANGE LIKE(lidmask) PASSWORD
  ```

  or

  ```
  CHANGE UID(uidmask) PASSWORD.
  ```

After you use one of the above methods, the following message appears asking you to enter your own password:

```
ACFpgm260R ENTER YOUR CURRENT CA-ACF2 PASSWORD
```

Then the following message appears and you must enter a new password for the logonids specified in the CHANGE subcommand:

```
ACFpgm256R ENTER NEW CA-ACF2 PASSWORD
```

CA ACF2 for VM requests that you enter the new password again for verification. You have successfully changed the passwords if the logonid listings appear on the screen after responding to the prompts. With either of these formats, you must literally enter the word **PASSWORD** in the CHANGE subcommand. The subsequent CA ACF2 for VM prompting ensures that CA ACF2 for VM does not display the new password on the terminal.

Additional Considerations

- If you remove the PROMPT=YES operand from the PASSWORD field of the @CFDE macro, changing a password does not differ from changing any other character field. For more information on the @CFDE macro of the ACFFDR, see the *Installation Guide*.
- When you log onto a group ID, you can change your own password, but not the group ID password unless CA ACF2 for VM privileges allow the group ID to change other passwords. When you log onto a group ID,
  CA ACF2 for VM treats your own logonid as the user's logonid.
- Password phrases are part of the PWPHRASE user profile record. Changes to password phrases with the ACF command are covered in the chapter "USER Profile Records" in the PWPHRASE Profile Record section.

Using ACF DELETE to Remove a Logonid

Use the ACF DELETE subcommand to delete a logonid. This subcommand deletes a specified logonid and its rule object records. The syntax for the ACF DELETE subcommand is:

```
       { *             }
       { logonid       }
DELete { Like(lidmask) }  [NORule]
       { Uid(uidmask)  }
```

*****

Tells CA ACF2 for VM to delete the last logonid you referenced. Your logonid is the default when you first enter the ACF command. CA ACF2 for VM uses it until you reference another logonid.

Logonid

Tells CA ACF2 for VM to delete this specific logonid record.

Like(lidmask)

Tells CA ACF2 for VM to delete a group of logonid records that match the mask you specified.

Uid(uidmask)

Tells CA ACF2 for VM to delete the logonid whose UID matches the mask you specified.

NORule

Tells CA ACF2 for VM not to delete the associated rule object record. You must specify this parameter if you want to delete the logonid, but retain its associated rules.

The LIKE and UID parameters let you delete groups of logonids as described in the CHANGE subcommand. The NORULE operand indicates that the access rule record of the same name (if any) is **not** deleted from the Rule database with the logonid record.

For example, to delete the TLCAMS logonid record, use the DELETE subcommand as follows:

```
delete TLCAMS

Logonid TLCAMS delete
```

Chapter 9: USER Profile Records

USER profile records are associated with a user of the system. Profile data information segments for a user include LINUX, OEVM, OMVS, PASSWORD, and PWPHRASE. The key of a USER profile record is a userid. For the currently supported USER profile records, this field is not maskable.

For all user profile records, you must use the SET command to enter:

```
SET PROFILE(USER) DIVISION(profile_data_name)
```

The LIST command will only display the currently active set of profile records.

This section contains the following topics:

Using the ACF Command

Once within the ACF command, use the SET command to enter USER profile administration. Enter:

```
SET PROFILE(USER) DIVISION(profile-data-name)
```

DIVISION is the profile data name of the data record.

The following commands can be used with profile records:

```
INSERT *|recid|USING(urecid) recid

LIST *|recid|LIKE(recidmask)

DELETE *|recid|LIKE(recidmask)

CHANGE recid|LIKE(urecid)
```

LINUX Profile Data Records

The LINUX profile data record contains information needed by the LINUX PAM Server to verify user access. It maps the Linux application user identity to an CA ACF2 for VM logonid.

Record ID	Fields
recid	LINUXGRP(*group-name*) LINUXHOM(*home-directory*) LINUXNAM(*application-user-id*) LINUXPGM(*program*) LINUXUID(uid#)

Field Descriptions

recid

Specify a one-to-eight character CA ACF2 for VM logonid. This value cannot be masked.

LINUXGRP(*group-name*)

Specifies the name of the LINUX group profile record. A one to eight character field. This field is optional.

LINUXHOM(*home_directory*)

Specifies the Pathname of the Initial Directory when a user enters a Linux command or the ISPF shell. A 1 to 1024 upper or lower case alphanumeric character field. This is a required field. If left undefined on an INSERT command, it defaults to '/home/%L'.

The following strings substitutions are allowed:

Symbolic String Substitution

- %L Substitutes an CA ACF2 for VM logonid (8 bytes).
- %N Substitutes to user name defined in CA ACF2 for VM logonid record.
- %X Substitutes to LINUX/390 application user id defined in LINUXNAM field.

Note: If LINUXNAM field is not defined, it is substituted to CA ACF2 for VM logonid.

LINUXNAM(*application-user-id*)

Specifies the LINUX Application User Identity. A 1 to 1024 upper or lower cased alphanumeric character field. This field is optional.

LINUXPGM(*program*)

Specifies LINUX/390 Service Shell Program when Linux command is first entered. A 1 to 1024 upper and lower cased alphanumeric character field. This field is required. If this field is left undefined on an INSERT command, it defaults to '/bin/bash'.

LINUXUID(*uid#*)

Specifies the LINUX uid value. A numeric field that accepts values from 0 to 2,147,483,647. This field is required.

Commands

To build a cross-reference table to map a LINUX/390 application user id to an CA ACF2 for VM logon id:

```
ACFSERVE RELOAD PROFILE LINUX USER
```

Example

The following example inserts LINUX user profile record, LNXUSER9 with Linux application user id value of Linuxuserid9.

```
INSERT LNXUSER9 linuxnam(linuxuserid9) linuxgrp(linuxgrp) linuxuid(1234)

  LINUX / LNXUSER9 LAST CHANGED BY USER01 ON 04/11/05-15:55

        LINUXGRP(LINUXGRP) LINUXHOM(/home/%L) LINUXNAM(linuxuserid9)

        LINUXPGM(/bin/bash) LINUXUID(1,234)
```

OEVM Profile Data Records

The OEVM profile data records contain information needed by OpenExtensions to verify user access.

Record ID	Fields
recid	UID(*uid*) HOME(*home-directory*) OEVMPGM(*program*) FILESYS(*bfsname*)

Field Descriptions

recid

The CA ACF2 for VM logonid. This value cannot be masked.

UID(*uid*)

A required numeric field that accepts values from zero to 2,147,483,647. A value of zero indicates that this user is a superuser.

HOME(home-directory)

A field that defines the pathname of the initial directory used when a user enters the OPENVM SHELL command. Specify from one to 1023 upper-case or lower-case characters. If HOME is not defined, OpenExtension sets root as the initial directory.

OEVMPGM(*program*)

An optional field that defines the user's OpenExtensions shell program started when the OPENVM SHELL command is entered. Specify from one to 1023 upper-case or lower-case characters. If OEVMPGM is not defined, OpenExtensions gives control to the default shell program.

FILESYS(*bfsname*)

A field that defines the name of the OpenExtensions Byte File System (BFS) for the user. The value for *bfsname* is specified in the format:

/../VMBFS:filepool:filespaceid/

where filepool is the SFS filepool the BFS resides in, and filespaceid is usually ROOT. Specify from one to 1023 upper-case or lower-case characters. If you do not define FILESYS, OpenExtensions VM requires the user to issue the OPENVM MOUNT command to mount a Byte File System.

Commands

If you insert or change a user profile record, then you must issue a REBUILD command to activate the changes.

```
ACFSERVE RELOAD PROFILE OEVM USER
```

Examples

The following example shows how to define an OEVM user:

```
SET PROFILE(USER) DIV(OEVM)

INSERT OEVMUSR UID(199) HOME(/u/oevmusr) OEVMPGM(/bin/sh)
FILESYS(/../VMBFS:VMSYS:ROOT/)
```

OMVS Profile Data Records

OMVS records are normally used only on CA ACF2 for VM for z/OS systems. For OpenExtensions on VM, see the OEVM record. For CA PAM for Linux for System z support see the LINUX record.

Previously, the OMVS profile data record was only used in VM for CA PAM for Linux for System z support. CA PAM for Linux for System z r8.0 and above is designed to use the information in the new fields of the Linux records instead of using the OMVS record. For compatibility, CA PAM for Linux for System z r8 still allows the use of OMVS records if **linuxdata OMVS** is included in the CA PAM for Linux for System z configuration file.

Record ID	Fields
recid	UID(*uid*) HOME(*home-directory*) OMVSPGM(*program*) CPUTIME(*max cputime for a dubbed process*) ASSIZE(*max address space size*) FILEPROC(*max files per process*) PROCUSER(*max number of processes*) THREADS(*max number of pthread_created threads*) MMAPAREA(*max data space pages for HFS mappings*)

Note: The CPUTIME, ASSIZE, FILEPROC, PROCUSER, THREADS, and MMAPAREA values are only valid at z/OS and OS/390 2.8 and above. You might set them at any level, but they will not be recognized until that level. These fields are not used in VM, but are supported by the ACF command for compatibility for users using the Database Synchronization Component.

Field Descriptions

recid

The CA ACF2 for VM logonid. This value cannot be masked.

UID(*uid*)

A required numeric field that accepts values from zero to 2,147,483,647. A value of zero indicates that this user is a superuser. The NO-OMVS attribute on the logonid nullifies the user profile record and BPX.DEFAULT.USER for this user. NO-OMVS is equivalent to NOUID in RACF.

HOME(home-directory)

A field that defines the pathname of the initial directory used when a user enters the OMVS command or enters the ISPF shell. Specify from one to 1023 upper-case or lower-case characters. If HOME is not defined, UNIX System Services (USS) sets root as the initial directory.

OMVSPGM(*program*)

An optional field that defines the user's UNIX System Services shell program started when the OMVS command is entered or when a UNIX System Services batch job is started using the BPXBATCH program. Specify from one to 1023 upper-case or lower-case characters. If OMVSPGM is not defined, UNIX System Services gives control to the default shell program.

CPUTIME(max-cputime-for-a-dubbed-process)

This field overrides the MAXCPUTIME parameter in the BPXPRMxx member of PARMLIB for this user. The value can be from 7 to 2,147,483,647. This field is not set when the record is inserted. If the field is not set, a RACROUTE EXTRACT will return a length of 4 and a value of X'FFFFFFFF'. Unix System Services takes the system defaults set in BPXPRMxx when this field is not set. You can remove this field from the record by changing it to a null value (change user CPUTIME()). CPUTIME is only valid for z/OS and OS/390 2.8 and above.

ASSIZE(max-address-space-size)

This field overrides the MAXASSIZE parameter in the BPXPRMxx member of PARMLIB for this user. The value can be from 10,485,760 to 2,147,483,647. This field is not set when the record is inserted. If the field is not set, a RACROUTE EXTRACT will return a length of 4 and a value of X'FFFFFFFF'. Unix System Services takes the system defaults set in BPXPRMxx when this field is not set. You can remove this field from the record by changing it to a null value (change user ASSIZE()). ASSIZE is only valid for z/OS and OS/390 2.8 and above.

FILEPROC(max-files-per-process)

This field overrides the MAXFILEPROC parameter in the BPXPRMxx member of PARMLIB for this user. The value can be from 3 to 65,535. This field is not set when the record is inserted. If the field is not set, a RACROUTE EXTRACT will return a length of 4 and a value of X'FFFFFFFF'. Unix System Services takes the system defaults set in BPXPRMxx when this field is not set. You can remove this field from the record by changing it to a null value (change user FILEPROC()). FILEPROC is only valid for z/OS and OS/390 2.8 and above.

PROCUSER(max-number-of-processes)

This field overrides the MAXPROCUSER parameter in the BPXPRMxx member of PARMLIB for this user. The value can be from 3 to 32,767. This field is not set when the record is inserted. If the field is not set, a RACROUTE EXTRACT will return a length of 4 and a value of X'FFFFFFFF'. Unix System Services takes the system defaults set in BPXPRMxx when this field is not set. You can remove this field from the record by changing it to a null value (change user PROCUSER()). PROCUSER is only valid for z/OS and OS/390 2.8 and above.

THREADS(max-number-of-pthread_created-threads)

This field overrides the MAXTHREADS parameter in the BPXPRMxx member of PARMLIB for this user. The value can be from 0 to 100,000. This field is not set when the record is inserted. If the field is not set, a RACROUTE EXTRACT will return a length of 4 and a value of X'FFFFFFFF'. Unix System Services takes the system defaults set in BPXPRMxx when this field is not set. You can remove this field from the record by changing it to a null value (change user THREADS()). THREADS is only valid for z/OS and OS/390 2.8 and above.

MMAPAREA(max-data-space pages-for-HFS-mappings)

This field overrides the MAXMMAPAREA parameter in the BPXPRMxx member of PARMLIB for this user. The value can be from 1 to 16,777,216. This field is not set when the record is inserted. If the field is not set, a RACROUTE EXTRACT will return a length of 4 and a value of X'FFFFFFFF'. Unix System Services takes the system defaults set in BPXPRMxx when this field is not set. You can remove this field from the record by changing it to a null value (change user MMAPAREA()). MMAPAREA is only valid for z/OS and OS/390 2.8 and above.

Commands

If you insert or change a user profile record, then you must issue a REBUILD command to activate the changes.

```
ACFSERVE RELOAD PROFILE OEVM USER
```

Examples

The following example shows how to define an OMVS user:

```
SET PROFILE(USER) DIV(OMVS)

INSERT OMVSUSR UID(199) HOME(/u/omvsusr) OMVSPGM(/bin/sh)
```

PASSWORD Profile Data Record

The PASSWORD segment of the USER profile is used to retain previous passwords. Previous passwords are stored in this record when the VMO PSWD record specifies PSWXHIST and PSWXHST(nn) = 5 or more. Most fields in this record are not displayed. Users can list and delete, but cannot insert or change this record. It is maintained internally by CA ACF2 for VM.

Record ID	logonid
logonid	#PSWDCNT #PWD-TOD

Field Descriptions

logonid

The CA ACF2 for VM logonid

#PSWDCNT

Specifies the number of previous passwords stored in the record for extended password history. Users cannot change this field. The #PSWDCNT field is always one count less than the maximum number named in the PSWXHST(*nn*) field of the VMO PSWD record. This is due to the fact that the current password is included in the count for PSWXHST(*nn*) and the User Profile password record keeps all but the current password.

#PWD-TOD

Specifies the date and time when the user's most recent old password was saved. CA ACF2 for VM lists the date in the format mm/dd/yy, dd/mm/yy, or yy/mm/dd, depending on the DATE field of the VMO OPTS record. You cannot set this field. CA ACF2 for VM maintains and displays it. This field should match the PSWD-TOD in the logonid record. Users cannot change this field.

PWPHRASE Profile Record

The PWPHRASE segment of the USER profile is used to retain user password phrase control information and history. A maximum of 32 password phrases can be stored in this record. By default, CA ACF2 for VM saves the current and three previous password phrases. Most fields in this record are not displayed and are maintained internally by CA ACF2 for VM.

Record ID	Fields
logonid	PWP-EXP|NOPWP-EXP PWP-HST(0|*nn*) PWP-MAXD(0|*nnn*) PWP-TOD(*date*) PWPHRASE(*password phrase*)

Using ACF CHANGE to Change the PWPHRASE Field

Changes can be applied to a password phrase by the individual user or by a user with security privileges. When changing another user's password phrase you must have security privileges.

To modify your password phrase

1. To enter USER profile administration issue the following from the ACF command:

   ```
   SET PROFILE(USER) DIVISION(PWPHRASE)
   ```

2. Change your password phrase by issuing the following:

   ```
   CHANGE logonid PWPHRASE
   ```

   The following message displays asking you to enter your current password. Your password phrase can be used also.

   ```
   ACFpgm260R ENTER YOUR CURRENT CA ACF2 PASSWORD
   ```

3. Enter your current password.

   The following message displays asking you to enter a new password phrase:

   ```
   ACFpgmB15R Enter new CA-ACF2 password phrase
   ```

4. Enter your new password phrase.

   The following message displays requesting you to verify the new password phrase:

   ```
   ACFpgmB14R Enter new CA-ACF2 password phrase again to verify
   ```

   The PHWPHRASE record displays verifying that the changes were made.

To modify another user's password phrase

1. Change another user's password phrase by issuing the following:

   ```
   CHANGE logonid PWPHRASE
   ```

   The following message displays asking you to enter your current password. Your password phrase can be used also.

   ```
   ACFpgm260R ENTER YOUR CURRENT CA ACF2 PASSWORD
   ```

   Note: You must have security privileges to change another user's password phrase.

2. Enter your current password.

   The following message displays asking you to enter a new password phrase:

   ```
   ACFpgmB15R Enter new CA-ACF2 password phrase
   ```

3. Enter your new password phrase.

   The following message displays requesting you to verify the new password phrase:

   ```
   ACFpgmB14R Enter new CA-ACF2 password phrase again to verify
   ```

   The PHWPHRASE record displays verifying that the changes were made.

4. Change one or more users' password phrase at a time by issuing the following:

   ```
   CHANGE LIKE(lidmask) PWPHRASE
   ```

You have successfully changed the password phrases if the PWPHRASE record listing appears on the screen after responding to the prompts, or the number or records changed is displayed if the LIKE format is used. With either of these formats, you must literally enter the word PWPHRASE in the CHANGE subcommand. The subsequent CA ACF2 for VM prompting ensures that CA ACF2 for VM does not display the new password phrase on the terminal.

Chapter 10: About Access Rules

CA ACF2 for VM protects all data by default. This means that no one can access (read, write, or execute) a file other than the **owner** of the file or a security administrator unless an access rule exists that specifically allows the access. CA ACF2 for VM automatically protects all data. CA ACF2 for VM provides for controlled access to data through access rules.

An access rule entry is a statement that defines the data you want to protect, who can access that data, and what type of access they can have (can they just look at the data, or can they change it). As you become more familiar with
CA ACF2 for VM, you can limit when users can access the data, and where they can access the data from. There are many other limitations you can put on users, and you will learn about those later in this guide.

An access rule set is a collection of rule entries that apply to one element. An element can be all of the minidisks a user ID owns, or the high level index of VSE or OS/390 data sets. The rule set name, or $KEY value, is a user ID or the high level index of a data set name. The easiest way to think of a rule set is that it is a file that contains rules that allow or prevent others from reading or changing your data.

Writing access rules for all data on CA ACF2 for VM may seem like a complicated task at first, but keep in mind these three important features:

- CA ACF2 for VM is integrated into your computer operating system in modes. These modes are designed to provide time for you to develop and test rules and local CA ACF2 for VM features. The five modes of
  CA ACF2 for VM are QUIET, LOG, WARN, RULE, and ABORT. For more information on these modes, see the *Implementation Planning Guid*e.
- You can centralize or decentralize the different aspects of rule administration. In a centralized environment, security administrators have the authority to maintain access rules. If you specify NOCENTRAL in the RULEOPTS VMO record, rule writing is decentralized. This lets each user change the rule set that matches his PREFIX (often the same as the user's logonid).
- You can also delegate rule administration to different individuals and control what rules they can administer through the SCPLIST facility or through rule sets that specify who can change various aspects of the rule. See the “About the Loginid Record” chapter for more information about SCPLIST.

  Masking lets you write rule entries that apply to groups of users and groups of data. Masking greatly decreases the number of necessary rules.

When you finish this chapter, you will know

- What access rules are and why you need them to protect your data
- What masking is and how to use it

- How to change the full-screen feature options and its PF keys
- How to set the correct mode for rule processing
- What ACF subcommands are valid for processing rules.

This section contains the following topics:

Why You Need Access Rules

CA ACF2 for VM employs a two-level access protection scheme for data accesses. One level is for minidisks and DASD volumes when you issue a CP LINK or ATTACH command, or when CA ACF2 for VM processes a VM directory LINK or DEDICATE statement. Minidisk and DASD volume protection is always active. The second level of protection applies to file accesses. This is called CMS file-level protection.

CMS file-level protection is optional and applies to file accesses using VM diagnose I/O or accesses using OS/390, VSE, or VSAM services of CMS. Issue the ACF SHOW SYSTEMS subcommand to determine if CMS file-level security is active and examine the CMS SECURITY field.

An access rule applies to different entities. For VM data, an access rule applies to minidisks and CMS data a VM user ID owns. The key to a VM rule is the user ID. For attachable DASD devices, a special rule called a SYSTEM rule validates accesses. The key to the SYSTEM rule is defined in the OPTS VMO record. Data set rules govern access to OS/390 and VSE data sets. The key to an OS/390 or VSE rule is the high-level index of the data set name.

Access Rule Entries

An access rule entry consists of many parts. The main parts of an access rule entry describe the data, who the rule entry applies to, and the access permissions. The data set name describes the data. For VM data, CA ACF2 for VM constructs a pseudo-DSN when a user accesses the data. The UID defines who the rule entry applies to. The access permissions instruct CA ACF2 for VM whether to allow the access, allow the access but also log it, or prevent and log the access.

VM Pseudo-DSNs

When a virtual machine tries to access data, CA ACF2 for VM constructs a pseudo-DSN as input into access validation. The pseudo-DSN for minidisk and DASD volume accesses is described below.

When CA ACF2 for VM processes a LINK, ATTACH or DEDICATE it constructs a VM volume pseudo-DSN.

```
owner.tccuu.VOLUME
```

owner

Specifies the data owner. For minidisk LINKs, the owner is the user ID that has the MDISK statement for the minidisk. For volume ATTACHES and DEDICATES, the owner is the value defined in the OPTS VMO record. SYSTEM is the default. The owner field is the key to the rule.

Specifies the volume type. V is for virtual volumes or minidisks. R is for real volumes.

ccuu

Specifies the VM directory address of the volume, for example, 0191.

VOLUME

Specifies the volume rule entry. The keyword volume indicates that the rule entry applies to a volume access through the LINK or ATTACH commands, or a VM directory LINK or DEDICATE access. For example, CA ACF2 for VM constructs pseudo-DSNs as follows:

```
LINK MAINT 0191 AS 1AF1 RR

Rule key MAINT, pseudo DSN V0191.VOLUME,
access read

LINK ACF2VM 201 AS 201 RR

Rule key ACF2VM, pseudo DSN V0201.VOLUME,
access read

ATTACH 123 TO VSEIPO

Rule key SYSTEM (or other value if overridden), pseudo DSN R0123.VOLUME,
access write.

ATTACH 201 TO MVSTEST AS 3F21 R/O

Rule key SYSTEM (or other value if overridden), pseudo DSN R0201.VOLUME,
access read.

DEDICATE 401

Rule key SYSTEM (or other value if overridden), pseudo DSN R0401.VOLUME, access
write.
```

CMS File Rules

If the optional CMS file-level protection is active, CA ACF2 for VM constructs a pseudo-DSN describing the file being accessed. When you open a file, or the first I/O occurs, CA ACF2 for VM validates the DSN. For CMS file accesses, the pseudo DSN is:

```
owner.Vccuu.filename.filetype
```

owner

Specifies the data owner. The data owner is the VM user ID that has the MDISK statement for the minidisk. The owner field is the key to the rule.

V

Specifies the volume type. Minidisks are virtual volumes.

ccuu

Specifies the VM directory address of the volume, for example, V0191.

filename

Specifies the CMS filename.

filetype

Specifies the CMS filetype.

If TDISKVLD is in effect, the access rules for CMS files on T-disks have a unique pseudo-dsn syntax:

owner.VMTDISK.filename.filetype

owner

Specifies the data owner. This is the user ID of the owner of the T-disk.

VMTDISK

Specifies a literal identifying this rule that applies to **all** T-disks, regardless of virtual address.

filename

Specifies the CMS filename.

filetype

Specifies the CMS filetype.

OS/390 and VSE File Rules

For OS/390 and VSE file accesses using the CMS OS/390, VSE, or VSAM file services, the complete data set name passes to CA ACF2 for VM for validation. The DSN is described below.

```
dsnqual1.dsnqual2.dsnqual3...
```

dsnqual1

Specifies the first segment of a DSN. This is also called the high-level index of the DSN. The high-level index is the key to the rule.

dsanqualn

Specifies a segment of a DSN.

CA ACF2 for VM constructs DSNs as follows. The access level is dependent on how the file was opened or the type of I/O read or write occurring:

```
file PAYROLL.MASTER.DATA

Rule key PAYROLL rule entry DSN is MASTER.DATA

CMS file PROFILE EXEC on MAINT 0191

Rule key MAINT, pseudo DSN V0191.PROFILE.EXEC
```

CMS DSN Considerations

If you are using CMS file-level protection, CA ACF2 for VM performs at least two validations. One when a user LINKs a minidisk against a VM minidisk rule entry, and the other when the user opens a file or performs the first I/O operation against a file.

The net result is that you have combined rules and separate rules depending on the type of files being accessed. For example, consider a user requesting access to the following data:

```
VSAM Files: TAX.MASTER.INDEX on PAYROLL 0201 minidisk
      TAX.MASTER.DATA on PAYROLL 0202 minidisk

CMS Files: CALCTAX MODULE on PAYROLL 0192 minidisk
      TAX   TABLE on PAYROLL 0193 minidisk
```

First, LINK to the minidisks. The LINK commands then generate the following DSNs:

```
PAYROLL.V0201.VOLUME
PAYROLL.V0202.VOLUME
PAYROLL.V0192.VOLUME
PAYROLL.V0193.VOLUME
```

If the files were opened or an I/O operation occurred against them, CA ACF2 for VM generates the following DSNs:

```
TAX.MASTER.INDEX
TAX.MASTER.DATA
PAYROLL.V0192.CALCTAX.MODULE
PAYROLL.V0193.TAX.TABLE
```

Taking the above DSNs and assuming that you did not use masking, your rules need to have the following DSNs:

```
$KEY(PAYROLL)
 V0192.CALCTAX.MODULE
 V0193.TAX.TABLE
 V0201.VOLUME
 V0202.VOLUME
 V0192.VOLUME
 V0193.VOLUME

$KEY(TAX)
 MASTER.INDEX
 MASTER.DATA
```

The VM minidisk access and CMS file rules are covered in one rule set based on who owns the minidisk. For the VSAM files, there are two rules involved. The PAYROLL rule covers the minidisk access, and the TAX rule covers the VSAM file access. This is because of the need to be compatible with CA ACF2 for VM for OS/390 where typically the high-level index determines file ownership. A good example is how CA ACF2 for VM for OS/390 generates TSO DSNs based on the logonid as the high-level index. However, by carefully choosing filenames, you can cover all of the accesses with one rule, for example, if you renamed the TAX.* files to PAYROLL.*.

VM LINK Modes

To ease rule writing, CA ACF2 for VM normalizes LINK modes to READ or WRITE access. For example, link modes R and RR are READ accesses. W, WR, M, MR, MW are WRITE accesses. CA ACF2 for VM does not interfere with the VM LINK logic for link modes. If a user requests an R link and someone already has a link to the disk, VM denies the link. If you need to protect the individual link modes, we provide this protection through command limiting. For example, you could prevent all MW links with the following rule:

```
$KEY(LINK)
 - MW PREVENT
```

CA ACF2 for VM allows for separate READ and WRITE validations. WRITE does not imply READ. On the minidisk level, validations for read links (such as R, RR, and so on) are validated as READ, and write links (such as W, WR, MR, MW, and so on) are validated as WRITE. However, some write links can end up as R/O links (such as if another user has the disk linked R/W). Also, once a write link is allowed, CP establishes a R/W link. This means that CP (and CA ACF2 for VM) allow data to be both read and written at the minidisk level. File level security does allow for WRITE-ONLY files in the constraints of the CMS operating system.

See the *Command and Diagnose Limiting Guide* for more information.

Using Masking in Access Rules

You can use masking when you write access rules with the full-screen feature or the ACF command. The UID parameter, for example, can contain masks that let many users access specific data. This section contains information about masking that you should read to make rule writing easier.

Masking the User Identification String

In access rule entries, the User Identification string (UID) describes users. Contrary to popular use, the UID is not hierarchical in nature. However, many users use it this way, which can lead to masking difficulties because of the way CA ACF2 for VM sorts rule sets. You can use the UID in a hierarchical manner if you use caution when writing access rules. Construct the UID with the
most-used field first, the second-used next, and so on. Below is an example of a UID constructed hierarchically with UIDs mixed in a rule set:

```
UID(C) READ(A) EXEC(A)
UID(*****TLCAMS) READ(A) EXEC(A) WRITE(A)
```

The sorting sequence is 1,2. With a mixed rule like this, if TLCAMS was a member of “C”, TLCAMS could not write to any of her files because CA ACF2 for VM would look at the rule number one first. You could change rule number two to UID(C****TLCAMS) to solve this. Then CA ACF2 for VM sorts it **before** the old rule number one.

To make rule writing easier, you can mask the UID. Standard CA ACF2 for VM masking characters are asterisks, imbedded blanks, and dashes.

Asterisks

When placed in the UID field in an access rule, asterisks indicate a one-to-one substitution of any character, including a blank. CA ACF2 for VM automatically pads out the UID to its maximum length (24 characters) for access rules. (The exception to this is the UID parameter in the SCPLIST field. See the “About the Loginid Record” chapter for more details.) For example, you can define your UID as ten positions. The following UID mask matches these UIDs:

Mask	Matches
UID(ABC)	ABC ABC ABCDEF ABCDEFG234

You can insert one or more asterisks after ABC and the same translation occurs because CA ACF2 for VM pads out the UID to its maximum length.

Mask	Matches
UID(ABC*)	ABC ABCDEF ABCDEFG234

Asterisks in or at the beginning of the UID are substituted on a one-to-one basis for any character, including a blank. Only ending asterisks match a null. Imbedded or leading asterisks are positional and never match a null.

Mask	Matches	Does Not Match
UID(ABC*23)	ABC123 ABCD23 ABC 23	ABC23
UID(*ABC)	1ABC XABC _ABC	ABC

Imbedded blanks

Blanks are treated as any other valid character and can appear in a UID. They can match only blanks and are not treated as masks.

Mask	Matches	Does Not Match
UID(ABC 234)	ABC 234	ABC1234
UID(ABC)	ABC 2 ABC 234	ABC1

To match, a blank must exist as the fourth character in these examples.

Dashes

You can use a dash as a masking character if it is in the last position of the mask and is translated the same way as an asterisk.

Consider these three UID masks:

```
UID(ABC-) UID(ABC*) UID(ABC)
```

All three are translated to the same name. Again, CA ACF2 for VM pads out the UID to its maximum length.

CA ACF2 for VM always treats an imbedded dash or one that is a leading character in a mask as a literal dash.

Mask	Matches	Does Not Match
UID(AB-CD)	AB-CD AB-CDE AB-CDEF	ABCD ABXCD
UID(-ABC)	-ABC	1ABC 12ABC

In these cases, CA ACF2 for VM treats the dash as a character, not as a masking element.

Access Rule Masking

You can mask access rule data set names, filenames, and UIDs in access rule entries. You cannot mask the $KEY control statement for access rules.

Masking the DSN Value

In access rules, the data set name (dsn) operand describes the data that the rule applies to. For example, the dsn can describe a minidisk, a CMS file, an OS/390 or VSE data set, an attachable DASD device, or a tape volume volser.

To make it easier to write access rules, CA ACF2 for VM lets you specify dsn values as masks. This is especially helpful when you need to write a large number of rules for similar minidisks and CMS files. Data set name masking characters include asterisks and dashes that you can combine in a single mask.

Asterisks

Asterisks are positional masking characters. You can substitute any character, including a blank, on a one-to-one basis for each asterisk in a mask.

Mask	Matches
V0191.WORK**.DATA	V0191.WORK11.DATA V0191.WORK22.DATA V0191.WORK33.DATA

If one or more asterisks are the last characters in a level, they also can match a null. This is the only case where asterisks match a null.

Mask	Matches	Does Not Match
V0191.WORK**	V0191.WORK11 V0191.WORK1 V0191.WORK	V0191.WORK111

Here, the filename mask WORK** must include the characters WORK and, optionally, any other two characters. The filename cannot be more than eight characters.

If one or more asterisks are the first characters in a level or are imbedded between characters, they cannot match a null, as shown below:

Mask	Matches	Does Not Match
*FILE.DATA	TFILE.DATA XFILE.DATA	FILE.DATA OLDFILE.DATA
FILE.DA*A	FILE.DATA	FILE.DAA

The mask *FILE.DATA indicates that immediately before the “F” in a filename beginning with FILE, there must be only one other character. The mask FILE.DA*A indicates that between the two “A”s in DA*A, there must be only one other character. In these cases, an asterisk does not match a null.

Dashes

Use a dash after one or more characters in a device address, filename, or filetype to indicate that the remainder of the level is treated as if it were padded with asterisks for its full length (a maximum of eight characters).

Mask	Matches
V091.WORK-.DATA	V0191.WORK.DATA V0191.WORK1.DATA V0191.WORK11.DATA V0191.WORK111.DATA V0191.WORK1111.DATA

The filename must begin with WORK and can continue with any other characters, including blanks, up to a total of eight positions.

Besides padding out one or more characters at the end of a level, a dash also pads out an entire level in either direction. If the dash is not between two periods (.), it can pad out more than one level.

Mask	Matches
V0191.- V0191.FILE1.- V0191.FILE1.L- V0191.-.LIST -.V0191.- -.FILE1.- -.LIST	V0191.FILE1.LIST

Using a dash is even more flexible in representing OS/390 data set names that can contain many levels:

Mask	Matches
WORK.DATA.-	WORK.DATA WORK.DATA.TEST WORK.DATA.TEST.LISTING

Mask	Matches
WORK-.LISTING	WORK.LISTING WORK.DATA.LISTING WORK.TEST.LISTING WORK.DATA.TEST.LISTING
-.LISTING	LISTING TEST.LISTING DATA.TEST.LISTING WORK.DATA.TEST.LISTING

One point to remember about a dash is that it is only treated as a masking character when it is at the end of a dsn level (V019-) or when it represents one or more entire levels (V0191.). If a dash is imbedded between characters in a level (FIL-1) or if it appears at the beginning of characters in a level (-FILE1), it is considered to be a literal dash and not a masking character.

Combination Asterisk and Dash

You can combine asterisks and dashes in a file mask.

Mask	Matches	Does Not Match
V0191.*-.****	V0191.OLD.DATA V0191.WORK.TEXT V0191.SHOW.EXEC	V0191 V0191.MAP V0191.WORK.ASM

This mask requires a device address of V0191, a filename of at least one character, and a filetype of any four characters (with blanks or trailing nulls valid).

Resetting the Character Delete Definition

Some CA ACF2 for VM rules contain the @ sign that VM interprets as a delete character. We recommend that you substitute another character for @. Enter the following command to substitute another character for @ where char is the character you want to use as the delete character:

```
TERMINAL CHARDEL char
```

Using NEXTKEY

The NEXTKEY operand splits a very large rule set into several sets or merges several rule sets together.

Specify the rule ID of an alternate rule set in the NEXTKEY operand. If access to a file based on the environment and permissions specified in the current rule set is prevented, CA ACF2 for VM searches the alternate rule set specified through NEXTKEY.

When NEXTKEY specifies an alternate rule set, a security administrator needs to grant authority to who is responsible for writing and maintaining that rule set through the %CHANGE and %RCHANGE control statements. These control statements are placed in the alternate rule set that must initially be established by the security administrator.

For %CHANGE and %RCHANGE authorizations to be active, you must specify the CHANGE operand of the RULEOPTS VMO record. If you specify NOCHANGE, all %CHANGE and %RCHANGE authorizations are inactive. The default is CHANGE. This default value is required to use %CHANGE or %RCHANGE.

Chapter 11: Maintaining Access Rules with the Full-Screen Feature

This chapter provides information for maintaining access rules through the full-screen feature.

When you complete this chapter, you will know how to use the full-screen feature to:

- Change your PF keys for access rules screens
- Write access rules (including rules for Shared File System (SFS) files and directories, minidisk access, OS/390 and VSE data sets)
- Change an existing access rule
- Display an access rule

- Delete an access rule (and verify the deletion)
- Test an access rule before it is stored on the Rule database.

This section contains the following topics:

Writing Access Rules

To begin writing access rules, you must access the Data Access Control (2) screen. To get to this screen, enter the full-screen feature described in the “Using the CA ACF2 for VM Full-Screen Feature” chapter and select option 2 from the Primary Option Menu.

Use this screen to select the type of data access maintenance you want to do. Enter your selection on the option line.

```
M9PA-2000      Data Access Control (2)                CA ACF2 for VM
OPTION ===> ____________________________________________________________
                                                            TIME 17:12
      0 Change options or PFkeys

      1 Add Access Rule Set

      2 Display Access Rule Set(s)

      3 Change Access Rule Set(s)

      4 Delete Access Rule Set(s)

      5 Test an Access Rule Set

      Option may be followed by “.rulekey” to select
      the rule set to which the option is to apply.




PF1=Help       2=Print       3=Quit       4=Return      5=        6=
PF7=           8=            9=Director   10=           11=       12=Retrieve
```

You can enter a number followed by a rulekey to indicate the rule set to be affected. For example, the following displays the rule set with the $KEY of TLCAMS:

```
OPTION ===> 2.TLCAMS
```

Valid options are:

0 Change Options or Pfkeys

Displays the current values of the PF key settings and lets you change them.

1 Add Access Rule Set(s)

Lets you add new access rule sets.

2 Display access rule set(s)

Displays current access rule sets. You cannot update this field.

3 Change Access Rule Set(s)

Lets you change currently existing access rule sets.

4 Delete Access Rule Sets

Lets you delete access rule sets. There is no recovery from this action, so use caution. If you delete a rule set in error, you will have to recreate it.

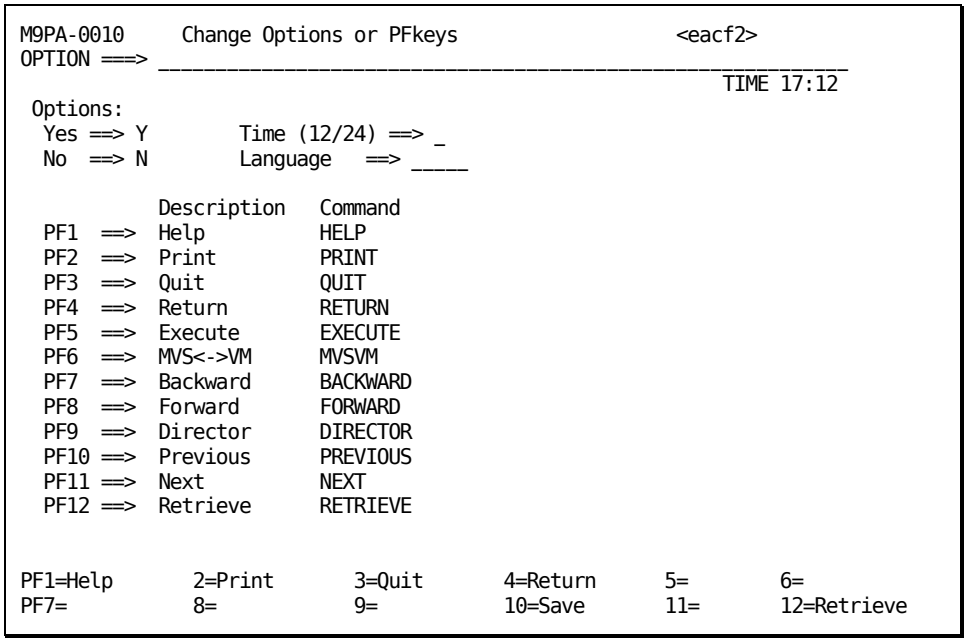
5 Test an Access Rule Set

Lets you test an access rule set before it is compiled and stored on the Rule database.

Changing Options or PF Keys

You see this screen when you select option 0 from the previous screen, Data Access Control. You can also select 2.0 from the Primary Option menu and go directly to this screen.

Use this screen to change the options or PF keys for the **access rules** screens only.



Following is a brief description of the fields on this screen.

Yes

Indicates how you want yes values displayed. Enter 1 (to indicate binary) or Y (to indicate alpha) in this field.

No

Indicates how you want no values displayed. Enter 0 (to indicate binary) or N (to indicate alpha) in this field.

Time(12/24)

Enter whether you want the time reported in a 12-hour (enter 12) or 24-hour (enter 24) clock.

Language

Enter the five-character code for the language you want to use for screens. CA supplies the following valid options:

AMENG

American English

UCENG

Upper Case English.

Your site may have added additional language support. Contact your security administrator or systems programmer for this information.

To change the values for PF keys, enter the value to be displayed on the screen for the PF key in the Description column. Enter the actual command names (in caps) in the Command column.

Creating an Access Rule Set

To see this screen, select option 1, Add Access Rule Set, from the Data Access Control screen. You can also select 2.1 from the Primary Option Menu and go directly to this screen.

```
M9PA-2110 Add Access Ruleset Control Information (2.1.1) CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 12:43

Rulekey     ==> ________
Prefix      ==> ________________________
Owner       ==> ________________________
Resowner    ==> ________
Mode        ==> _____
Userdata    ==> ____________________________________________________
Sort Option ==> _









Total Rule Entries : ___

PF1=Help       2=Print       3=Quit       4=Return      5=Execute   6=
PF7=           8=            9=           10=Previous   11=Next     12=Ret
```

Following is a brief description of the fields on this screen.

Rulekey

Specifies the key value of the rule set. Normally, the rulekey is the ID of a VM user who has one or more MDISKs specified in the VM directory under his user entry. In most cases, this is also the user's CA ACF2 for VM logonid. The rulekey can also be the high-level qualifier of an OS/390 or VSE data set. The rule ID specified can be up to eight characters long for access rules. You cannot mask this field.

Prefix

Tells CA ACF2 for VM logonid the value to use as the high-level qualifier for validations instead of the rulekey. You can enter up to 24 characters and specify multiple levels in this control statement (TLCAMS.V0191). You do not need to specify this field.

Owner

Tells CA ACF2 for VM logonid the ID of the owner of this rule set. This field is for information only. CA ACF2 for VM logonid does not process rules based on this parameter. This option does not grant the owner any special privileges relative to the rule set. You can enter the logonid, UID, name of the owner, or other values for local tracking purposes. This information is displayed when the rule set is decompiled.

Resowner

Specifies the resource owner of the data set. CA ACF2 for VM logonid DFSMS support uses this information. CA ACF2 for VM does not use it. For more information, see the *Administrator Guid*e.

Mode

Specifies the value of the mode for the rule set. Valid options are QUIET (the access is not checked), LOG (violations are allowed but logged), WARN (violators are warned, but the violation access is allowed), and ABORT (violations are not allowed). Use the $MODE option as a transition aid to phase in protection of data on a rule set basis. Two conditions must be met before the $MODE option is in effect:

- You must specify the MODE(RULE,no-rule,no-$mode) option in the OPTS VMO record.
- A prevent condition must occur when CA ACF2 for VM logonid validates access to data this rule set protects.

Userdata

Use this space for any comments you care to enter. This information does not affect the rule. This field can contain up to 64 characters. Information you enter in this field is stored with the rule set.

Sort Option

Specifies how CA ACF2 for VM logonid sorts this rule set. If you enter 1 or Y in this field, CA ACF2 for VM logonid does not sort the rule set from most specific to most general. This field does not appear if sort option is not active at your site.

Total rule entries

Specifies the number of rule entries currently in the rule.

Press PF11 (NEXT) to add additional information to this rule set.

Adding VM Access Rule Entry Lists

To see this screen, press PF11 (NEXT) after you have viewed the Add Access Rule Set Control Information screen. Use this screen to add VM rule entries.

To add OS/390 or VSE rule entries, press PF6.

```
M9PA-2123   Add VM Access Rule Entry List (2.1.2.3)     CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                            TIME 17:12
 Rulekey ==> ________                Entry  1 of  3

This ruleset also has MVS-style rules not shown here. Press PF6 to view.
 ___                                                                  WRE
A001 Addr: V0191 Fn:      PROFILE Ft: EXEC   Uid: SHSADTLCAMS_____________ A
     Fpool: ________ Dir: ____________________________________________
                          ____________________________________________
                          ____________________________________________
 002 Addr: _____ Fn:      /_______      Ft: ________ Uid: *****PER________ AA
     Fpool: SYSPROG  Dir: TLCMAS.ACF2.SECURITY.REPORTS
                          ____________________________________________
                          ____________________________________________
 003 Addr: _____ Fn:      PROFILE Ft: XEDIT  Uid: ******TLC_____________ A
     Fpool: TECH     Dir: UNITED.STATES.ILLINOIS.COOK.CHICAGO.STREET
                          NUMBER.FLOOR_____________________________
                          ____________________________________________



PF1=Help       2=Print       3=Quit       4=Return     5=Execute   6=MVS<->VM
PF7=Backward   8=Forward     9=Director   10=Previous   11=Next     12=Retrieve
```

Enter an S (for SELECT) in the prefix area to see all of the details on the rule entry.

A

Indicates the prefix area for this screen. The prefix area is where you select rules or enter information.

Following is a brief description of the fields on this screen.

Rulekey

Specifies the $KEY value of the rule set. Normally, the rulekey is the ID of a VM user who has one or more Shared File System (SFS) files or directories, or minidisks specified in the VM directory under his user entry. In most cases, this is also the user's CA ACF2 for VM logonid logonid. The rulekey can also be the high-level qualifier of an OS/390 or VSE data set. The rule ID specified can be up to eight characters long for access rules. You cannot mask this field.

CA ACF2 for VM logonid displays the address, filename, filetype, SFS filepool and SFS directory name as it currently exists. The UID displays the user that the rule is written for.

Use the following commands in the prefix area to display rule entries:

A (After)

Inserts the moved rule entry after this line (same as F).

B (Before)

Inserts the moved rule entry before this line (same as P).

D (Delete)

Deletes the rule entry on this line.

F (Following)

Inserts the moved rule entry following this line (same as A).

I (Insert)

Inserts a rule entry following this line.

M (Move)

Moves this rule entry.

P (Previous)

Inserts the moved rule entry previous to this line (same as B).

S (Select)

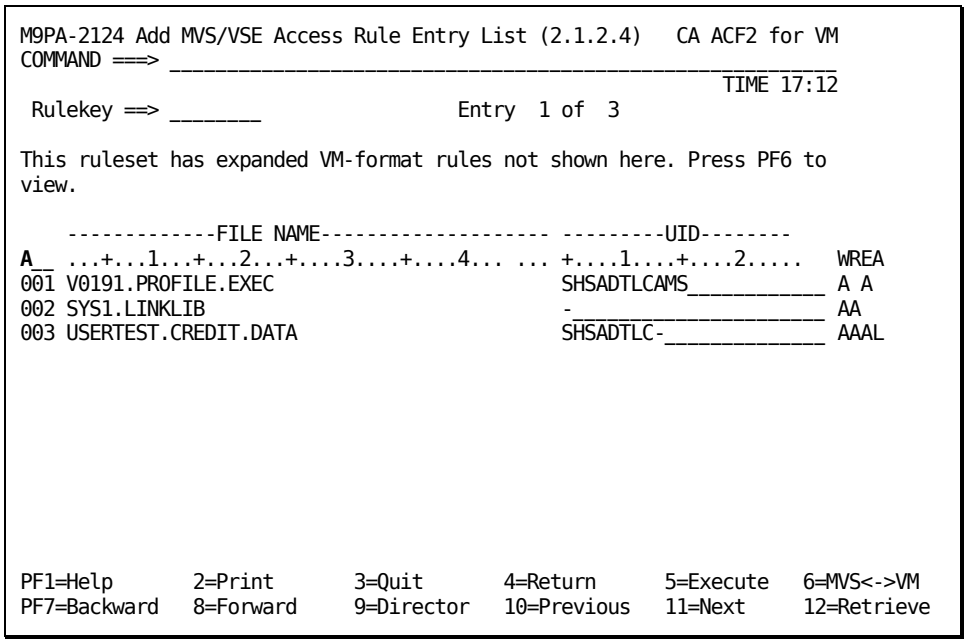
Selects this rule entry. CA ACF2 for VM logonid checks the rule to see if it is formatted like a VM rule or an OS/390 rule, then CA ACF2 for VM logonid displays the appropriate screen with additional information.

The three characters at the end of the line indicate the various access types. They stand for W=Write, R=Read, and E=Execute, respectively. Enter the access permission to be granted (A=Allow, L=Log, P=Prevent) under the appropriate access type. The above panel displays how you might define access privileges.

The user with a UID of SHSADTLCAMS has WRITE access to the PROFILE EXEC on the 191 disk. All users whose logonid starts with PER can READ and EXECUTE the TLCMAS.ACF2.SECURITY.REPORTS files in the SYSPROG filepool. Also, all users whose logonid begins with TLC can write to the PROFILE XEDIT file in the SFS directory named UNITED.STATES.ILLINOIS. COOK.CHICAGO.STREET.NUMBER.FLOOR in the TECH filepool.

Adding OS/390 and VSE Access Rule Entry Lists

To see this screen, press PF11 (NEXT) after you have viewed the Add Access Rule Set Control Information screen. Use this screen to add OS/390 and VSE rule entries.



This screen contains only summary information of the various rule entries. Enter an S (for SELECT) in the prefix area to see all of the details on the rule entry.

A

Indicates the prefix area for this screen. The prefix area is where you select rules or enter information.

Following is a brief description of the fields on this screen.

Rulekey

Specifies the $KEY value of the rule set. Normally, the rulekey is the ID of a VM user who has one or more MDISKs specified in the VM directory under his user entry. In most cases, this is also the user's CA ACF2 for VM logonid logonid. The rulekey can also be the high-level qualifier of an OS/390 or VSE data set. The rule ID specified can be up to eight characters long for access rules. You cannot mask this field.

CA ACF2 for VM logonid displays the data set and UID for this rule set as it currently exists. The information shown under FILE NAME displays the name of the data set. The UID displays the user that the rule is written for.

Use the following commands in the prefix area to display rule entries:

A (After)

Inserts the moved rule entry after this line (same as F).

B (Before)

Inserts the moved rule entry before this line (same as P).

D (Delete)

Deletes the rule entry on this line.

F (Following)

Inserts the moved rule entry following this line (same as A).

I (Insert)

Inserts a rule entry following this line.

M (Move)

Moves this rule entry.

P (Previous)

Inserts the moved rule entry previous to this line (same as B).

S (Select)

Selects this rule entry. CA ACF2 for VM logonid checks the rule to see if it is formatted like a VM rule or an OS/390 rule, then CA ACF2 for VM logonid displays the appropriate screen with additional information.

The four characters at the end of the following line indicate the various access types:

```
___ ...+...1....+...2....+...3....+...4... ...+...1...+....2.WREA
```

They stand for W=Write, R=Read, E=Execute, and A=Allocate, respectively. Enter the access permission to be granted (A=Allow, L=Log, P=Prevent) under the appropriate access type.

Following is an example of how you might define access privileges.

```
___ ....+...1....+...2....+...3....+...4....  ...+...1...+....2.  WREA
001 V0191.VOLUME                              SHSADTLC-__________ AL
```

All users with a UID beginning SHSADTLC have READ access to V0191.VOLUME. All EXECUTES on this volume are LOGGED. You did not specify an access permission for WRITE. Because CA ACF2 for VM logonid provides protection by default, no permission implies PREVENT.

Adding Rule Entries for VM Data

You see this screen if the rule set you selected on the previous screen (Add Access Rule Entry List) is a VM rule set. Use this screen to add access rules for VM minidisks or Shared Files System (SFS) files and directories.

```
M9PA-2126   Add Rule Entry For VM Data (2.1.2.6)        CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 17:12
Rulekey           ==> ________           Entry ___ of ___
Minidisk Address  ==> _____         SFS Filepool ==> ________
Filename/"VOLUME" ==> ________       Filetype    ==> ________
SFS Directory     ==> ____________________________________________
             ____________________________________________
             ____________________________
UID String        ==> ____________________  Source    ==> ________
Shift             ==> ________              Nextkey   ==> ________
Access valid until ==> ________   Access from program ==> ________

Access (Allow/Log/Prevent):
               WRITE     ==> _______
               READ      ==> _______
               EXECUTE   ==> _______

Data   ==> ____________________________________________________________


PF1=Help       2=Print       3=Quit       4=Return      5=Execute   6=MVS<->VM
PF7=Backward   8=Forward     9=Director   10=           11=         12=Retrieve
```

Following is a brief description of the fields on this screen.

Rulekey

Specifies the key value of the rule set. For minidisk access rules, the rulekey is the same as the user ID specified in the VM directory entry that contains the minidisk. The rulekey specified can be up to eight characters long. You cannot mask this field.

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in the rule set.

Minidisk address

Specifies the address of the minidisk the rule applies to (for example, V0191) where

R

Specifies real DASD.

V

Specifies virtual address. This is the default.

Filepool

Specifies the Shared File System (SFS) filepool identifier this rule applies to.

SFS Directory

Specifies the Shared File System (SFS) directory identifier this rule applies to.

Filename/"VOLUME"

If the rule applies to a minidisk, enter VOLUME for the filename.

Filetype

Specifies the filetype of the file that this rule applies to.

UID string

Specifies the User Identification string of the user this rule entry applies to, or a pattern specifying the set of users that this rule should apply to. If you omit this field, the rule entry applies to all users (optional).

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user is logged onto that terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names.

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access this rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times (optional).

Nextkey

Specifies the rule ID of the next (or alternate) rule set that should be checked for this access. If access to this file or data set is denied based on the rule set environment and access permissions in the original rule, CA ACF2 for VM logonid proceeds to the rule specified in the NEXTKEY operand for further checking (optional).

Data

Indicates any character string (up to 64 characters) retained with the rule set and formatted when the rule set is decompiled. Your site might have standards for formatting this string. Standard CA ACF2 for VM logonid does not use values in the string, but they can be meaningful in your local implementation of CA ACF2 for VM through special program exit checking (optional).

Access valid until

Indicates the last date that this rule entry is valid. Valid input is in the Gregorian date form (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending on the DATE parameter of the OPTS VMO record (optional).

Access from Program

Specifies the program name that accesses the file or volume.

Access: (ALLOW|LOG|PREVENT)

Specifies the type of access that applies to this user. Valid options are ALLOW (allow the access to the file or volume), LOG (allow the access, but record the event), and PREVENT (do not allow the access). PREVENT is the default.

You can allow, log, or prevent users from having these types of access: WRITE (if allowed, the user can write to the file), READ (if allowed, the user can read the file), and EXECUTE (the user can execute the file).

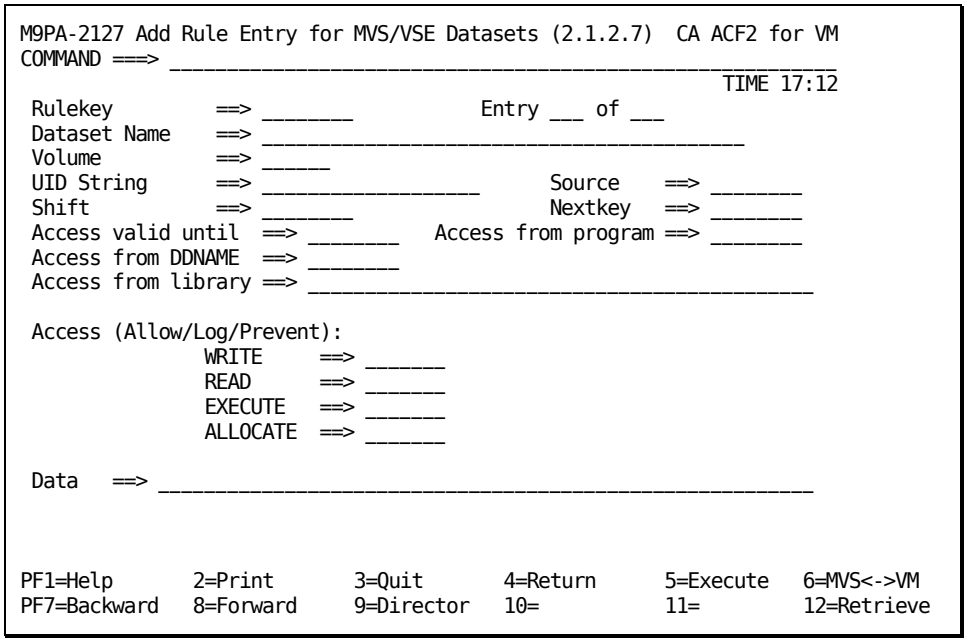
You can mix the access permissions (ALLOW|LOG|PREVENT) with the access types (WRITE|READ|EXECUTE). For example, you can let a user read and execute a file, but not write to it. Do this by specifying:

WRITE==> P
READ ==> A
EXECUTE==> A

READ also implies EXECUTE.

Adding Rule Entries for OS/390 or VSE Data Sets

You see this screen if the rule set you selected on the previous screen (Add Data Set Rule Entry List) was an OS/390 or VSE rule set.



Following is a brief description of the fields on this screen.

Rulekey

Specifies the high-level qualifier of the data set name. The rulekey can be up to eight characters long. You cannot mask this field.

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Data set name

Specifies the rest of the name of the data set.

Volume

Specifies the volume of the data set.

UID string

Specifies the User Identification string of the user that this rule entry applies to.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user is logged onto that terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access the rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times.

Nextkey

Specifies the rule ID of the next rule set to be checked for this access. If CA ACF2 for VM logonid denies access access to this file or data set based on the rule set environment and access permissions in the original rule, CA ACF2 for VM logonid proceeds to the rule specified in the NEXTKEY operand for further checking (optional).

Data

Indicates any character string (up to 64 characters) retained with the rule set and formatted when the rule set is decompiled. Your site might have standards for the formatting of this string. Standard CA ACF2 for VM logonid does not use values in the string, but they can be meaningful in your local implementation of CA ACF2 for VM through special program exit checking (optional).

Access valid until

Specifies the last date that this rule entry is valid. Valid input is in the Gregorian date form (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending the DATE parameter of the OPTS VMO record.

Access from Program

Specifies the program name that accesses the file or volume.

Access from DDNAME

Specifies the ddname that this user must use to access the data set.

Access from Library

Specifies the name of the library that the program must execute out of. (This field is only valid for OS/390 and VSE data sets.)

Access: (ALLOW|LOG|PREVENT)

Specifies the type of access that applies to this user. Valid options are ALLOW (allow the access to the file or volume), LOG (allow the access, but record the event), and PREVENT (do not allow the access). PREVENT is the default.

You can allow, log, or prevent users from having these types of access: WRITE (if allowed, the user can write to the file), READ (if allowed, the user can read the file), and EXECUTE (the user can execute the file).

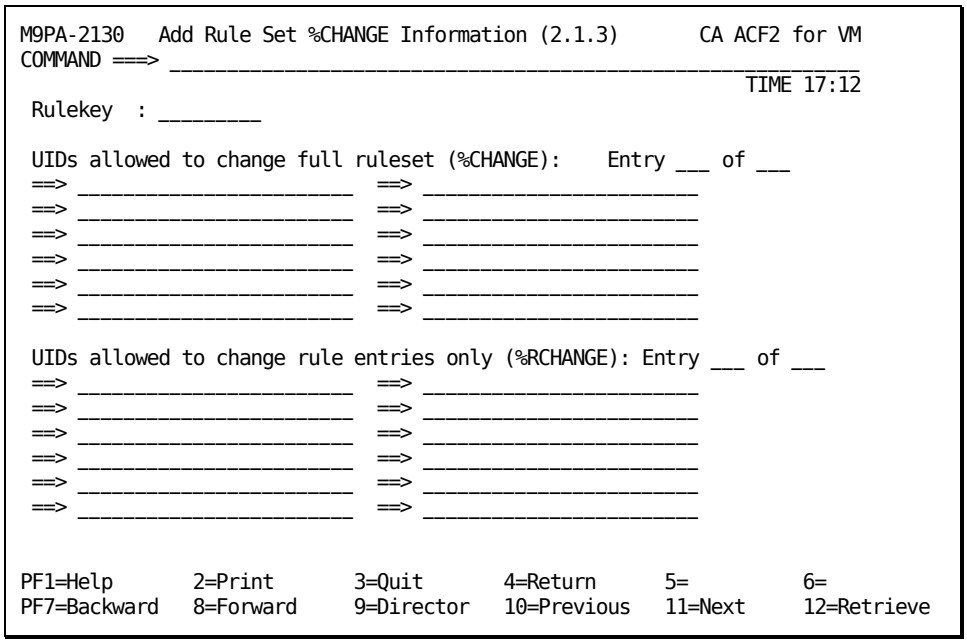
You can mix the access permissions (ALLOW|LOG|PREVENT) with the access types (WRITE|READ|EXECUTE|ALLOCATE). For example, you can let a user read and execute a file, but not write to it. Do this by specifying: :

WRITE==> P
READ ==> A
EXECUTE==> A

READ also implies EXECUTE.

Adding Rule Set %CHANGE Information

To see this screen, press Enter after you have viewed the Add Rule Entry for a VM Minidisk or Add Rule Entry for MVS/VSE Data Sets screens. Use this screen to grant authority to users so that they can make changes to the specified rule set.



Following is a brief description of the fields on this screen.

Rulekey

Specifies the key of the rule set. Normally, the rulekey is the ID of a VM user who has one or more MDISKs specified in the VM directory under his user entry. In most cases, this is also the user's CA ACF2 for VM logonid logonid. The rulekey can also be the high-level qualifier of an OS/390 or VSE data set. The rule ID can be up to eight characters long. You cannot mask this field.

UIDS allowed to change full rule set (%CHANGE)

Specifies the UIDs of users with full change authority. These users can make changes to the specified rule set. This control statement indicates who (besides the data owner or security administrator) can replace a particular set of rules. A security administrator can compile a rule set with only $KEY and %CHANGE control statements, establishing a base rule set to distribute rule writing permissions.

UIDS allowed to change rule entries only (%RCHANGE)

Specifies the UIDs of users who have limited change authority. These users can make rule entries, but cannot delegate the same authority to anyone else. The designated users cannot change control statements. They cannot further delegate change authority or delete the rule set. If the same user matches entries in both %CHANGE and %RCHANGE, the higher authority (%CHANGE) is in effect for that user (optional).

Entry ___ of ___

Indicates the %CHANGE and %RCHANGE UIDs listed first for this screen and the total number of %CHANGE and %RCHANGE entries found.

Use your backward and forward PF keys to add additional entries.

Displaying an Access Rule

This section explains how you can display information about selected users. You must have the necessary authorization to display user information.

When you have finished this section, you will know how to use the full-screen feature to display:

- Summary and detailed access rule information
- VM minidisk access rules
- OS/390 and VSE data set access rules
- Who can change specific access rules.

Displaying Access Rule Sets

You see this screen if you selected option 2, Display Access Rule Set(s) from the Data Access Control (2) screen. This screen displays summary information about compiled rule sets and lets you select a rule for complete display.

```
M9PA-2200   Display Access Rule Set(s) (2.2)        CA ACF2 for VM
COMMAND ===> ____________________________________________________
                                                      TIME 17:12
Rulekey ==>

Enter 's' to select a full display:                    Page

    KEY          COMPILED BY    LAST COMPILED     LENGTH %UTIL
 ==> _ COMMON    COMMON SOURCE DISK   TLC99COMMON
 ==> _ TLCGLB    GEORGE BAKER         TLC99TLCGLB
 ==> _ TLCPJM    PETE J. MILLER       TLC99TLCPJM
 ==> _ TLCAMS    ANN M. SMITH         TLC99TLCAMS
 ==> _ ________  ____________________ ________________________




PF1=Help       2=Print       3=Quit       4=           5=          6=
PF7=Backward   8=Forward     9=Director   10=          11=         12=Retrieve
```

This screen is not displayed if you passed an unmasked rulekey from the menu screen or if the key had no masking. If you did not specify a rulekey on the initial display of this screen, lines 6 through 21 are not displayed.

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Here is a short description of the type of information that might be displayed.

Key

Specifies the $KEY of the rule set

Compiled by

Specifies the logonid of the user that last compiled the rule set

Last compiled

Specifies the date and time the rule set was last compiled

Length

Specifies the length of the rule set

%UTIL

Specifies the percentage of the database used.

To get more information on any of the displayed rule sets, enter an S (for SELECT) on the field before the KEY and press Enter. CA ACF2 for VM logonid displays another screen with more detailed information.

Displaying Access Rule Set Control Information

This screen displays more detailed information on the rule set that you selected on the Display Access Rule Set screen. This is a display only screen.

```
M9PA-2210  Display Access Ruleset Control Info. (2.2.1)  CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                            TIME 12:43
Rulekey     : ________                     Rule length    : ___
Prefix      : ________________________     % Utilization   : __
Owner       : ________________________  Total Rule Entries : ___
Resowner    : ________
Mode        : _____
Userdata    : ____________________________________________________
Sort Option :  _


Use the 'Next' function to see more rule data.




PF1=Help       2=Print       3=Quit       4=Return      5=          6=
PF7=           8=            9=           10=Previous   11=Next     12=Ret
```

Below is a brief explanation of the information on this screen.

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Rule length

Specifies the length of the compiled rule set.

Prefix

Specifies the ID associated with this user's owned minidisks. The user's access to CMS files on a minidisk whose ID matches the prefix is always granted. The prefix also identifies the access rule sets the user can decompile and store. By default, PREFIX is the same as the user's logonid. You can mask with asterisks (*), but not with dashes (-) This PREFIX can be up to eight characters long.

% Utilization

Specifies the amount of the database used.

Owner

Specifies the ID of the owner of this rule set. This control statement is for informational purposes only. No CA ACF2 for VM logonid processing is based on this parameter. This control statement does not grant the owner any special privileges for the rule set. You can enter the logonid, UID, name of the owner, or other values (up to 24 characters) for local tracking purposes. This information is displayed when the rule set is decompiled (optional).

Resowner

Specifies the resource owner of the data set. CA ACF2 for VM logonid DFSMS support uses this information. CA ACF2 for VM does not use it.

Total rule entries

Specifies the number of rule entries currently in the rule.

Mode

Specifies the value of the mode for the rule set. Valid options are:

QUIET

The access is not checked

LOG

Violations are allowed but logged

WARN

Violators are warned, but the violation access is allowed

ABORT

Violations are not allowed.

Sort Option

Specifies how CA ACF2 for VM logonid sorts this rule set. If you enter 1 or Y in this field, CA ACF2 for VM logonid does not sort the rule set from most specific to most general. This field does not appear if sort option is not active at your site.

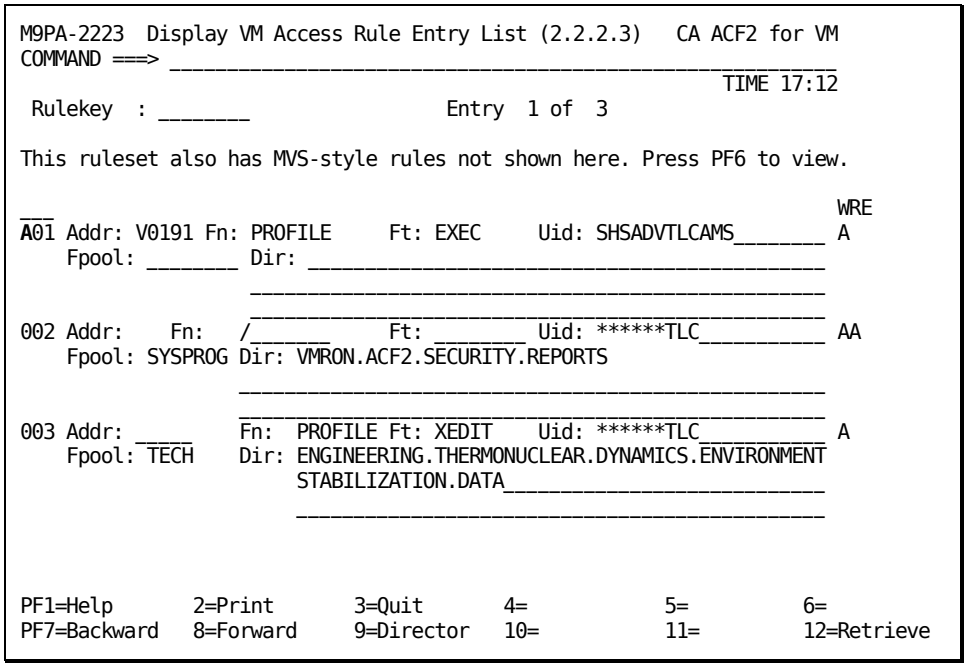
Userdata

Use this space for any comments you care to enter. These comments do not affect the rule entry. This field can contain up to 64 characters. Unlike comment cards that are removed when the rule is compiled, information placed in this field is stored with the rule set (optional).

Displaying VM Access Rule Entry Lists

This screen displays summary VM rule entry information for the specified RULEKEY. This is a display-only screen. To get additional information on any of the rule entries, enter an S (for SELECT) in the prefix area and press Enter. An additional screen is displayed.

To display OS/390 or VSE rule entries, press PF6.



A

Indicates the prefix area for this screen. The prefix area is where you select rules or enter information.

Observe the three characters at the end of the line. These characters stand for the type of access to be granted, respectively W=Write, R=Read, and E=Execute.

All users with a logonid beginning TLC have READ access to V0191.VOLUME. All executes on this volume are logged. No access permission has been granted for WRITE. Because CA ACF2 for VM logonid provides protection by default, granting no permission implies PREVENT.

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Displaying OS/390 and VSE Access Rule Entry Lists

This screen displays summary OS/390 and VSE rule entry information for the specified RULEKEY. This is a display-only screen. To get additional information on any of the rule entries, enter an S (for SELECT) in the prefix area and press Enter. An additional screen is displayed.

```
M9PA-2224 Display MVS/VSE Access Rule Entry List (2.2.2.4) CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                              TIME 17:12
 Rulekey  : ________                Entry  1 of  3

This ruleset has expanded VM-format rules not shown here. Press PF6 to
view.

    ------------FILE NAME---------------------- ----------UID----------
A__ ....+...1....+....2....+....3....+....4.... ....+....1....+....2.. WREA
001 V0191.PROFILE.EXEC                          SHSADTLCAMS___________ A A
002 SYS1.LINKLIB                                -_____________________ AA
003 USERTEST.CREDIT.DATA                        SHSADSVM-_____________ AAAL









PF1=Help       2=Print      3=Quit       4=Return     5=           6=MVS<->VM
PF7=Backward   8=Forward    9=Director   10=Previous   11=Next     12=Retrieve
```

A

Indicates the prefix area for this screen. The prefix area is where you select rules or enter information.

Observe the four characters at the end of the following line:

```
___ ....+....1....+....2....+....3....+.... ....+....1....+....2.RWEA
```

These characters stand for the type of access to be granted, respectively W=Write, R=Read, E=Execute, and A=Allocate.

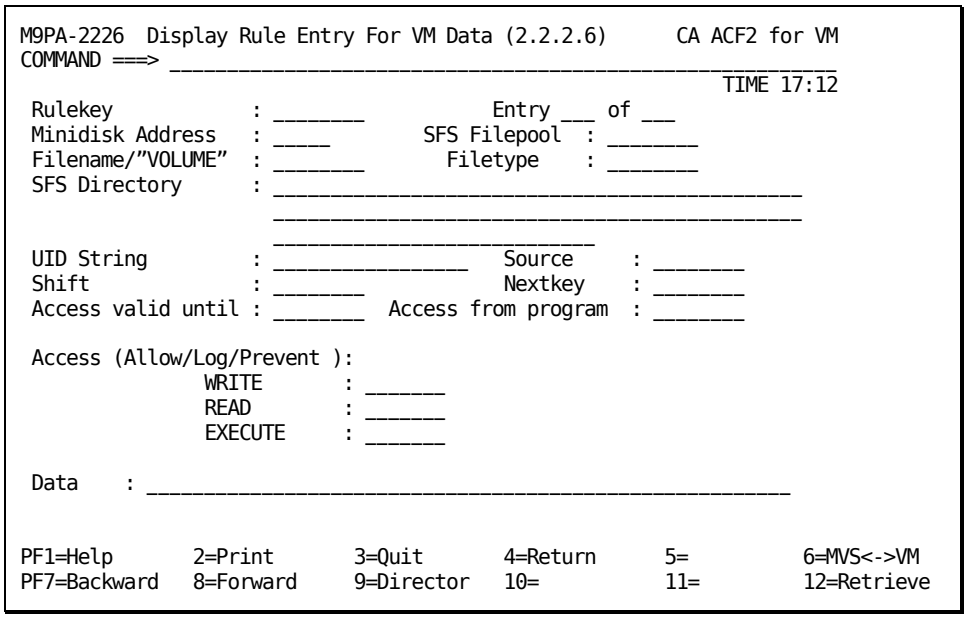
All users with a logonid beginning TLC have READ access to V0191.VOLUME. All executes on this volume are logged. No access permission has been granted for WRITE. Because CA ACF2 for VM logonid provides protection by default, granting no permission implies PREVENT.

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Displaying Rule Entries for VM Data

This screen displays information about a VM minidisk.



Following is a brief explanation of the fields on this screen.

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Minidisk address

Specifies the address of the minidisk the rule applies to (for example, V0191).

R

Specifies real DASD.

V

Specifies virtual address. This is the default.

SFS Filepool

Specifies the Shared File System (SFS) filepool identifier this rule applies to.

Filename/"VOLUME"

Specifies the filename of the file this rule applies to. If the rule applies to a minidisk, VOLUME is the filename.

Filetype

Specifies the filetype of the file this rule applies to.

SFS Directory

Specifies the Shared File System (SFS) directory identifier this rule applies to.

UID String

Specifies the User Identification string of the user that this rule entry applies to.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access this rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times.

Nextkey

Specifies the rule ID of the next rule set that CA ACF2 for VM logonid checks for access. If access to this file is denied based on the rule set environment and access permissions in the original rule, CA ACF2 for VM logonid proceeds to the rule specified in the NEXTKEY operand for further checking (optional).

Access valid until

Specifies the last date that this rule entry is valid. Valid input is in the Gregorian date format (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending the DATE parameter of the OPTS VMO record (optional).

Access from Program

Specifies the program name that accesses the file or volume.

Access: (ALLOW|LOG|PREVENT)

Specifies the type of access that applies to this user. Valid options are ALLOW (allow the access of the file or volume), LOG (allow the access, but record the event), and PREVENT (do not allow the access). PREVENT is the default.

You can allow, log, or prevent users from having the following types of access: WRITE (if allowed, the user can write to the file), READ (if allowed, the user can read the file), ALLOCATE (if allowed, user can create, rename, delete, and catalog a dataset, and EXECUTE (the user can execute the file). You can mix the access permissions (ALLOW|LOG|PREVENT) with the access types (WRITE|READ|EXECUTE). For example, you can let a user read and execute a file, but not write to it. This is accomplished by specifying:

```
WRITE  :P
READ   :A
EXECITE :A
```

READ also implies EXECUTE.

Data

Enter up to 64 characters in this field that are retained with the rule set and formatted when the rule set is decompiled. Your site might have standards about the format of this field. Standard CA ACF2 for VM logonid does not use values in the field, but they can be meaningful in your local implementation of CA ACF2 for VM through special program exit checking (optional).

Displaying OS/390 or VSE Data Set Rule Entries

This screen is displayed if the rule set you selected on the Display Data Set Rule Entry List screen was an OS/390 or VSE rule set.

```
M9PA-2227 Display Rule Entry for MVS/VSE Datasets (2.2.2.7) CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 17:12
 Rulekey             : ________           Entry ___ of ___
 Dataset Name        : ____________________________________________
 Volume              : ______
 UID String          : ________________  Source     : ________
 Shift               : ________          Nextkey    : ________
 Access valid until  : ________   Access from program : ________
 Access from DDNAME  : ________
 Access from library : ___________________________________________

 Access (Allow/Log/Prevent):
                WRITE      : _______
                READ       : _______
                EXECUTE    : _______
                ALLOCATE   : _______

 Data    : ____________________________________________________________

PF1=Help       2=Print      3=Quit       4=Return     5=          6=MVS<-VM
PF7=Backward   8=Forward    9=Director   10=          11=         12=Retrieve
```

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Data set name

Specifies the name of the data set.

Volume

Specifies the volume of the data set.

UID string

Specifies the User Identification string of the user that this rule entry applies to.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access the rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times.

Nextkey

Specifies the rule ID of the next rule set that CA ACF2 for VM logonid checks for access. If access to this file or data set is denied based on the rule set environment and access permissions in the original rule, CA ACF2 for VM logonid proceeds to the rule specified in the NEXTKEY operand for further checking (optional).

Data

Enter up to 64 characters in this field that are retained with the rule set and formatted when the rule set is decompiled. Your site might have standards about the format of this field. Standard CA ACF2 for VM logonid does not use values in the field, but they can be meaningful in your local implementation of CA ACF2 for VM through special program exit checking (optional).

Access valid until

Specifies the last date that this rule entry is valid. Valid input is in the Gregorian date format (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending upon the DATE parameter of the OPTS VMO record (optional).

Access from Program

Specifies the program name that accesses the file or volume.

Access from DDNAME

Specifies the ddname this user must use to access the data set.

Access from Library

Specifies the library name this user must use to access the data set.

Access: (ALLOW|LOG|PREVENT)

The type of access that applies to this user. Valid options are ALLOW (allow the access of the file or volume), LOG (allow the access, but record the event), and PREVENT (do not allow the access). PREVENT is the default.

You can allow, log, or prevent users from having the following types of access: WRITE (if allowed, the user can write to the file), READ (if allowed, the user can read the file), ALLOCATE (if allowed, user can create, rename, delete, and catalog a dataset, and EXECUTE (the user can execute the file).

You can mix the access permissions (ALLOW|LOG|PREVENT) with the access types (WRITE|READ|EXECUTE|ALLOCATE). For example, you can let a user read and execute a file, but not write to it. This is accomplished by specifying:

WRITE :P

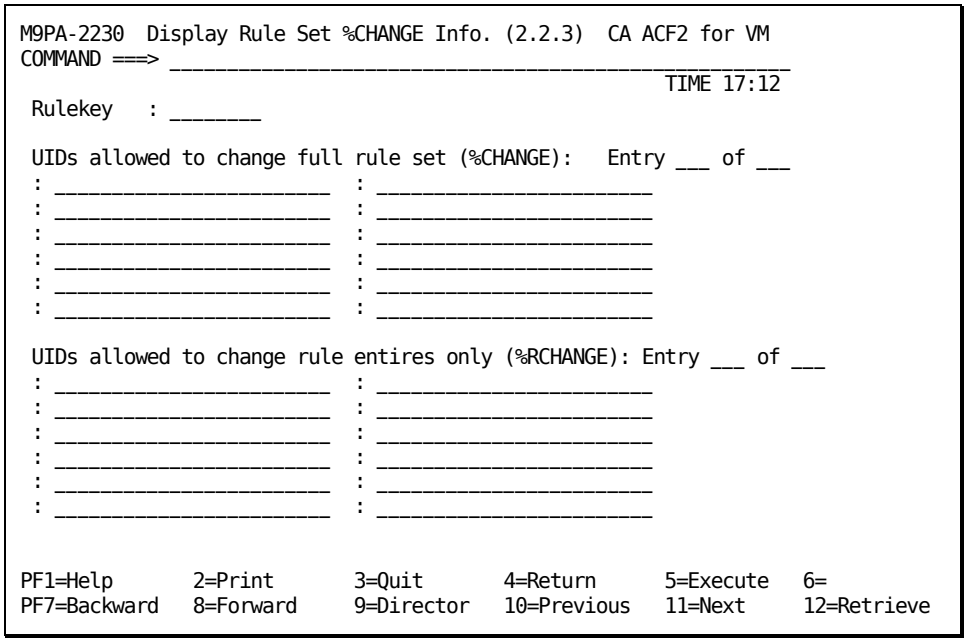
READ :A

EXECUTE :A

READ also implies EXECUTE.

Displaying Rule Set %CHANGE Information

This screen displays those users who are authorized to make changes to the specified rule set. This is a display-only screen.



Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

UIDs allowed to change full rule set (%CHANGE)

Specifies the UIDs of users with full change authority. These users can make any and all changes to the specified rule set. This control statement indicates who (besides the data owner or security administrator) can replace a particular set of rules. A security administrator can compile a rule set with only $KEY and %CHANGE control statements, establishing a base rule set to distribute rule writing permissions (optional).

UIDs allowed to change rule entries only (%RCHANGE)

Specifies the UIDs of users who have limited change authority. These users can change rule entries, but cannot delegate the same authority to anyone else. The designated users cannot change control statements. They cannot further delegate change authority or delete the rule set. If the same user matches entries in both %CHANGE and %RCHANGE, the higher authority (%CHANGE) is in effect for that user (optional).

Entry ___ of ___

Indicates the %CHANGE and %RCHANGE UIDs listed for this screen and the total number of %CHANGE and %RCHANGE entries found.

Use your backward and forward scroll keys to add additional entries.

Changing an Access Rule

This section explains how to use the full-screen feature to:

- Change compiled access rule sets
- Change rule set information (such as prefix, owner, and mode)
- Change rule entries in existing rule sets
- Change VM minidisk rules
- Change OS/390 and VSE data set rule entries
- Redefine who can change specific access rules.

Changing Access Rule Sets

CA ACF2 for VM logonid displays this screen if you selected option 3, Change Access Rule Set(s), from the Data Access Control (2) screen. This screen lets you change compiled rule sets. It is not displayed if you passed an unmasked rulekey from the menu screen, or the key had no masking. If you did not specify a rule key on the initial display of this screen, lines 6 through 21 are not displayed.

```
M9PA-2300   Change Access Rule Set(s) (2.3)          CA ACF2 for VM
COMMAND ===> _____________________________________________________
                                                        TIME 17:12
Rulekey ==> **M-

Enter 's' to select a full display:        Page

       KEY      COMPILED BY  LAST COMPILED   LENGTH %UTIL
 ==> _ TLCAMS   TLISO        09/12/98 12:45  112    20
 ==> _ TLCLSL   TLCLSL       01/11/98 18:34  498    42
 ==> _ COMMON   TLISO        04/23/98 08:23  3092   87
 ==> _ TLCGLB   TLCGLB       09/17/98 17:23  540    34









PF1=Help       2=Print       3=Quit       4=Return      5=Execute   6=
PF7=Backward   8=Forward     9=Director   10=           11=         12=Retrieve
```

Following is a brief explanation of the fields on the screen.

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long.

Key

Specifies the $KEY of the rule set.

Compiled by

Specifies the logonid of the user that last compiled the rule set.

Last compiled

Specifies the date and time the rule set was last compiled.

Length

Specifies the length of the rule set.

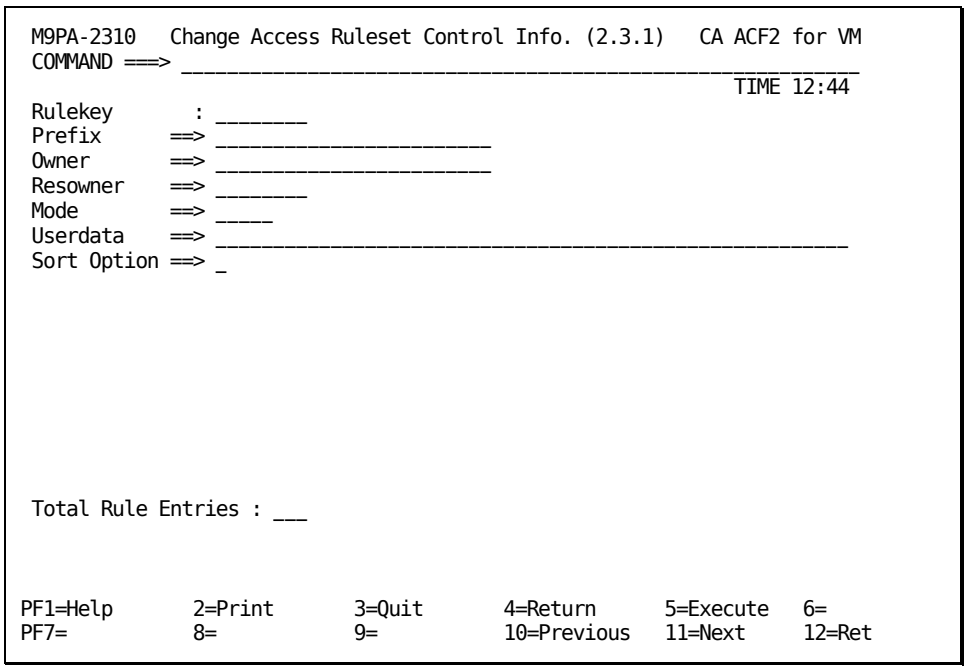
%UTIL

Specifies the percentage of the database used.

To get more information on any of the displayed rule sets, enter an S (for SELECT) on the field before the KEY and press Enter. Another screen is displayed with more detailed information.

Changing Access Rule Set Control Information

This screen displays more detailed information on the rule set that you selected on the Change Access Rule Sets screen and lets you change the information.



Following is a brief explanation of the fields on the screen.

Rulekey

Specifies the key value of the rule set.The rulekey you specify can be up to eight characters long. You cannot mask this field.

Prefix

Indicates a value used as the minidisk ID for CMS files (instead of the rule ID). You can enter up to 24 characters and specify multiple levels in this control statement (TLCAMS.V0191). If you specify $PREFIX(), the prefix is the same as the $KEY entry and the $PREFIX control statement is not generated when the rule is decompiled. CA ACF2 for VM logonid issues a warning message indicating that the $PREFIX is null and ignored (optional).

Owner

Specifies the ID of the owner of this rule set. This control statement is for information only. No CA ACF2 for VM logonid processing is based on this parameter. This control statement does not grant the owner any special privileges for the rule set. You can enter the logonid, UID, name of the owner, or other values (up to 24 characters) for local tracking purposes. These values are displayed when the rule set is decompiled (optional).

Resowner

Specifies the resource owner of the data set. CA ACF2 for VM logonid DFSMS support uses this information. CA ACF2 for VM does not use it. For more information, see the Administrator Guide.

Mode

Specifies the value of the mode for the rule set. Valid options are QUIET (the access is not checked), LOG (violations are allowed but logged), WARN (violators are warned, but the violation access is allowed), and ABORT (violations are not allowed).

Userdata

Use this space for any comments you care to enter. These comments do not affect the rule entry. This field can contain up to 64 characters. Unlike comment statements that are removed when compiled, information placed in this field is stored with the rule set (optional).

Sort Option

Specifies how CA ACF2 for VM logonid sorts this rule set. If you enter 1 or Y in this field, CA ACF2 for VM logonid does not sort the rule set from most specific to most general. This field does not appear if sort option is not active at your site.

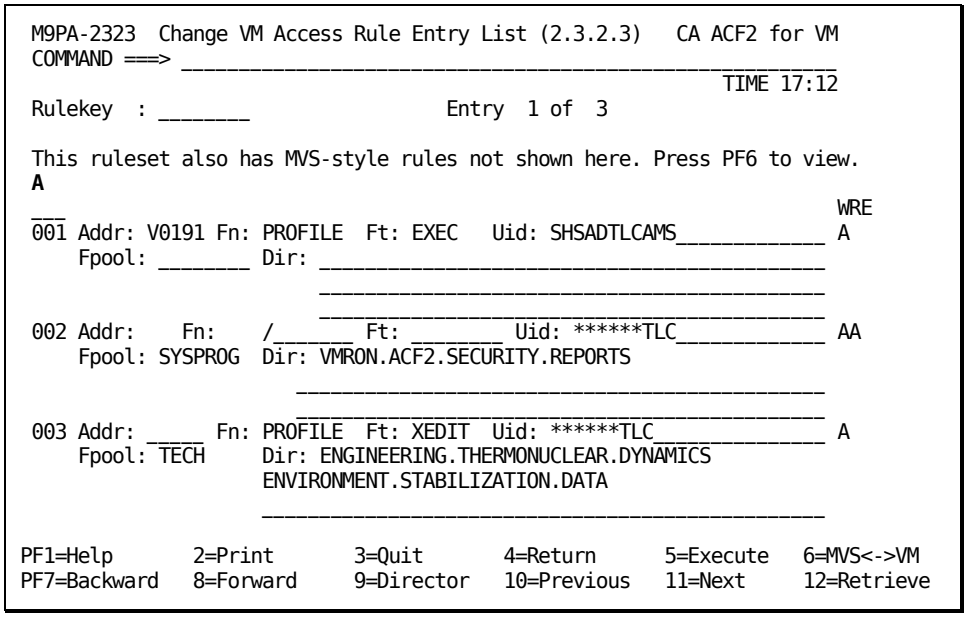
Total rule entries

Specifies the number of rule entries currently in the rule.

Changing VM Access Rule Entry Lists

This screen changes VM rule entries in a rule set that already exists. It contains only summary information of the various rule entries. Enter an S (for SELECT) to see all of the details on the rule set.

To change OS/390 or VSE rule entries, press PF6.



A

Indicates the prefix area for this screen. The prefix area is where you select rules or enter information.

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

CA ACF2 for VM logonid displays the address, filename, filetype, SFS filepool and SFS directory name as it currently exists. The UID displays the user that the rule is written for.

Use the following commands in the prefix area to display rule entries:

A (After)

Inserts the moved rule entry after this line (same as F)

B (Before)

Inserts the moved rule entry before this line (same as P)

D (Delete)

Deletes the rule entry on this line.

F (Following)

Inserts the moved rule entry following this line (same as A)

I (Insert)

Inserts a rule entry following this line.

M (Move)

Moves this rule entry.

P (Previous)

Inserts the moved rule entry previous to this line (same as B)

S (Select)

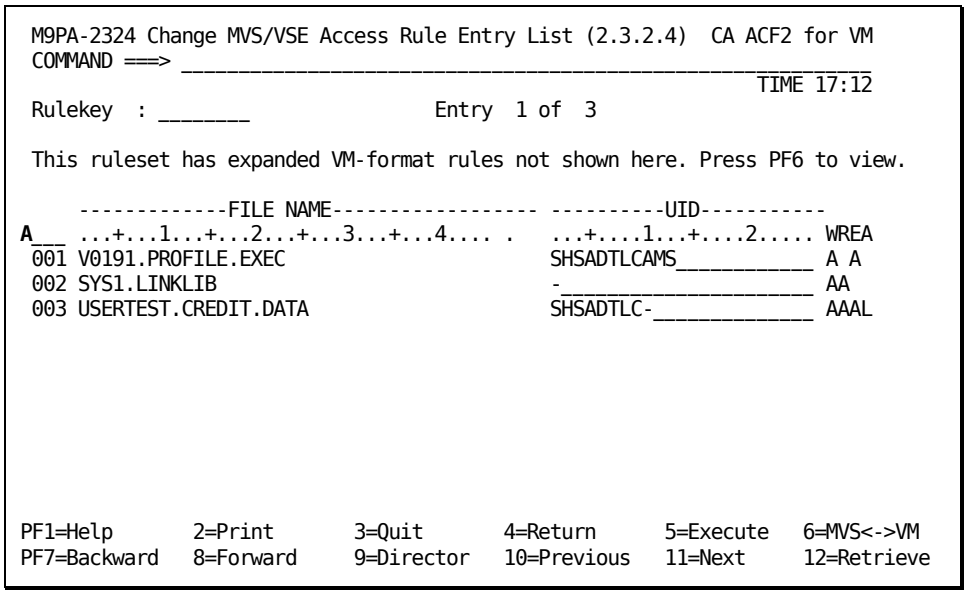
Selects this rule entry. CA ACF2 for VM logonid checks the rule to see if it is formatted like a VM rule or an OS/390 rule, then CA ACF2 for VM logonid displays the appropriate screen with additional information.

The three characters at the end of the line indicate the various access types. They stand for W=Write, R=Read, and E=Execute, respectively. Enter the access permission to be granted (A=Allow, L=Log, P=Prevent) under the appropriate access type. The above panel displays how you might define access privileges.

The user with a UID of SHSADTLCAMS has WRITE access to the PROFILE EXEC on the 191 disk. All users whose logonid starts with PER can READ and EXECUTE the TLCMAS.ACF2.SECURITY.REPORTS files in the SYSPROG filepool. Also, all users whose logonid begins with TLC can write to the PROFILE XEDIT file in the SFS directory named UNITED.STATES.ILLINOIS. COOK.CHICAGO.STREET.NUMBER.FLOOR in the TECH filepool.

Changing OS/390 and VSE Access Rule Entry Lists

This screen changes OS/390 and VSE rule entries in a rule set that already exists. It contains only summary information of the various rule entries. Enter an S (for SELECT) to see all of the details on the rule set.



A

Indicates the prefix area for this screen. The prefix area is where you select rules or enter information.

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

This screen displays the data set and UID information for this rule set as it currently exists. The information displayed under DATASET NAME displays the name of the data set. The UID displays the users the rule has been written for. You can use the following commands in the prefix area to manipulate displayed rule entries:

A (Add)

Inserts the moved rule entry after this line (same as F)

B (Before)

Inserts the moved rule entry before this line (same as P)

D (Delete)

Deletes the rule entry on this line.

F (Following)

Inserts the moved rule entry following this line (same as A)

I (Insert)

Inserts a rule entry following this line.

M (Move)

Moves this rule entry.

P (Previous)

Inserts the moved rule entry before to this line (same as B)

S (Select)

Selects this rule entry. The rule is checked to see if it is formatted like a VM rule or an OS/390 rule; then the appropriate screen is displayed with additional information.

Observe the four characters at the end of the following line:

```
___ ....+....1....+....2....+....3....+...4 .....+....1....+....2.WREA
```

These four characters stand for the type of access to be granted, respectively W=Write, R=Read, E=Execute, and A=Allocate. Enter the access privilege to be granted (A=Allow, L=Log, P=Prevent) under the appropriate access type. Here is an example of how you might define access privileges.

```
___ ....+....1....+....2....+....3....+...4 ....+....1....+....2.WREA
001 V0191.VOLUME                           SHSADTLC-            AL
```

All users with a UID beginning SHSADTLC have READ access to V0191.VOLUME. All executes on this volume are logged. No access permission has been granted for WRITE. Because CA ACF2 for VM logonid provides protection by default, no permission implies PREVENT.

Changing a VM Minidisk Rule Entry

This screen lets you change the VM rule set you selected on the Change Access Rule Entry List screen. To change any of the following information, type over the old value. If the new value is shorter than the old value, erase the rest of the field.

```
M9PA-2326  Change Rule Entry For VM Data (2.3.2.6)     CA ACF2 for VM
COMMAND ===> _____________________________________________________________
                                                              TIME 17:12
Rulekey             : ________          Entry ___ of ___
Minidisk Address   ==> _____          SFS Filepool ==> ________
Filename/"VOLUME"  ==> ________         Filetype   ==> ________
SFS Directory      ==> ____________________________________________
                     ____________________________________________
                     ____________________________________________
UID String          ==> ________________   Source    ==> ________
Shift               ==> ________           Nextkey   ==> ________
Access valid until ==> ________  Access from program ==> ________

Access (Allow/Log/Prevent):
                WRITE     ==> _______
                READ      ==> _______
                EXECUTE   ==> _______

Data    ==> ____________________________________________________


PF1=Help       2=Print       3=Quit       4=Return      5=Execute   6=MVS<->
PF7=Backward   8=Forward     9=Director   10=          11=         12=Retrieve
```

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Minidisk address

Specifies the address of the minidisk the rule applies to (for example, V0191) where:

R

Specifies real DASD

V

Specifies virtual address. This is the default.

Filepool

Specifies the Shared File System (SFS) filepool identifier this rule applies to.

Filename/"VOLUME"

Specifies the filename of the file this rule applies to. If the rule applies to a minidisk, VOLUME is the filename.

Filetype

Specifies the filetype of the file this rule applies to.

SFS Directory

Specifies the Shared File System (SFS) directory identifier this rule applies to.

UID

Specifies the User Identification string of the user that this rule entry applies to.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access the rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times.

Nextkey

Specifies the rule ID of the next rule set that CA ACF2 for VM logonid checks for access. If access to this file or data set is denied based on the rule set environment and access permissions in the original rule, CA ACF2 for VM logonid proceeds to the rule set specified in the NEXTKEY operand for further checking (optional).

Data

Enter up to 64 characters in this field to be retained with the rule set and formatted when the rule set is decompiled. Your site might have standards about the format of this field. Standard CA ACF2 for VM logonid does not use values in the field, but they can be meaningful in your local implementation of CA ACF2 for VM through special program exit checking (optional).

Access valid until

Specifies the last date that this rule entry is valid. Valid input is in the Gregorian date format (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending upon the DATE parameter of the OPTS VMO record (optional).

Access from Program

Specifies the program name that accesses the file or volume.

Access: (ALLOW|LOG|PREVENT)

Specifies the type of access to apply to this user. Valid options are ALLOW (allow the access of the file or volume), LOG (allow the access, but record the event), and PREVENT (do not allow the access). PREVENT is the default.

You can allow, log, or prevent users from having the following types of access: WRITE (if allowed, the user can write to the file), READ (if allowed, the user can read the file), and EXECUTE (the user can execute the file).

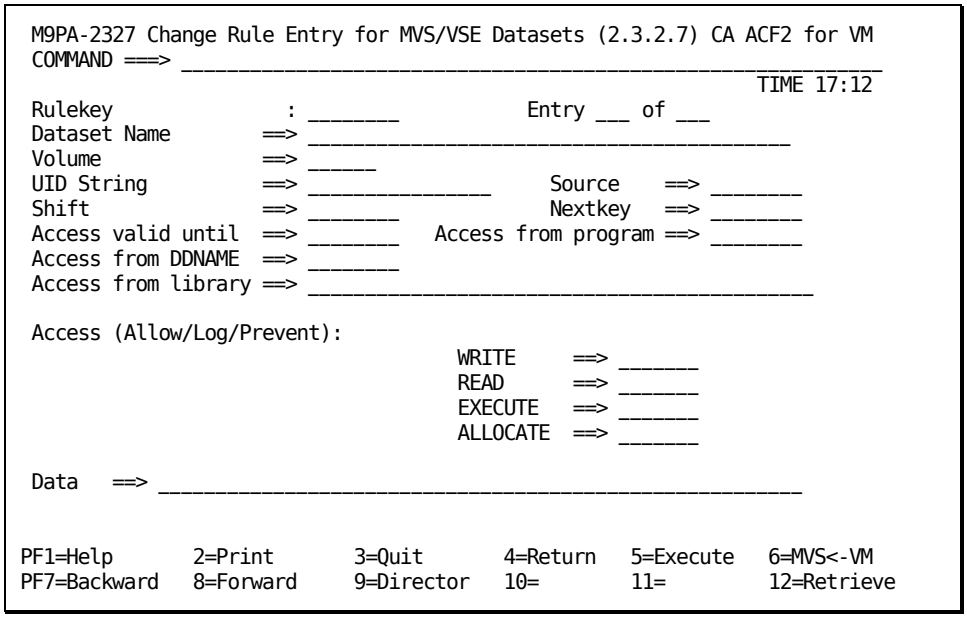
You can mix the access permissions (ALLOW|LOG|PREVENT) with the access types (WRITE|READ|EXECUTE). For example, you can let a user read and execute a file, but not write to it. This is accomplished by specifying:

```
WRITE     :P
READ      :A
EXECUTE  :A
```

READ also implies EXECUTE.

Changing OS/390 or VSE Data Sets

This screen lets you change the OS/390 or VSE rule set you selected on the Change Data Set Rule Entry List screen. To change any of the following information, type over the old value. If the new value is shorter than the old value, erase the rest of the field.



Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Data set name

Specifies the name of the data set.

Volume

Specifies the volume of the data set.

UID

Specifies the user identification string of the user that this rule entry applies to.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access the rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times.

Nextkey

Specifies the rule ID of the next rule set that CA ACF2 for VM logonid checks for access. If access to this file or data set is denied based on the rule set environment and access permissions in the original rule, CA ACF2 for VM logonid proceeds to the rule set specified in the NEXTKEY operand for further checking (optional).

Data

Enter up to 64 characters in this field to be retained with the rule set and formatted when the rule set is decompiled. Your site might have standards about the format of this field. Standard CA ACF2 for VM logonid does not use values in the field, but they can be meaningful in your local implementation of CA ACF2 for VM through special program exit checking (optional).

Access valid until

Specifies the last date that this rule entry is valid. Valid input is in the Gregorian date format (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending upon the DATE parameter of the OPTS VMO record (optional).

Access from Program

Specifies the program name that accesses the file or volume.

Access from DDNAME

Specifies the ddname that this user must use to access the data set.

Access from Library

Specifies the library name that this user must use to access the data set.

Access: (ALLOW|LOG|PREVENT)

Specifies the type of access to apply to this user. Valid options are ALLOW (allow the access of the file or volume), LOG (allow the access, but record the event), and PREVENT (do not allow the access). PREVENT is the default.

You can allow, log, or prevent users from having the following types of access: WRITE (if allowed, the user can write to the file), READ (if allowed, the user can read the file), ALLOCATE (if allowed, user can create, rename, delete, and catalog a dataset, and EXECUTE (the user can execute the file).

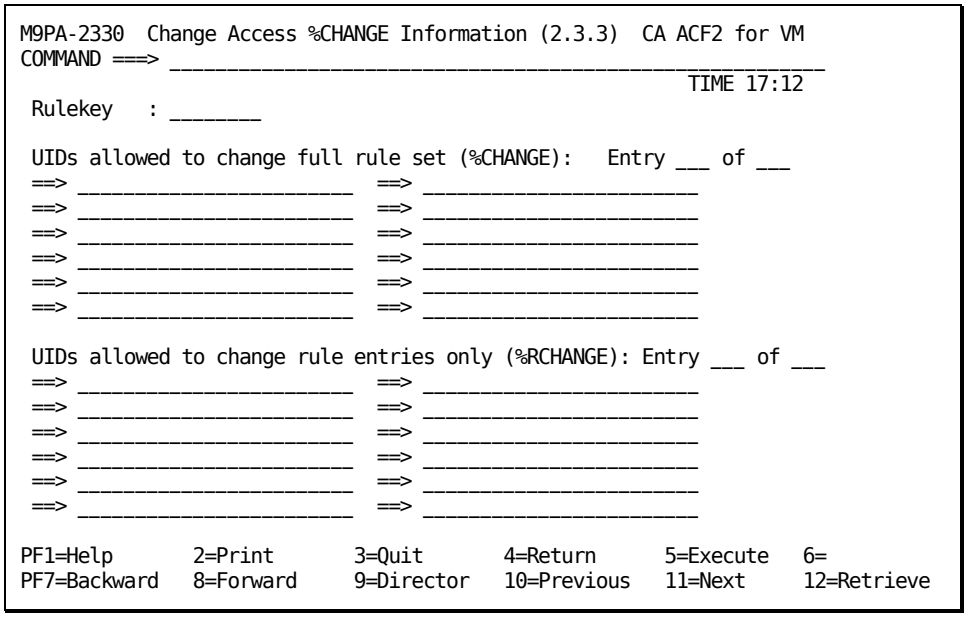
You can mix the access permissions (ALLOW|LOG|PREVENT) with the access types (WRITE|READ|EXECUTE). For example, you can let a user read and execute a file, but not write to it. This is accomplished by specifying:

```
WRITE      :P
READ       :A
EXECUTE :A
```

READ also implies EXECUTE.

Changing Rule Set %CHANGE Information

This screen lets you change the list of users who are authorized to make changes to the specified rule set. To remove users from the list, erase their UIDs.



Rulekey

Specifies the key value of the rule set. The rulekey can be up to eight characters long. You cannot mask this field.

UIDs allowed to change full rule set (%CHANGE)

Specifies the UIDs of users with full change authority. These users can make any and all changes to the specified rule set. This control statement indicates who (other than the data owner or security administrator) can replace a particular set of rules. A security administrator can compile a rule set with only $KEY and %CHANGE control statements, establishing a base rule set to distribute rule writing permissions (optional).

UIDs allowed to change rule entries only (%RCHANGE)

Specifies a list of those users authorized to make changes to this rule set, but cannot delegate this authority.

Entry ___ of ___

Indicates the %CHANGE and %RCHANGE UIDs listed for this screen and the total number of %CHANGE and %RCHANGE entries found.

Use your backward and forward PF keys to add additional entries.

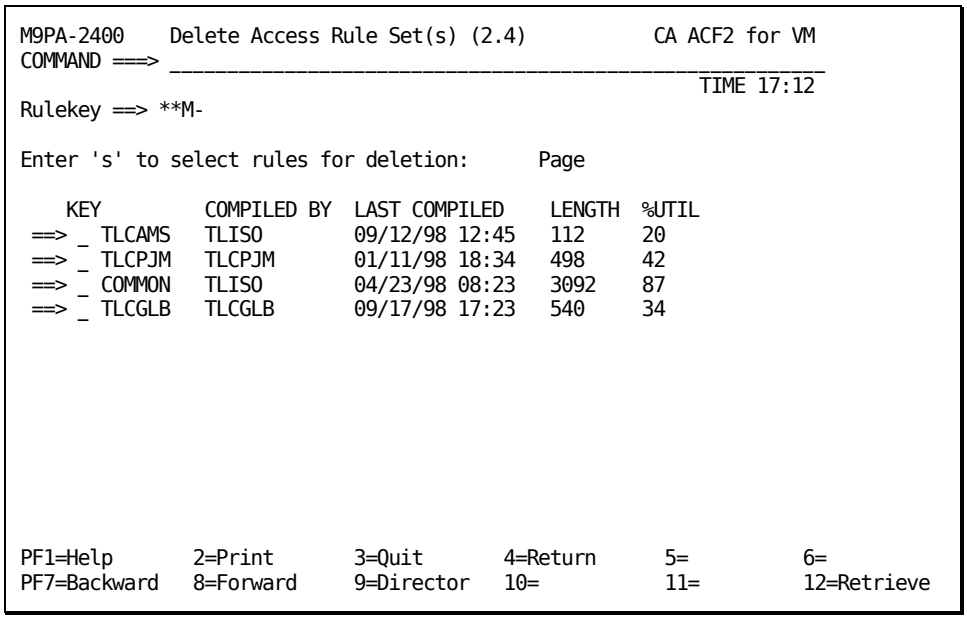
Deleting an Access Rule

This section contains information for using the full-screen feature to:

- Delete access rule sets
- Help you from accidentally deleting rule sets.

Deleting Access Rule Sets

CA ACF2 for VM logonid displays this screen if you selected option 4, Delete Access Rule Set(s), from the Data Access Control (2) screen. This screen lets you delete compiled access rule sets. It is not displayed if you passed an unmasked rule key from the menu screen. If you did not specify a rule key on the initial display of this screen, lines 6 through 21 are not displayed.



```
M9PA-2400    Delete Access Rule Set(s) (2.4)          CA ACF2 for VM
COMMAND ===> ____________________________________________________
                                                        TIME 17:12
Rulekey ==> **M-

Enter 's' to select rules for deletion:      Page

    KEY         COMPILED BY  LAST COMPILED    LENGTH  %UTIL
 ==> _ TLCAMS   TLISO        09/12/98 12:45   112     20
 ==> _ TLCPJM   TLCPJM       01/11/98 18:34   498     42
 ==> _ COMMON   TLISO        04/23/98 08:23   3092    87
 ==> _ TLCGLB   TLCGLB       09/17/98 17:23   540     34









PF1=Help       2=Print       3=Quit       4=Return      5=          6=
PF7=Backward   8=Forward     9=Director   10=           11=         12=Retrieve
```

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long.

Following is a brief explanation of the fields on the screen.

Key

Specifies the $KEY of the rule set.

Compiled by

Specifies the logonid of the user that last compiled the rule set.

Last compiled

Specifies the date and time the rule set was last compiled.

Length

Specifies the length of the rule set.

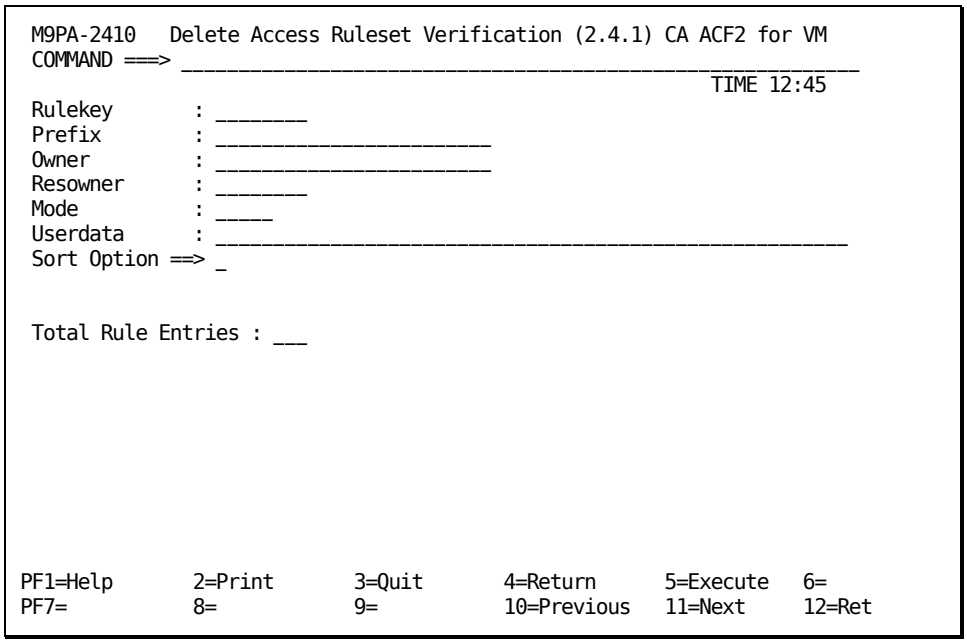
%UTIL

Specifies the percentage of the database used.

To delete a rule set, enter an S on the field before the KEY and press Enter. Another screen is displayed for verification.

Verifying Access Rule Set Deletion

This screen lets you delete the rule set that you selected on the Delete Access Rule Set screen and gives you the chance to verify the action before you delete the rule set. If you accidentally delete a rule set, you must recreate it.



Following is a brief explanation of the fields on the screen.

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Prefix

Indicates that CA ACF2 for VM logonid uses the minidisk ID for CMS files (instead of the rule ID). You can enter up to 24 characters and specify multiple levels in this control statement (TLCAMS.V0191). If you specify $PREFIX(), the prefix is the same as the $KEY entry and the $PREFIX control statement is not generated when the rule is decompiled. CA ACF2 for VM logonid issues a warning message indicating that the $PREFIX is null and ignored (optional).

Owner

Specifies the ID of the owner of this rule set. This control statement is for information only. No CA ACF2 for VM logonid processing is based on this parameter. This control statement does not grant the owner any special privileges relative to the rule set. You can enter the logonid, UID, name of the owner, or other values for local tracking purposes. These values are displayed when the rule set is decompiled (optional).

Resowner

Specifies the resource owner of the data set. CA ACF2 for VM logonid DFSMS support uses this information. CA ACF2 for VM does not use it.

Mode

Specifies the value of the mode for the rule set. Valid options are QUIET (the access is not checked), LOG (violations are allowed but logged), WARN (violators are warned, but the violation access is allowed), and ABORT (violations are not allowed).

Userdata

Use this space for any comments you care to enter. These comments do not affect the rule entry. This field can contain up to 64 characters. Unlike comment cards that are removed when compiled, information placed in this field is stored with the rule set (optional).

Sort Option

Specifies how CA ACF2 for VM logonid sorts this rule set. If you enter 1 or Y in this field, CA ACF2 for VM logonid does not sort the rule set from most specific to most general. This field does not appear if sort option is not active at your site.

Total rule entries

Specifies the number of rule entries currently in the rule.

To delete the rule set, press your designated EXECUTE PF key. CA ACF2 for VM logonid deletes the rule set from the Rule database.

Testing an Access Rule

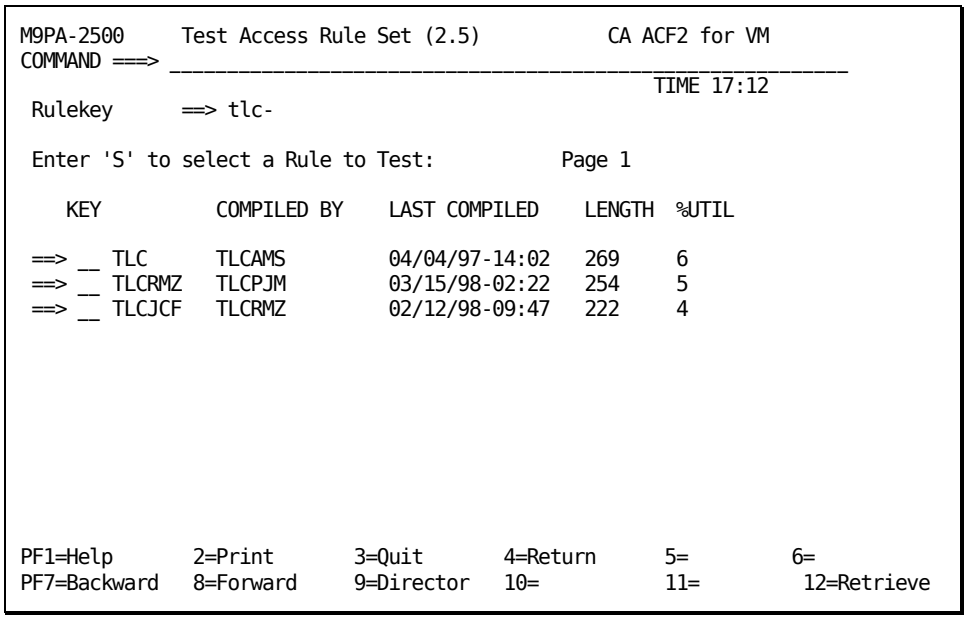
You should develop the habit of testing access rules before you store them to be sure they do what you intended. The screens in this section make testing access rules easy.

When you finish this section, you will know how to use the full-screen feature to:

- Test accesses against a specific rule to be sure the rule does what you want it to do
- Test VM minidisk link rules to determine if the rule allows the link
- Test OS/390 and VSE data set access rules.

Testing an Access Rule Set

You see this screen if you selected option 5, Test an Access Rule Set, from the Data Access Control (2) screen. This screen lets you select access rule sets for testing. To use this screen, enter the KEY (or masks) of the rule sets you want to test.



```
M9PA-2500     Test Access Rule Set (2.5)          CA ACF2 for VM
COMMAND ===> ____________________________________________________
                                                    TIME 17:12
 Rulekey      ==> tlc-

 Enter 'S' to select a Rule to Test:          Page 1

    KEY          COMPILED BY    LAST COMPILED    LENGTH  %UTIL

 ==> __ TLC      TLCAMS         04/04/97-14:02   269     6
 ==> __ TLCRMZ   TLCPJM         03/15/98-02:22   254     5
 ==> __ TLCJCF   TLCRMZ         02/12/98-09:47   222     4









PF1=Help       2=Print       3=Quit       4=Return      5=         6=
PF7=Backward   8=Forward     9=Director   10=           11=         12=Retrieve
```

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You can mask this field with standard masking characters.

If you use masking, you see a list of rule sets to select from. To select a specific rule set, enter an S (for SELECT) on the line preceding the KEY.

The displayed information for a list of rule sets is described below:

Key

Specifies the key of the rule set

Compiled by

Specifies the logonid of the user that last compiled the rule set

Last compiled

Specifies the date and time the rule set was last compiled

Length

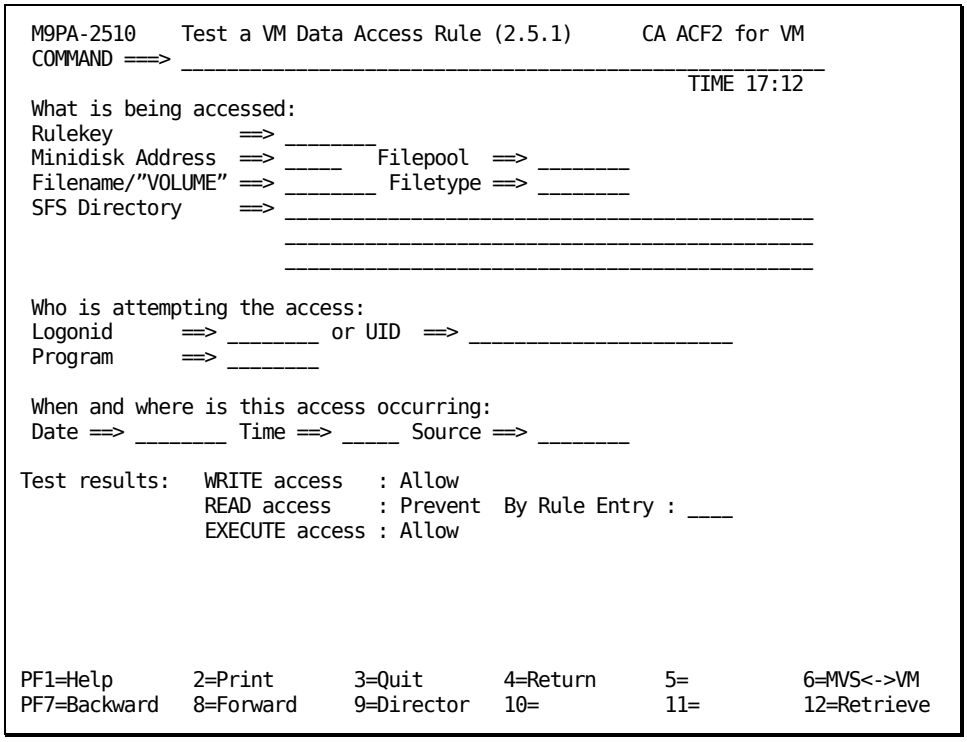
Specifies the length of the rule set

%UTIL

Specifies the percentage of the database used.

Testing a VM Access Rule

This screen lets you test rules and assists you in verifying that the rules you have created do what you want them to do. To use this screen, enter the information you want to test.



What Is Being Accessed

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Minidisk address

Specifies the address of the minidisk that the rule applies to (for example, V0191) where:

R

Specifies real DASD

V

Specifies virtual address. This is the default.

Filepool

Specifies the Shared File System (SFS) filepool identifier this rule applies to.

Filename/"VOLUME"

Specifies the filename of the file this rule applies to. If the rule applies to a minidisk, VOLUME is the filename.

Filetype

Specifies the filetype of the file this rule applies to.

SFS Directory

Specifies the Shared File System (SFS) directory identifier this rule applies to.

Who Is Attempting the Access:

Logonid

Specifies the logonid of the user.

or

UID

Specifies a pseudofield concatenation of selected information from the logonid record that can include information from user-defined fields, such as company code, department, job function, and the logonid field.

Note: Pick only one of these fields. If you fill in both the logonid field and the UID field, CA ACF2 for VM logonid uses the logonid field.

Program

If applicable, enter the program name attempting the simulated access.

When and Where Is this Access Occurring

Date

Specifies the date (in the format mm/dd/yy, dd/mm/yy, or yy/dd/mm, depending upon the DATE field in the OPTS VMO record) that the simulated access is attempted.

Time

Specifies the time of the simulated access.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Test Results

The access permission is displayed following the access type.

Allow

Specifies that the access is allowed.

Log

Specifies that the access is allowed but logged.

Prevent

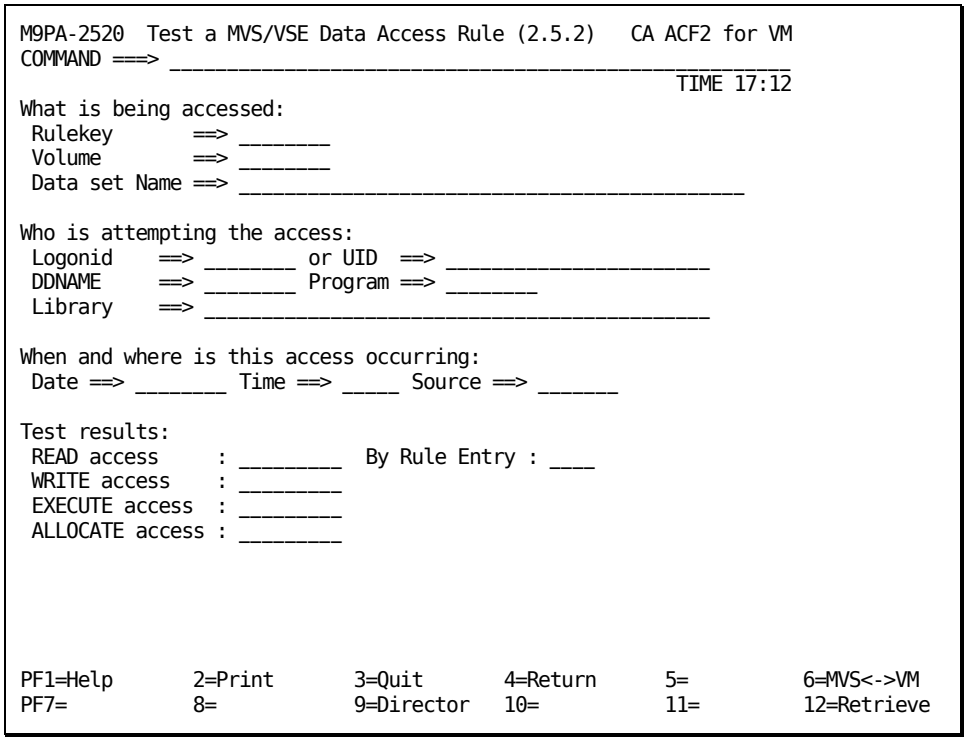
Specifies that the access is denied.

By rule entry:

Specifies that the rule entry line number for the simulated access.

Testing an OS/390 or VSE Data Access Rule

This screen is for testing rules and to assist you in verifying that the rules do what you want them to do. Enter the information you want to test on this screen.



What Is Being Accessed

Rulekey

Specifies the key value of the rule set. The rulekey you specify can be up to eight characters long. You cannot mask this field.

Volume

Specifies the volume of the data set.

Data set name

Specifies the name of the data set.

Who Is Attempting the Access

Logonid

Specifies the logonid of the user.

UID

Specifies a pseudofield concatenation of selected information from the logonid record that can include information from user-defined fields, such as company code, department, job function, and the logonid field.

DDNAME

Specifies the ddname to be accessed.

Program

If applicable, enter the program name attempting the simulated access.

Library

Specifies the library name for the attempted access.

When and Where Is this Access Occurring

Date

Specifies the date (in the format mm/dd/yy, dd/mm/yy, or yy/dd/mm, depending upon the DATE field in the OPTS VMO record) that the access was attempted.

Time

Specifies the time of the simulated access.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Test Results

The access permission is displayed following the access type.

Allow

Specifies the access is allowed.

Log

Specifies the access is allowed but logged.

Prevent

Specifies the access is denied.

By rule entry

Specifies the rule entry line number that applied to the simulated access.

Chapter 12: Maintaining Access Rules with the ACF Command

This chapter provides information for maintaining access rules through the ACF command. The ACF command lets you maintain access rules through line commands instead of the full-screen panels. Whatever method you choose, you must have the necessary privileges in your logonid record to process access rules.

After you complete this chapter, you will know how to use the ACF command to:

- Establish the RULE setting and issue ACF subcommands
- Create an access rule, including control statements and rule entries
- Create access rules for all portions of your VM system
- Compile rule information from a CMS file
- Store an access rule on the Rule database
- Decompile an existing access rule
- Change an access rule
- Delete an access rule
- Test an access rule before you store it.

This section contains the following topics:

Beginning Rule Processing

Issue the SET RULE subcommand to establish the ACF command setting that processes rules.

```
set rule
```

```
RULE
```

The system responds with the following to tell you that you have established the RULE setting for the ACF subcommands:

```
RULE
```

After you have established the RULE setting, you can issue the ACF subcommands. See the “Using the ACF Command” chapter for a list of the ACF subcommands that are valid under the RULE setting.

The following chapters provide information on different types of access rule processing.

Access Rule Components

An access rule set consists of the following:

- Control statements, identified by the $ or % symbols in column 1.
- Comment statements, also considered control statements, identified by the * symbol in column 1.
- Rule entries that must follow the control statements. When entering a rule as input from a terminal or creating or updating a file as compiler input, begin the rule input text in column two. To indicate continuation when writing rules, use a blank space followed by a dash (-) at the end of the line to be continued. If the continuation line begins a new rule keyword, the continuation line must begin in column two or after. If the new line is a continuation of a keyword or its value, the continuation line must begin in column one.

The complete syntax for an access rule set is:

```
  $Key(ruleid)
[ $Mode(Quiet|Log|Warn|Abort)           ]
[ $NOSort                               ]
[ $Owner(ownerid)                       ]
[ $Prefix(prefix)                       ]
[ $Resowner(resource-owner)             ]
[ $Userdata(userdata)                   ]
[ %Change uidmask1,uidmask2,...,uidmaskn ]
[ %Rchange uidmask1,uidmask2,...,uidmaskn]
 *comment

  Vnnnnmask.VOLUME|Vnnnn.fileidmask|Rnnnnmask|DSNmask
[ Volume(volmask)                       ]
[ UId(uidmask)                          ]
[ SOurce(source)                        ]
[ SHift(shiftname)                      ]
[ ACtive(date)                          ]
[ UNtil(date)|For(days)                 ]
[ PGm(pgmmask)                          ]
[ Read(Allow|Log|Prevent)               ]
[ Write(Allow|Log|Prevent)              ]
[ Exec(Allow|Log|Prevent)               ]
[ Allocate(Allow|Log|Prevent)           ]
[ FPool(filepool)                       ]
[ DIR(filedir)                          ]
[ Library(libmask)                      ]
[ DDname(ddnmask)                       ]
[ DAta(text)                            ]
  Nextkey(nextkey)
```

- The lines beginning with $ or % show the syntax for the different types of control statements
- All $ control statements must come before any rule entries
- The entries beginning in column two show the syntax for a single rule entry.

CA ACF2 for VM changes any CMS file containing rule text exceeding 80 characters into continuation lines. This makes it easier to edit and print the file. The compiler accepts input files that already exist from previous releases that have record lengths up to 256 characters.

$KEY(ruleid)

Specifies the logonid of the user this rule is written for, usually the ID of a VM user who has one or more MDISKs specified in the VM Directory under his user entry. In most cases, this is also the user's CA ACF2 for VM logonid. The rule ID specified can be up to eight characters long for access rules. You cannot mask this field.

$MODE(QUIET|LOG|WARN|ABORT)

Aids the transition to phase in data protection on a rule set basis. You must meet two conditions before $MODE can be in effect: you must specify the MODE(RULE,no-rule,no-$mode) option in the OPTS VMO record, and an ABORT condition occurs when CA ACF2 for VM validates access to data the rule set protects (optional).

$OWNER(ownerid)

Specifies the owner of a rule set. It is for information only. No CA ACF2 for VM processing is based on this parameter. It does not grant the owner any special privileges relative to the rule set. You can enter the logonid, UID, name of the owner, or other values (up to 24 characters) for local tracking purposes. CA ACF2 for VM displays this value when the rule set is decompiled (optional).

$PREFIX(prefix)

Specifies a value as the minidisk ID for CMS files. You can enter up to 24 characters. You can specify multiple levels (for example, TLCAMS.V0191). If you specify $PREFIX(), the prefix is set equal to the $KEY entry and the $PREFIX control card is not generated when the rule is decompiled. CA ACF2 for VM issues a warning message indicating that the $PREFIX specified is null and is ignored (optional).

$RESOWNER(resource-owner)

Specifies the resource owner (RESOWNER) of the data set. You can specify up to eight characters for the logonid acting as the RESOWNER of the dataset. CA ACF2 for VM does not use $RESOWNER. For more information, see the *Administrator Guid*e.

$NOSORT

Specifies that the CA ACF2 for VM compiler does not sort the rule set. If you do not specify this statement, CA ACF2 for VM sorts the rule set from most specific to most general. **We recommend that you do not use this option.** (You must also specify the VSM RULEOPTS $NOSORT parameter.)

$USERDATA(text)

Contains text up to 64 characters. USERDATA information is stored with the rule set (optional).

%CHANGE uidmask1,uidmask2,...,uidmaskn

Specifies who (besides the data owner or security administrator) can replace a particular set of rules. A security administrator can compile a rule set with only $KEY and %CHANGE control statements, establishing a base rule set to distribute rule writing permissions (optional).

%RCHANGE uidmask1,uidmask2,...,uidmaskn

Specifies who has **restricted** CHANGE authority over the rule set. The designated users can change individual rule entries, but not control statements. They cannot delegate change authority or delete the rule set. If the same user matches entries in %CHANGE and %RCHANGE, %CHANGE takes effect (optional).

Comment statements

Specified by an asterisk (*) in column one. Unlike the $USERDATA field, comment statements are lost on a compile and decompile sequence. They can appear anywhere in the input (optional).

dsn

Indicates the format of the dsn keyword depends on the type of data that is shared.

Minidisks

Specify the virtual device address, preceded by a V (indicating virtual device), followed by VOLUME (such as V0190.VOLUME).

CMS files

Indicate the virtual device, preceded by a V, followed by the filename and filetype (such as V0191.WORK.DATA).

SFS files

Indicate the filename and filetype.

OS/390 data sets

Specify a data set name using OS/390 data set naming conventions (such as SYS1.PARMLIB). But, the high level index is the $KEY(rule ID), not part of the dsn parameter.

VSE data sets

Specify a data set name using VSE naming conventions (such as PAYROLL.MASTER.DATA). CA ACF2 for VM appends the $KEY(rule ID) field to the beginning of the data set name field to form a full 44-byte DOS filename.

Attachable DASD devices

R, followed by the real address of the device (such as R190) for the disk device.

Tape volumes

Indicate the volume serial identification (volser) or VOLUME, depending on the TAPEVOLS VMO record.

The compiler prefixes the dsn value with the $KEY(rule ID) or the character string specified in the $PREFIX control operand. You can place single quotes around a dsn to specify an entire dsn, including the high-level qualifier, however, you would normally do this only for NEXTKEY processing.

VOL(volmask)

Specifies the volume serial number of the DASD volume that must be mounted on the device to match this rule. If omitted, any volume is considered matched. This parameter is valid only for attachable DASD device access rules. You can mask this parameter.

UID(uidmask)

Specifies who the rule should apply to. If omitted, the entry applies to all users (optional).

SOURCE(source)

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user is logged onto the specific terminal. If no source is specified, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

SHIFT(shiftname)

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines days, dates, and times when access is allowed. If you do not specify this parameter, any access the rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times (optional).

ACtive(date)

Specifies a Gregorian date in the form mm/dd/uu, yy/mm/dd, or dd/mm/yy (depending on a site option) that is the first date when this rule is considered valid. (See the DATE field of the OPTS VMO record for the site option information.) Years specified as 70-99 assume a date in the 20th century (1970-1999). Years specified as 00-69 assume a date in the 21st century (2000-2069). This parameter is valid only when the VMO RULEOPTS RULELONG parameter is set.

UNTIL(date)

Indicates a Gregorian date (specified as mm/dd/yy, dd/mm/yy, or yy/mm/dd) that is the last date when this rule is valid (optional).

FOR(days)

Specifies the number of days that this rule is considered valid, starting from the day the access rule set was compiled. The minimum number you can specify is zero (today). The maximum number is 365 (optional).

READ(A|L|P)

Indicates a character (A, L, or P) specifying the read access permission to be applied if there is a successful match of the dsn, vol, UID, and source parameters (optional).

A

Access is allowed

L

Access is allowed but logged

P

Access is prevented (the default).

WRITE(A|L|P)

The same as READ, except that it applies to write access (opening the file for output) (optional).

EXEC(A|L|P)

The same as READ, except that it applies to execute-only access. This value is always set to its specified value or the value specified for the READ parameter-whichever grants more permissive access. Execute authority is already allowed for any modules or macros that reside on the S-disk. Attempts to EXEC(P) or EXEC(L) execs on the S-disk are logged (optional).

This keyword is not valid for SFS rule entries.

ALLOCATE(A|L|P)

The same as READ except that this parameter applies to data set allocation. For CMS access validation requests, this parameter applies only to VSAM files.

This keyword is not valid for SFS rule entries.

FPOOL(filepoolmask)

Specifies the SFS filepool portion of a SFS file identifier that you want to protect. The presence of the FPOOL keyword means that this is a SFS file rule entry and it will apply only to SFS access validations. If you omit FPOOL on a SFS file rule entry, FPOOL defaults to FPOOL(-) and matches all SFS filepool names.

DIR(filedirmask)

Specifies the SFS directory portion of a SFS file identifier that you want to protect. The presence of the DIR keyword means that this is a SFS file rule entry and it will apply only to SFS access validations. If you omit DIR on a SFS file rule entry, DIR defaults to a null value and matches SFS file identifiers without any directory levels (a SFS root directory). To match any directory, specify DIR(-).

PGM(pgmmask)

Specifies the name of a program (DDR or FORMAT) you want to protect (optional).

This keyword is not valid for SFS rule entries.

DATA(text)

Specifies a 64-character string retained with the rule set and formatted when the rule set is decompiled. Your site may have standards concerning the format of this string. Standard CA ACF2 for VM does not use these values, but they can be meaningful in your local implementation of CA ACF2 for VM through special program exit checking (optional).

NEXTKEY(nextkey)

Specifies the rule ID of the next (or alternate) rule set to check for this access. If CA ACF2 for VM denies access to this file or data set based on the rule set environment and access permissions in the original rule, CA ACF2 for VM proceeds to the rule specified in the NEXTKEY operand for further checking (optional).

LIBRARY(libmask)

This is an OS/390 keyword. If you specify this keyword in a rule entry, CA ACF2 for VM rule validation skips it. The ACF command access rule TEST subcommand supports this keyword.

DDNAME(ddnmask)

This is an OS/390 parameter. If you specify this parameter in a rule entry, CA ACF2 for VM rule validation skips it. The ACF command access rule TEST subcommand supports this parameter.

Control statements and rule entry parameters are explained in the following sections.

Access Rule Control Statements

Access rules for each minidisk are grouped together and built into a rule object record. The input to the rule compiler consists of the access rule entries and the control statements listed below. You can use a blank space and a dash (-) at the end of a line to continue all input on multiple statements.

The $KEY control statement is the only required control statement. You can specify any or all of the additional control statements, but only one of each type is allowed per rule set. All $ control statements must precede % control statements. Control statements begin in column one.

You can specify multiple $ control statements on the same line with a single $ in column one. For example, you can use either of the following formats:

Format 1:

```
$KEY(SYS9)
$PREFIX(SYS*)
$USERDATA(comments)
```

Format 2:

```
$KEY(SYS9) PREFIX(SYS*) USERDATA(comments)
```

$KEY When TLCAMS is compiling a rule set to allow access to the file MYFILE COBOL on MDISK 0191, the $KEY control statement contains $KEY(TLCAMS). When you write rules for your own files, the $KEY control statement usually contains your logonid because it is often specified as the PREFIX for your own files or minidisks.

For DASD devices that can be dedicated to a particular virtual machine using the CP ATTACH command or the DEDICATE VM directory control statement, the $KEY is always SYSTEM for DASD.

$MODE The no-rule parameter indicates the action CA ACF2 for VM takes if no access rule is found. The no$mode parameter indicates the action
CA ACF2 for VM takes if no $MODE control statement is found in the applicable access rule set. Valid values for no-rule and no-$mode are QUIET, LOG, WARN, and ABORT.

For example, in the following rule set only user TLCPJM could link to TLCAMS's 0191 minidisk with read and write access. The $MODE statement with a LOG mode lets all other users access (with loggings). Let's look at the example below:

```
$KEY(TLCAMS)
$MODE(LOG)
 V0191.VOLUME UID(TLCPJM) R(A) W(A)
```

%CHANGE The format of the %CHANGE control statement is %CHANGE (uidmask,...,uidmask). Each UID mask is a full UID or UID pattern. You can deactivate the %CHANGE function through the CHANGE|NOCHANGE operand of the RULEOPTS VMO record. %CHANGE lets designated users change the rule set to delegate %CHANGE authority to other users. They can change or delete any part of the rule set. There is no limit to the number of change commands that you can enter.

%RCHANGE A security administrator can compile a rule set with only $KEY and %RCHANGE control statements, establishing a base rule set to distribute rule writing permissions. %RCHANGE lets designated users change only rule entries in the rule set. They cannot delegate rule writing authority. There is no limit to the number of change commands that you can enter.

Access Rule Entries

Individual access rule entries follow the control statements in a rule set and specify the environment and access permissions when access to specific data takes place. Like control statements, you can use a blank space and a dash (-) at the end of the line to continue a rule entry from one statement to the next. Rule entries must start in column two so that CA ACF2 for VM does not treat an entry beginning with an asterisk (*), dollar sign ($), or percent sign (%) as a comment line or control statement.

Different Types of Access Rules

When writing access rules, keep the following points in mind:

- You can specify access rules in fixed or variable format. Fixed format records with a length of 80 have the last eight characters reserved for a sequence field.
- You can use a blank space and a dash (-) at the end of the line to continue all control statements and rule entries on multiple lines.
- The $KEY control statement is the only required control statement. You can specify any or all of the additional control statements, but only one of each type is allowed per rule set. All control statements that begin with a dollar sign ($) must precede all control statements that begin with a percent sign (%).
- Control statements begin in column one. You can specify multiple $ control statements on the same line with a single $ in column one. Rule set entries must begin in column two so an entry beginning with an asterisk is not treated as a comment line.

This section describes how to write access rules to control access to each of the following portions of your VM system:

- Minidisk links
- CMS files
- SFS files
- OS/390 data sets
- VSE data sets
- VSAM allocation protection
- DASD ATTACH commands
- Tape volumes
- DASD Dump Restore (DDR) service program
- VM Batch environment and subsystem.

We provide sample access rules for each of these items. We also explain how to use NEXTKEY to merge and split access rule sets.

CMS File, OS/390 and VSE Data Set Level Integrity

The NCSC C2 rating evaluates objects at the device or minidisk level and not at the file or data set level. There is limited inherent security in the CMS environment. CMS file, OS/390 and VSE data set integrity cannot be guaranteed in a C2-rated system. However, SFS file security offers the same level of integrity as that we provide for minidisk links.

IBM has not yet issued an integrity statement for CMS because the CMS environment lets the general user program execute in supervisor state, enter storage protect key zero, execute privileged instructions, issue input and output commands, and process interrupts independently of the CMS nucleus. A systems programmer could take advantage of these CMS characteristics and modify the CMS nucleus to compromise CMS, OS/390, and VSE data set level security. This lack of integrity in the CMS environment limits any security system from providing absolute file level protection for CMS files. However, we provide absolute security for SFS files.

Despite being under the constraints of the CMS environment, CA ACF2 for VM stills provide a high level of CMS file security (although not C2 file level security). An explanation of CMS file, SFS file, and OS/390 and VSE data set access follows.

Minidisk and CMS File Access Rules

CA ACF2 for VM protects VM minidisks, CMS files, and SFS files. Access rules for minidisks and files determine who can access them and what access privileges (read, write, or execute) they are allowed.

For each VM user, there is a VM directory entry and a CA ACF2 for VM logonid record. In the VM directory entry, most users are assigned minidisks that they own. There is one MDISK entry for each minidisk the user owns. Typically, users access their own minidisks and files frequently. To avoid unnecessary validations, CA ACF2 for VM always lets users access their own minidisks and files. The PREFIX field in the logonid record defines ownership.

Users can issue LINK commands to access minidisks they do not own. The LINK command names the user that owns the minidisk and its virtual address. For example, TLCPJM might issue the following command:

```
LINK TLCAMS 0191 0291 RR readpswd
ACC 0291 B
```

The LINK and the ACCESS command let TLCPJM read (RR) TLCAMS 0191 minidisk at address 0291. Users can also issue the ACCESS command to access SFS directories they do not own. The ACCESS command names the filepool, the directory owner ID, and the directory name. For example:

```
ACCESS APPLDEV
TLCAMS.PROJDATA B
```

You can issue these LINK and access commands

- Through a user's VM directory statement, causing an automatic LINK when the user logs onto CA ACF2 for VM. CA ACF2 for VM validates all attempted LINKs.
- Through a user's PROFILE EXEC. Each user can create his own profile that is automatically executed when the user logs onto VM. CA ACF2 for VM validates these LINKs.
- Through EXEC procedures, often used to stack commands for easy execution and often contain LINK commands. CA ACF2 for VM validates these LINKs.
- Directly from the terminal. CA ACF2 for VM validates these LINKs.

After a user LINKs to another user's minidisk, CA ACF2 for VM validates access to the files on the minidisk if CMS security is turned on. This provides two levels of validation, LINK and file. For example, to let a user LINK to one of your minidisks, you must write a rule using your logonid as the $KEY in the rule set. Then you can write additional rule entries to define exactly which CMS files the user can access.

The only exception is that access to an S-mode minidisk automatically allows execution of any module on that minidisk.

If SFS security is installed, CA ACF2 for VM always validates access to individual files in a directory.

Access Rules for Minidisks

In a minidisk access rule, CA ACF2 for VM constructs a pseudo dsn. The dsn parameter includes the virtual address of the minidisk (preceded by V), followed by a period, followed by the constant, VOLUME. The VM user ID of the MDISK owner is specified in the $KEY or $PREFIX control statements.

The syntax of a minidisk access rule is:

```
                { UID(uid)              }
                { UNTIL(date)|FOR(days) }
 Vccuu.VOLUME   { READ(A|L|P)           }
                { WRITE(A|L|P)          }
```

The parameters for this access rule are defined below.

ccuu

Specifies the virtual address of the minidisk (such as 0190 or 0191).

UID(uid)

Indicates a pattern specifying the users where this rule applies.

UNTIL(date)

Specifies the date that the rule becomes invalid.

FOR(days)

Defines how many days that rule is in effect.

READ A|L|P, WRITE A|L|P

Defines precisely when and how this user can access this minidisk.

A sample rule letting all application programmers link to TLCAMS's 0191 minidisk for one week follows:

```
$KEY(TLCAMS)
 V0191.VOLUME UID(APP-) FOR(7) READ(A)
```

The access specified is READ. By default, CA ACF2 for VM automatically prevents WRITE access. The FOR operand specifies that this rule remains in effect for seven days, beginning with the day that the rule is compiled. This minidisk rule lets users matching the UID mask APP- link to the virtual address 0191 minidisk minidisk TLCAMS owns. These users do not have access to any CMS files on that minidisk. You must write another access rule to allow access to a CMS file or group of files residing on the 0191 minidisk.

In access rules for CMS files, V must precede the virtual address of the minidisk where the file is located. Also, you must specify the filename and filetype for the CMS file access rules.

Access Rules for CMS Files

In CMS file access rules, CA ACF2 for VM constructs a pseudo dsn. The dsn includes the virtual address of the minidisk (preceded by a V), followed by a period, followed by the filetype. Specify the VM user ID of the minidisk owner in the $KEY or $PREFIX control statements. The syntax of a CMS file access rule is:

```
                         {UID(uid)               }
                         {UNTIL(date)|FOR(days)  }
 Vcccu.filename.filetype {READ(A|L|P)            }
                         {WRITE(A|L|P)           }
                         {EXEC(A|L|P)            }
```

The parameters for this access rule are defined below:

Vcccu

Specifies the virtual address of the minidisk

Filename

Specifies the CMS filename

Filetype

Specifies the CMS filetype

UID(uid)

Indicates a pattern specifying the users this rule applies to

UNTIL(date)

Specifies the date that the rule becomes invalid

FOR(days)

Defines how many days the rule is in effect

READ A|L|P, WRITE A|L|P, EXEC A|L|P

Defines precisely when and how this user can access this minidisk.

For example, TLCAMS has allowed links to her 0191 minidisk through the minidisk rule shown in the previous example. To allow access to a particular COBOL file on that minidisk, she must write an additional rule. The following example illustrates this:

```
$KEY(TLCAMS)
 V0191.VOLUME UID(APP-) FOR(7) READ(A)
 V0191.MYFILE.COBOL UID(APP-) FOR(7) READ(A)
```

Now all users whose UIDs match the mask APP- have READ and EXEC access to the CMS file MYFILE.COBOL TLCAMS owns residing on minidisk 0191.

What if TLCAMS wants to let one particular applications programmer (TLCPJM) write (or update) the COBOL file? She must write an additional minidisk rule entry to allow this more specific access. The following CMS file rule is also needed:

```
$KEY(TLCAMS)
 V0191.VOLUME UID(APP-) FOR(7) READ(A)
 V0191.MYFILE.COBOL UID(APP-) FOR(7) READ(A)
 V0191.VOLUME UID(APPTLCPJM) FOR(7) READ(A) WRITE(A)
 V0191.MYFILE.COBOL UID(APPTLCPJM) FOR(7) READ(A) WRITE(A)
```

Access Rules to Rename CMS Files

To rename a CMS file, the user must have read and write access to both the current CMS filename and the new CMS filename. If these conditions are not met, the file is not renamed and the attempted action is logged. CA ACF2 for VM issues appropriate messages.

Access Rules for Execute-Only CMS Files

Execute is one of the access parameters of a CA ACF2 for VM access rule. As with READ or WRITE access, you can define execute access as ALLOW (A), LOG (L), or PREVENT (P). Execute-only access means that specified users can only execute the file. They cannot read or write to the file. Execute-only access prevents users from reading, copying, or issuing traces for sensitive files, programs, modules, and execs.

You can define execute-only access for a CMS file when a rule prevents or logs read access attempts and allows or logs execute accesses. The following rule entries define execute-only environments:

```
V0191.MYFILE1:module.UID(*) READ(P) EXEC(A)
V0191.MYFILE2:module.UID(*) READ(P) EXEC(L)
V0191.MYFILE3:module.UID(*) READ(L) EXEC(A)
V0191.MYFILE4:module.UID(*) READ(L) EXEC(L)
```

You must meet two conditions before you can create execute loggings: CMS files must have the EXEC(L) or EXEC(P) access environments in the rule, and CMS files with EXEC(P) or EXEC(L) access environments cannot reside on the CMS 190 S-disk. CA ACF2 for VM does not perform execute access validation for files with READ(A) access. The second and fourth rules in the previously displayed example create SMF loggings.

To improve performance, no execute-only loggings are created for any modules on the S-disk.

Writing Execute-Only Rules

This example is similar to the one in the previous section. TLCAMS has allowed links to her 0191 minidisk through a minidisk rule:

```
$KEY(TLCAMS)
 V0191.VOLUME UID(APP-) READ(A)
```

To log READ access and allow execute-only access to a particular module on that minidisk, she must write an additional rule entry:

```
$KEY(TLCAMS)
 V0191.VOLUME UID(APP-) READ(A)
 V0191.MYFILE:module.UID(APP-) READ(L) EXEC(A)
```

Now, all users whose UIDs match the APP- mask are logged for READ access and allowed EXEC access to the MYFILE:module.file TLCAMS owns that resides on the 0191 minidisk.

What if TLCAMS wants to let one particular applications programmer (TLCPJM) execute the MODULE but prevent read access? She must write an additional rule entry to prevent this more specific access.

```
$KEY(TLCAMS)
 V0191.VOLUME UID(APP-) READ(A)
 V0191.MYFILE:module.UID(APPTLCPJM) READ(P) EXEC(A)
 V0191.MYFILE:module.UID(APP-) READ(L) EXEC(A)
```

This new rule entry lets APPTLCPJM execute TLCAMS's CMS module MYFILE:module.that resides on the 0191 minidisk, but prevents APPTLCPJM from reading the module.

Performing Execute Validations

CA ACF2 for VM performs execute-only validations in all the following circumstances, providing that the access rule specifies READ(P) or READ(L). (CA ACF2 for VM does not create loggings for modules that reside on the
S-disk.):

- Initiating a module
- Initiating an exec
- Initiating an XEDIT macro
- Loading a nucleus extension with the CMS NUCXLOAD command
- Loading an exec or XEDIT macro with the CMS EXECLOAD command.

As long as the execute-only file remains in storage, all storage displays initiated through the CP DISPLAY, DUMP, and VMDUMP commands are suppressed unless the user has the SECURITY or DUMPAUTH privilege. Additionally, all traces invoked through the PER and TRACE commands and all EXEC traces are suppressed unless the user has the SECURITY or DUMPAUTH privilege.
CA ACF2 for VM does not perform execute-only validations on TEXT files.

There are a small number of CMS module files that call DMSRLD at execution time to read a second copy of the module into virtual storage. When those modules are initiated, CA ACF2 for VM does an execute validation. After these modules begin executing, the call to DMSRLD causes CA ACF2 for VM to perform a read validation. Therefore, you may need to allow users to have read access to those particular modules so these modules can execute properly. The WAKEUP, FILESTAK, and FILESTCK modules are known to make this type of call to DMSRLD.

Clearing Storage

When the execute-only file is dropped from storage, CA ACF2 for VM clears the storage to binary zeros when:

- Terminating an execute-only module
- Terminating an execute-only exec that was not explicitly loaded through EXECLOAD
- Terminating an execute-only XEDIT macro that was not explicitly loaded through EXECLOAD
- Dropping a previously loaded execute-only exec or XEDIT macro with EXECDROP
- Dropping a previously loaded execute-only nucleus extension with NUCXDROP.

All outstanding execute-only files must be dropped from storage before a user can reinitiate a display or trace. For information on using CP commands for execute-only files, see the Syste*ms Programmer Guid*e.

Access Rules for Shared File System Files

SFS allows CMS users to use CMS files that do not reside on user minidisks, but in hierarchical directories that are very similar to the directories available under MS-DOS, to extend the CMS file system.

SFS File Identifiers

Each SFS file server manages a collection of SFS files called a filepool. There is only one filepool for each SFS server machine. Every user must know the name of the filepool where any files reside that they might need access to. The filepool name is an integral part of the file identifier (we will go into more detail about this later).

Every SFS file in a filepool resides in a root directory of the same name as the owning user ID so it is associated with its owner. Like MS-DOS, there can also be hierarchical subdirectories under each root directory that also contain SFS files. SFS supports up to eight levels of subdirectories. Each subdirectory name can be up to 16 characters long.

Let's take a look at how all these components fit together to form the file identifier for an SFS file. As an example, consider a CMS file called MY DATA that user ID TLCAMS owns in the filepool APPLDATA. This file resides in TLCAMS' root directory. The complete file identifier for this file would be:

```
APPLDATA:TLCAMS.MY.DATA
```

It is very important that a colon (:) always follow the filepool name and the owner ID always follow the colon.

Now, let's consider a more complex example where the file resides three levels deep under subdirectories. The first level is a directory called UNITEDSTATES. The second level is called ILLINOIS. The third level is called CHICAGO. The full file identifier is:

```
APPLDATA:TLCAMS.UNITEDSTATES.ILLINOIS.CHICAGO.MY.DATA
```

It is important to note that the last two elements of the file identifier are always the filename and filetype of the CMS file.

A few CMS commands let the user refer to an SFS file using the full file identifier, however, this is usually awkward. Most users instead prefer to access the directory where the file resides at a CMS filemode exactly as they would for a minidisk.

Let's assume that you want to access the directory in our most recent example as the E disk. You would enter the following:

```
ACCESS APPLDATA:TLCAMS.UNITEDSTATES.ILLINOIS.CHICAGO E
```

After completing this step, all the files in that subdirectory are accessible as if they resided on a minidisk at mode E. In this example, you would refer to MY DATA E.

Because you will normally want to access files that reside entirely in a single filepool, there is a way to declare a filepool so that you do not have to enter the filepool name again. An example of this syntax follows:

```
SET FILEPOOL APPLDATA:
```

After you enter the SET FILEPOOL command, use the following command syntax to access the directory:

```
ACCESS TLCAMS.UNITEDSTATES.ILLINOIS.CHICAGO E
```

You can also refer to your own root directory with a period (.) instead of entering its name. For example, to access this directory, TLCAMS enters:

```
ACCESS .UNITEDSTATES.ILLINOIS.CHICAGO E
```

Writing Access Rules for SFS Directories

To access a file in an SFS directory, you need the proper authorization to access the relevant directory. You need to write a directory access rule as part of your rule set. The SFS directory access rule is required to access files and to issue the CMS ACCESS command for the directory.

For example, if TLCPJM tries to access TLCAMS' directory called UNITEDSTATES.ILLINOIS.CHICAGO that resides in the APPLDATA filepool, he must issue the following command:

```
ACCESS APPLDATA:TLCAMS.UNITEDSTATES.ILLINOIS.CHICAGO E
```

Because TLCPJM does not own this directory, the action defaults to access this directory in read only mode. To allow access, the following read access rule is required:

```
$KEY(TLCAMS)
 / FILEPOOL(APPLDATA) DIR(UNITEDSTATES.ILLINOIS.CHICAGO)UID(TLCPJM) READ(A)
```

Even though some CMS commands (such as RENAME) can directly address SFS files without use the CMS ACCESS command to first access it, you still need the directory access rule because implied access is in effect for the duration of the file access. You can use standard CA ACF2 for VM masking characters for the filepool (abbreviated FP) and the directory values.

User ID TLCPJM can explicitly request access to this directory in write mode as follows:

```
ACCESS APPLDATA:TLCAMS.UNITEDSTATES.ILLINOIS.CHICAGO E (FORCERW
```

There are two types of control directories, the File Control directory and the Directory Control (DIRCONTROL) directory. For the File Control directory, the most common type, the above access command would work even if the user only has read authority for the directory.

However, for the DIRCONTROL directory, the user needs both read and write access permission in the rule to authorize access. For example:

```
$KEY(TLCAMS)
 / FILEPOOL(APPLDATA) DIR(UNITEDSTATES.ILLINOIS.CHICAGO) UID(TLCPJM) -
 READ(A) WRITE(A)
```

Because SFS directories accessed in write mode can also be read, you must always specify READ(A) when you specify WRITE(A) for SFS access rules.

Writing Access Rules for SFS Files

After you write a rule allowing a user to access a directory, you must also write a rule allowing him to open the files in that directory. Unlike native SFS security where DIRCONTROL directories provide no individual file level security, CA ACF2 for VM SFS security protects individual files in both File Control and DIRCONTROL directories.

Suppose TLCPJM needs to read TLCAMS' file called MY DATA in the UNITEDSTATES.ILLINOIS.CHICAGO directory. That directory resides in the APPLDATA filepool, so the following rule is required:

```
$KEY(TLCAMS)
 / FILEPOOL(APPLDATA) DIR(UNITEDSTATES.ILLINOIS.CHICAGO) UID(TLCPJM) READ(A)
 MY.DATA FILEPOOL(APPLDATA) DIR(UNITEDSTATES.ILLINOIS.CHICAGO) UID(TLCPJM)
 READ(A)
```

In the above example, we have included the directory access rule and the file access rule because both are required. If you wanted to write a single rule that would authorize TLCPJM to read all the files in this directory and also authorize read access for the directory itself, you could write a rule like this one:

```
$KEY(TLCAMS)
 - FILEPOOL(APPLDATA) DIR(UNITEDSTATES.ILLINOIS.CHICAGO) UID(TLCPJM) READ(A)
```

Writing Access Rules for Remote Users

You can configure SFS filepools to let users on remote nodes transparently read and write files on the local node. At this time, we can only determine if an access is local or remote. We cannot determine the precise APPC node of origin for a remote user.

To provide some control for remote and local user file access, you should use CA ACF2 for VM source validation in access rules. Remote SFS users will always have a terminal ID named SFSREMOT. Local SFS users will have their true terminal ID on the local system

The following rule set provides an example that would give user ID TLCRAM write access to the PAYROLL DATA file in TLCAMS's root directory when TLCRAM is accessing the data locally, but only permits read access when TLCRAM tries to access the data from a remote system.

```
$KEY(TLCAMS)
 / FPOOL(APPLDATA) UID(TLCRAM) SOURCE(SFSREMOT) READ(A)
 / FPOOL(APPLDATA) UID(TLCRAM) READ(A) WRITE(A)
 PAYROLL.DATA FPOOL(APPLDATA) UID(TLCRAM) SOURCE(SFSREMOT) READ(A)
 PAYROLL.DATA FPOOL(APPLDATA) UID(TLCRAM) READ(A) WRITE(A)
```

Command Limiting SFS Commands

Native Shared File System (SFS) administration provides for one or more administrators who have unrestricted access to everything in a filepool. This forces you to create separate filepools whenever you want to grant an SFS administrator authority over a subset of users. While you may want multiple filepools for performance reasons, they may not be the answer to providing greater ease of administrative authority. Even creating multiple filepools does not always solve the problem of splitting authority in a detailed and flexible manner.

CA ACF2 for VM provides the option of validating SFS-related commands SFS administrators issue through the CA ACF2 for VM command limiting feature.

The SFS command syntax CA ACF2 for VM validates is not free form as CP commands. Instead, SFS reconstructs each command in a fixed format before it passes it to CA ACF2 for VM .

For information on command limiting SFS commands, see the *Command and* Diagnose *Limiting Guide.*

Loading an SFS Directory into a VM Dataspace

VM lets a particular type of directory, called a DIRCONTROL directory, be loaded into a dataspace. This loading makes the files in this directory memory resident and any user with access to the directory can directly read it.

After a user accesses a directory that is loaded into a dataspace, the Shared File System (SFS) server and CA ACF2 for VM do not intervene for read access to the files in the directory. Files opened for input bypass all security. However, CA ACF2 for VM does continue to perform write validations for any files opened for output.

Because of this restriction, it is very important for you to carefully select directories for loading into a dataspace that have no CA ACF2 for VM rules designed to selectively limit read access to individual files in the directory.

To facilitate centralized control over both file access rules and over the loading of directories into dataspaces, you should limit DATASPACE command use. See the *Command and* Diagnose *Limiting Guid*e for more information on command limiting.

Access Rules for Minidisk Format Protection

CA ACF2 for VM also validates accesses made to a minidisk through the CMS FORMAT command and the ACCESS command with the ERASE option. Normally, only the owner of a minidisk can use the FORMAT or ACCESS with ERASE commands. CA ACF2 for VM can enforce this standard through access rules.

When a user issues a FORMAT command or an ACCESS command with the ERASE option, CA ACF2 for VM allows access if the user is the owner of the minidisk or the user has authorized write access to the minidisk.

Also, you can write access rules that allow write access to a minidisk while also preventing the use of FORMAT and ACCESS with ERASE. Use the PGM(FORMAT) parameter in an explicit prevent rule.

```
$KEY(PAYROLL)
 V0191.- UID(ABCPAYTLCAMS) READ(A) WRITE(A) EXECUTE(A)
 V0191.- UID(ABCPAY) WRITE(P) PGM(FORMAT)
 V0191.- UID(ABCPAY) READ(A) WRITE(A) EXECUTE(A)
```

The first rule entry lets user ABCPAYTLCAMS read, write, and execute the PAYROLL minidisk. This user can also use the FORMAT command. The second rule entry prevents other users with UIDs beginning with PAY from executing the FORMAT command on that minidisk. The third rule entry lets those users read, write, and execute the minidisk.

After an access rule is compiled (and sorted), the format rule must appear first, before any rule that allows the user write access to the minidisk.

OS/390 Data Set Access Rules

We can use Ann Smith's Applications Department to look at some simplified examples of access rules for OS/390 data sets. Assume that the payroll information, including salary rates and other confidential information, is on the data set PAYROLL.MASTER.DATA. Ann wants to let herself and only her lead programmer (TLCPJM) access this data set. In addition, the programmer is only allowed access to read the data for thirty days. Ann has already written a rule set to allow link access to her OS/390 minidisk. Ann's rule set to allow access to PAYROLL.MASTER.DATA looks like this:

```
$KEY(PAYROLL)
 MASTER.DATA UID(APPMGRTLCAMS) READ(A)
 MASTER.DATA UID(APPPRGTLCPJM) READ(A) FOR(30)
```

The $KEY indicates that the high level index of the data set is PAYROLL. The first field in each rule entry is the remainder of the data set name (MASTER.DATA). The UID fields specify the users being given read authority. The FOR parameter indicates the programmer has access only for 30 days. Ann keeps a current project list online for her department in the data set TLCAMS.CURRENT.PROJECTS. She wants to let everyone in the department read this data set. She has authorized her lead programmer to update the rule set that governs her own data sets, which she indexes using her logonid, TLCAMS. Below is a sample of how Ann's rule set appears:

```
$KEY(TLCAMS)
*MY LEAD PROGRAMMER CAN UPDATE MY RULE SET
%CHANGE APPPRGTLCPJM
*ALL APPLICATION PERSONNEL CAN READ THE PROJECT LIST
 CURRENT.PROJECTS UID(APP) READ(A)
```

Using the VOL Parameter

The VOL parameter indicates the specific volume serial numbers a user can access to provide additional flexibility and control when creating an access rule. For example, an existing rule set lets a systems programmer, ABCSYSTLCABC, link access to OS/390 volumes PROD01 and TEST01. If both volumes contain a data set named SYS1.PARMLIB, you can use the following rule set to grant ABCSYSTLCABC read access to the copy of SYS1.PARMLIB on volume TEST01:

```
$KEY(SYS1)
 PARMLIB UID(ABCSYSTLCABC) READ(A) VOL(TEST01)
```

You can also mask the VOL parameter.

```
$KEY(SYS1)
 PARMLIB UID(ABCSYSTLCABC) READ(A) VOL(TEST**)
```

In this example, ABCSYSTLCABC can read data set SYS1.PARMLIB on any DASD device with a volume serial number beginning with TEST.

You can also use the VOL parameter for OS/390 file protection.

VSE Data Set Access Rules

You can also write access rules to protect VSE files that can be accessed through CMS. The syntax of the access rule sets is the same as for OS/390 data sets.

```
$KEY(PAYROLL)
 MASTER.DATA UID(APPMGRTLCAMS) READ(A)
 MASTER.DATA UID(APPPRGTLCAMS) READ(A) UNTIL(01/01/99)
```

This rule set specifies read access for the VSE file PAYROLL.MASTER.DATA. The first rule entry allows read access to user APPMGRTLCAMS. The second grants read access to user APPPRGTLCAMS until (and including) January 1, 1997.

Access rules allow VSE filenames of up to 44 total characters and up to eight characters per index level. If you want to protect VSE files with nonstandard names, you must insert your own code to convert those names to standard ones for CA ACF2 for VM validation. You can implement this code through the
CA ACF2 for VM data set prevalidation exit.

VSAM File Access Rules

You can also write access rules to protect VSAM files that you can access through CMS. The syntax of the access rule sets is the same as for OS/390 data sets.

```
$KEY(PAYROLL)
 MASTER.DATA UID(APPMGRTLCAMS) READ(A)
 MASTER.DATA UID(APPPRGTLCAMS) READ(A) UNTIL(01/01/99)
```

This rule set specifies read access for the VSAM file PAYROLL.MASTER.DATA. The first rule entry allows read access to user APPMGRTLCAMS. The second grants read access to user APPPRGTLCAMS until (and including) January 1, 1997.

Access rules allow VSAM filenames of up to 44 total characters and up to eight characters per index level.

VSAM Allocation Protection Access Rules

CA ACF2 for VM validation is available for VSAM catalog management allocation and deallocation requests programs or the CMS AMSERV command issue.

CA ACF2 for VM VSAM AMSERV protection validates the VOL parameter of an access rule only when the VOL parameter is provided. Not all VSAM catalog requests provide the VOLSER. If an access rule contains a VOL parameter and the AMSERV request does not provide the VOLSER information, that rule does not match the request, access is denied, and a violation message is issued. We recommend that you do not use the VOL parameter in VSAM allocation rules.

VSAM Allocation Considerations

Data set names VSAM generates and related VSAM dataspaces can require special considerations for allocation processing. For this reason, we recommend you name both the data and index portions of the VSAM cluster with additional data set name qualifiers of DATA and INDEX, respectively.

Cluster-name

A.B.C

Data-name

A.B.C.DATA

Index-name

A.B.C.INDEX

For read or write access of a VSAM cluster, authorization is required only for the cluster name. Normal data set name access rules are sufficient for normal user processing. For allocate access, authorization is also required for the data and index names. In the following example, user TLCAMS can read and write to the cluster and user TLCPJM can define, delete, and alter the cluster:

```
$KEY(A)
 B.C UID(TLCAMS) READ(A) WRITE(A)
 B.C- UID(TLCPJM) ALLOC(A)
```

Protecting VSAM Dataspaces

Names are not given to VSAM dataspaces. When a VSAM dataspace or page space is created or deleted, CA ACF2 for VM validates access at the volume level. For dataspaces, VSAM generates an internal name, such as Z9999999x. CA ACF2 for VM recognizes the VSAM naming convention and generates a pseudo data set name in the format @volser.VOLUME or VOLUME.@volser. CA ACF2 for VM searches for a volume rule that allows access.

For example, both of the following access rule sets let user TLCSDH define or delete VSAM dataspaces on the volume VSAM01:

```
$KEY(@VSAM01)
 VOLUME UID(TFINTLCSDH) ALLOC(A)
```

-or-

```
$KEY(VOLUME)
 @VSAM01 UID(TFINPAYSDH) ALLOC(A)
```

The NOVOLRULE|VOLRULE operand of the RULEOPTS VMO record determines the format of the pseudo data set name that specifies VSAM dataspaces.

DASD ATTACH Commands Access Rules

CA ACF2 for VM access rules control real DASD devices. You can attach real DASD devices to a virtual machine using the CP ATTACH command or the DEDICATE VM directory control statement.

The dsn format for attachable DASD devices is Rccuu where R indicates real devices and is a required constant. You must specify the device address portion of the rule as a four-digit hexadecimal number.

In the examples that follow, references to the ATTACH command also apply to the DEDICATE VM directory control statement. Both ATTACH and DEDICATE perform the same function, except that ATTACH is a command issued from a user console, and DEDICATE is a VM directory control statement that causes an automatic ATTACH when a virtual machine is logged on. CA ACF2 for VM validates both the same way.

If you use permanently-mounted DASD devices, such as 3390s, you might construct rule entries for ATTACH commands as shown below:

```
$KEY(SYSTEM)
*ALLOW REAL 130 TO BE ATTACHED TO UID ACCPGRDLT
 R0130 UID(ACCTLCDLT) WRITE(A)
*ALLOW UID ACCTLCDLT TO ATTACH THE REAL 230
 R0230 UID(ACCTLCDLT) WRITE(A)
```

This rule lets any user attach DASD device 130 to ACCPGRDLT. This rule applies to all ATTACH 130 or 230 commands TLCDLT issued, regardless of the volume serial number of the volume mounted on these two devices. This rule is sufficient for DASD devices that always have the same volume mounted. We do not recommend this rule format if you use removable DASD devices (such as 3330s) because it does not effectively control access to a particular volume serial number (specific disk pack).

When you use removable DASD devices, use the VOL parameter to indicate the specific volume serial number that a user can ATTACH. The following rule allows any DASD device that has volume PAY001 mounted, to be attached to UID ACCTLCDLT:

```
$KEY(SYSTEM)
*ALLOW PAY001 TO BE ATTACHED TO UID ACCTLCDLT
 R- UID(ACCTLCDLT) VOL(PAY001) WRITE(A)
```

This rule provides more flexibility and control over attachable devices because PAY001 can only be attached to user ACCTLCDLT only if volume PAY001 is mounted. You can specify the VOL parameter as a mask, such as PAY***. If you specified the rule shown above, in VOL(PAY***), user ACCTLCDLT could attach any DASD device that had a volume with a serial number beginning with PAY.

You can use other combinations of the dsn and VOL parameters to provide the proper balance between flexibility and control. For example, the rule shown below lets user ACCTLCDLT attach only 13x DASD devices that have a PAY volume mounted:

```
$KEY(SYSTEM)
*ALLOW UID ACCTLCDLT TO ATTACH ANY 13x WITH PAYxxx VOLUME
 R013* UID(ACCTLCDLT) VOL(PAY***) WRITE(A)
```

You can enforce tape volume access protection through CA ACF2 for VM validation and system operator procedures.

TAPEVOLS Requirement for Tape Volume Access Rules

For validation, you must specify the volume serial label (volser) for the volume in the TAPEVOLS VMO record. Volume serial labels must consist of one to six alphanumeric characters. They can be masked using asterisks in any or all of the six characters. For example, in TAPEVOLS(767305, 123***) the TAPEVOLS field specifies that CA ACF2 for VM validates tape volumes with volser 767305 and volsers beginning with 123, followed by zero to three alphanumeric characters. The default TAPEVOLS specification is null, no volumes are protected tape volume access validation does not occur. Issue the SHOW STATE subcommand to display the current TAPEVOLS setting. For more information on the TAPEVOLS field, see the “About the Logonid Record” chapter.

Writing Tape Volume Access Rules

CA ACF2 for VM validates volumes that are defined in the TAPEVOLS field list through tape volume access rules. The rule set syntax for tape volume access is similar to that of any other access rule set:

```
[ volser|VOLUME   ]
[ UID(uid)        ]
[ SOURCE(source)  ]
[ SHIFT(shift)    ]
[ UNTIL(date)     ]
[ FOR(days)       ]
[ READ(A|L|P)     ]
[ WRITE(A|L|P)    ]
[ DATA(data)      ]
[ NEXTKEY(ruleid) ]
```

The access types, for example, are syntactically identical and have the same meanings:

- READ(A) specifies that read access to a tape is allowed.
- WRITE(A) specifies that write access to a tape is allowed.

The volser value is specified as:

- @volser.VOLUME: The $KEY value is the tape volume label (volser) with the prefix @. The pseudo data set name (or filename) is VOLUME. A separate rule set is required for each tape volume secured in the TAPEVOLS list. For example, with the previous TAPEVOLS list, the following rule sets apply:

```
$KEY(@767305)
 VOLUME UID(*****TLC) WRITE(A)

$KEY(@123AAA)
 VOLUME UID(*****TLC) WRITE(A)
```

- VOLUME.@volser: The $KEY value is VOLUME. The pseudo data set name (or filename) is the tape volume label (volser) with the prefix @. Only one rule set is required for all tape volumes secured in the TAPEVOLS list. For example, with the previous TAPEVOLS list, the following rule set applies:

```
$KEY(VOLUME)
 @767305 UID(*****TLC) WRITE(A)
 @123AAA UID(*****TLC) WRITE(A)
```

Use the NOVOLRULE|VOLRULE operand of the RULEOPTS VMO record to determine how to specify the volser in tape volume access rules. If you specify VOLRULE, you must specify the volser of a tape volume in the @volser.VOLUME format. If you specify NOVOLRULE (the default), you must specify the volser of a tape volume in the VOLUME.@volser format.

Issue the SHOW STATE subcommand to display the current VOLRULE setting. For more information on the VOLRULE field of the RULEOPTS VMO record, see the “Defining Structured Infostorage Records” chapter.

To access a tape volume, you must enter the ACFMOUNT command specifying the volser of the tape. If you enter the command syntax correctly and the volser is defined in the TAPEVOLS VMO record, a tape volume access rule is validated to see if the user is allowed to read from or write to the volume. If access is allowed, CA ACF2 for VM prompts the system operator to mount the indicated volume. (You can enforce additional validation through command limiting of ACFMOUNT.)

Another aspect of CA ACF2 for VM protection for tape volumes concerns tape volume reuse protection, implemented through the ACFERASE utility that erases all of a tape's data. The system operator responsible for the protection of tape volumes is typically given authority to use this utility, although an exec or program normally calls ACFERASE.

DASD Dump Restore and FORMAT Access Rules

You can write access rules to control the use of the DASD Dump Restore (DDR) and FORMAT service programs for protecting minidisks. This control is achieved through a unique CA ACF2 for VM validation process.

To understand this validation process, observe the following rule set. The order of the rule entries is how they are arranged when the rule set is compiled, sorted, and then decompiled. It is the order that the entries are searched when access to PAYROLL is attempted. Look at the following example:

```
$KEY(PAYROLL)
 V0191.- UID(ABCTLCNLT) R(A) W(A)
 V0191.- UID(ABCTLC) PGM(DDR)
 V0191.- UID(ABCTLC) PGM(FORMAT)
 V0191.- UID(ABCTLC) R(A) W(A)
```

The first rule entry lets user ABCTLCNLT read and write to the Payroll Department's 0191 minidisk. This access inherently lets ABCTLCNLT use the DDR and FORMAT programs on that minidisk.

You can write additional, more specific rule entries to prevent a user who has read or write access to a minidisk from using DDR and FORMAT. This is illustrated with the rest of the entries in the rule set.

The last rule entry lets all users in the Payroll Department (other than ABCTLCNLT) have read and write access to the same 0191 minidisk. These users are prevented from using the DDR and FORMAT programs because of the second and third rule entries.

- The second rule entry prevents users from having DDR access on the minidisk. More specifically, they are prevented from using the DDR DUMP function to move data from DASD onto tape. They are also prevented from using the DDR RESTORE function to return data onto DASD.

  To use DDR DUMP if a PGM(DDR) rule entry is in effect, a user must be granted PGM(DDR) read access through the R(A) or R(L) access permissions. To use DDR RESTORE if a PGM(DDR) rule entry is in effect, a user must be granted PGM(DDR) write access through the W(A) or W(L) access permissions. Additionally, another rule entry must allow the use of the FORMAT program.

- The third rule entry prevents users from having FORMAT access on the minidisk. More specifically, they are prevented from using the DDR RESTORE function to return data onto DASD. They are also restricted from using the FORMAT program to prepare DASD for use (eliminating all previous data).

  To use DDR RESTORE if a PGM(FORMAT) rule entry is in effect, a user must be granted PGM(DDR) write access in a previous rule entry (such as the second entry in the example) through the W(A) or W(L) access permissions and PGM(FORMAT) write access through the W(A) or W(L) access permissions.

  When DDR RESTORE is executed, the PGM(FORMAT) validation is not performed when the previous validation- performed through PGM(DDR)-denies the access. Also, if the previous PGM(DDR) validation logs but allows the access, two loggings are created instead of one if the PGM(FORMAT) validation also causes a logging.

  To use FORMAT if a PGM(FORMAT) rule entry is in effect, a user must be granted PGM(FORMAT) write access through the W(A) or W(L) access permissions.

After the rule set is compiled and sorted, the DDR and FORMAT rule entries must appear before any rule entries that let the user access the minidisk. The previous example illustrates this clearly.

VM Batch Environment and Subsystem Access Rules

You must write BATCHMON VM access rules to let the VM BATCH's execution machines link to VM BATCH's 0192 and 0194 minidisks. Use the rule ID of VMD4AUTH for this purpose.

```
$KEY(VMD4AUTH)
 V0192.- UID(VMD4-) READ(A)
 V0194.- UID(VMD4-) READ(A)
```

This rule set lets any UIDs that begin with VMD4 link to VMD4AUTH's 0192 and 0194 minidisks for read access. Similarly, an access rule is necessary to allow a link to the VMBATx 0191 minidisk for read and write access.

```
$KEY(BATCH1)
 V0191.- UID(-) READ(A) WRITE(A)
```

Universal read and write access (UID(-)) is necessary because when VM BATCH jobs are submitted, VMBATx executes jobs on behalf of whoever submits them. The BATCHMON job you submit inherits your validated logonid and your authority.

The above BATCHMON entries are shown as samples only. Consult your local systems programmer for the appropriate values to be substituted for the user IDs and rule IDs at your site.

Merging Rule Sets

Similar files (such as production files) may require similar CA ACF2 for VM validation. The following rule sets provide an example of using NEXTKEY to merge multiple rule sets:

```
$KEY(ACCT01)
 V0191.A.DATA NEXTKEY(ACCTXX)

$KEY(ACCT02)
 V0191.A.DATA NEXTKEY(ACCTXX)

$KEY(ACCT03)
 V0191.A.DATA NEXTKEY(ACCTXX)

$KEY(ACCT25)
 V0191.A.DATA NEXTKEY(ACCTXX)
```

In the following screen, accounting personnel (UIDs that match the mask ACCTUID) have READ and WRITE access to all of the above accounting files, each falling under a different rule ID. (They can belong to different users.) Rather than write a rule entry that allows access to each file, use the NEXTKEY operand to direct CA ACF2 for VM to one main rule set (ACCTXX) to govern all of these indexes. That rule set would be written as follows:

```
$KEY(ACCTXX)
$PREFIX(ACCT**)
%RCHANGE ACCTMGR
 V0191.A.DATA UID(ACCTUID) R(A) W(A)
```

The %RCHANGE control statement in this example establishes that the account manager (ACCTMGR) is responsible for writing and maintaining this rule set (after a security administrator establishes it). The account manager can change individual rule entries in the rule set, but not the control statements.

The NEXTKEY rule set must contain the $PREFIX control statement to ensure that the VM user ID associated with the specified files matches the file ID masks specified.

There are two ways to control access to one particular file. When you use NEXTKEY to merge rule sets, you can enclose the data set name in single quotes to specify the CA ACF2 for VM file ID (such as ACCT29.V0191.A.DATA) as a rule entry in the alternate ACCTXX rule set:

```
$KEY(ACCTXX)
$PREFIX(ACCT**)
%RCHANGE ACCTMGR
 'ACCT29.V0191.A.DATA' UID(ACCTUID01) R(P) W(P) E(A)
 V0191.A.DATA UID(ACCTUID) R(A) W(A)
```

Here, user ACCTUID01 has only EXECUTE access to ACCT29.V0191.A.DATA. All other users (including ACCTUID01) have READ and WRITE access to all accounting files.

Another way to control access permission to a particular file is to place a rule entry in the original ACCT29 rule set to specify READ, WRITE, and EXECUTE access for ACCTUID01, still retaining the NEXTKEY rule entry to govern all other access attempts:

```
$KEY(ACCT29)
 V0191.A.DATA UID(ACCTUID01) R(P) W(P) E(A)
 V0191.A.DATA NEXTKEY(ACCTXX)
```

Splitting Rule Sets

The NEXTKEY feature can also split an access rule set. This may be necessary to selectively delegate rule maintenance (%CHANGE or %RCHANGE) authority. Or, it may be necessary if a rule set is very large and exceeds the 4K physical storage size limit.

For example, you can have several entries for a rule set, all under the same VM user ID, TLCAMS. The NEXTKEY feature can redirect or split the rule set for VM TLCAMS into smaller sets as follows:

```
$KEY(TLCAMS)
 V0191.- NEXTKEY(A)
 V0192.- NEXTKEY(B)
 V0193.- NEXTKEY(C)
```

In this example, the first three rule entries specify the NEXTKEY rule sets to validate access to the V0191, V0192, and V0193 minidisks and the CMS files that reside on them. You can then write three smaller rule sets as follows:

```
$KEY(A)
$PREFIX(TLCAMS.V0191)
%RCHANGE PAYDIR
 - UID(PC1) R(A) W(A)
 - UID(PC2) R(A) W(A)
 - UID(PC3) R(A) W(A)
 - UID(MGR) R(A)

$KEY(B)
$PREFIX(TLCAMS.V0192)
%RCHANGE OPSDIR
 - UID(OPR) R(A)

$KEY(C)
$PREFIX(TLCAMS.V0193)
$RCHANGE PRGDIR
 - UID(PRG) R(A)
```

These rule sets allow the delegation of %RCHANGE rule authority, but are smaller than a single rule set required for VM user ID TLCAMS. Specify the $PREFIX control statement to append TLCAMS to each minidisk and CMS file ID. For more information about NEXTKEY, see the “About Access Rules” chapter.

In the first rule set for V0191 files, the Director of Payroll (PAYDIR) can change rule entries governing only TLCAMS files through the %RCHANGE control statement. Similar authority is delegated to the Director of Operations (OPSDIR) in the V0192 rule set, and to the Director of Programming (PRGDIR) in the V0193 rule set.

Payroll Clerks (PC1, PC2, and so on) need READ and WRITE access to V0191 files to update them. These accesses are specified in the first rule set governing V0191 files. The Payroll Manager (MGR) is only given READ access to V0191 files.

Computer operators (OPR) must have READ access to only the V0192 files. This access is allowed in the second rule set ($KEY(B)) governing TLCAMS.V0192.-. Similarly, programmers (PRG) can link for READ access only to the V0193 files in the third rule set above.

CA ACF2 for VM validation is directed to the rule set specified in the NEXTKEY option only when access based on the current rule set is prevented. You can have a chain of up to 25 NEXTKEY options. If you specify more than 25,
CA ACF2 for VM denies access and writes a KEYEXECS trace record that appears on the ACFRPTDS report. The same rule set cannot be referenced twice during a single validation. That is, the chain of NEXTKEY options cannot form a loop. If the same rule set is referenced twice, the access is denied and a NKEYLOOP trace record is written to the ACFRPTDS report. See the *Reports and Utilities Guide* for complete information about ACFRPTDS and NEXTKEY reporting.

How CA ACF2 for VM Sorts Rules

The CA ACF2 for VM rule compiler sorts the rule entries to ensure that
CA ACF2 for VM validates accesses properly. You can place a $NOSORT control statement in the rule set and specify $NOSORT in the RULEOPTS VMO record to prevent the compiler from sorting the rule entries. The compiler also converts all rule data into a form that CA ACF2 for VM can easily read when it validates an access request. It orders the rules according to the following criteria:

1. DSN patterns, from most specific to most general.
2. VOL patterns, from most specific to most general. VOL is valid only for system access rules (attachable DASD).
3. UID patterns, from most specific to most general.
4. PGM patterns, from most specific to most general. In VM, PGM is valid for FORMAT and DDR only.
5. SHIFT operands, in alphabetical order, with “none specified” last.
6. SOURCE operands, in alphabetical order, with “none specified” last.
7. UNTIL dates, from earliest to latest.

The first rule entry that matches the actual file, UID, source, shift, program, and date being used (the defined environment) is the rule entry that determines the access privileges.

Using ACF COMPILE to Build Access Rules from a Terminal

The COMPILE subcommand builds a set of access rules. The syntax for this subcommand is:

```
           [ List|NOList   ]
 COmpile * [ Store|NOStore ]
           [ Force|NOForce ]
```

*****

Indicates the data that follows is entered into the compiler. In an online environment, the system prompts you for the information.

List|NOList

LIST displays the compiler input on your terminal screen. This is the default. NOLIST suppresses the display.

Store|NOStore

The STORE option stores the rule set at compilation time. NOSTORE does not store the rule set. To store the rule, issue the STORE subcommand.

Force|NOForce

The NOFORCE option compiles a file containing multiple rule sets, but only stores rule sets that do not already exist on the Rule database. FORCE (the default) stores all compiled rule sets.

Compiling an Access Rule from a CMS File

When entering a rule set into a file, you can use the COMPILE subcommand of ACF to enter the rule set into the Rule database. The filetype of the rule set must always be RULE. CA ACF2 for VM automatically attempts to store your rules (that is, add them to the database). The syntax for this subcommand is:

```
COmpile fn [ List|NOList  ]
           [ Store|NOStore]
           [ Force|NOForce]
```

fn

Indicates the filename you entered when you write a rule in a specific file and want to compile it from that file. This file must have a filetype of RULE. Do not use filenames of LIST or NOLIST, FORCE or NOFORCE, or STORE or NOSTORE (or any of their valid abbreviations) when you compile data set access rules because you will receive error messages.

List|NOList

LIST displays the compiler input on your screen. This is the default. NOLIST suppresses the input display.

Store|NOStore

The STORE option stores the rule set when CA ACF2 for VM compiles it. NOSTORE does not store the rule. To store it, issue the STORE subcommand. STORE is the default.

Force|NOForce

The NOFORCE option compiles a file containing multiple rule sets, but only stores rule sets that do not already exist on the Rule database. FORCE (the default) stores all compiled rule sets.

CA ACF2 for VM uses the normal CMS search order when looking for the specified filename. If you have not previously entered rules into a file, you can use the COMPILE and STORE subcommands to enter them directly into CA ACF2 for VM:

```
acf
ACF

COMPILE (filename)
ACFpgm510I ACF COMPILER ENTERED

 $KEY (index)

  (rule entry)
  (rule entry)

  (press ENTER)

ACFpgm551I TOTAL RECORD LENGTH=length BYTES - percent PERCENT
UTILIZED

ACF

STORE
```

Ditto Function

CA ACF2 for VM also provides a convenient method for repeating specific components of a rule entry. For example, you can write rule entries to allow the same group of users access to your files. You can use a ditto, or quotation mark, to repeat that group (UID parameter) in several rule entries:

```
ACF
acf
COMPILE
$KEY(MAINT)
 V0191.BACKLIDS UID(AUT*1*A) SOURCE(GRP1) R(A) W(A)
 V0191.BACKRULE UID(“) SOURCE(“) R(A) W(A)
```

When CA ACF2 for VM compiles the above rule, the UID (“) and SOURCE(“) entries are copied from the V0191.BACKLIDS rule entry. This feature is limited to values specified in rule entries only. Ditto marks do not apply to the control statements of the rule set.

Because a ditto mark (“) in CP is the default terminal escape character, we recommend that you redefine the default CP escape character to a character that you do not use very often.

Using ACF STORE to Store Access Rules

The STORE subcommand under the RULE setting stores the previously compiled access rule set. This is available if you used the NOSTORE operand with the COMPILE subcommand. The syntax for the STORE subcommand is:

STore [Force|NOForce]

Force|NOForce

The NOFORCE option compiles a file containing multiple rule sets, but only stores rule sets that do not already exist on the Rule database. FORCE (the default) stores all compiled rule sets.

Using ACF DECOMP to Display Access Rules

Use the DECOMP subcommand to display access rules stored on the Rule database. You can list a rule set on a terminal or in a CMS file. To DECOMP rules belonging to other users, you must have at least the AUDIT privilege. You can set this privilege in the RULEOPTS VMO record. Contact your system administrator for more information about the CA ACF2 for VM privilege levels. The syntax for the DECOMP command is:

```
        { *              }
 DEComp { ruleid         } [INTO(fn)]
        { LIKE(rulemask) }
```

*****

Specifies that CA ACF2 for VM displays the last rule you referenced.

ruleid

Identifies a specific rule ID ($KEY) to display.

LIKE(rulemask)

Lists multiple rules at one time. All rules that match the rulemask are decompiled and displayed.

INTO(fn)

Puts decompiled rule sets into a file named fn (filename) on your A-disk. The filetype is always RULE. If rule sets exist in the file, CA ACF2 for VM adds the newly decompiled rules to the end of the file.

After you have built the rule, you can display it just as CA ACF2 for VM sees it. That is, in the exact order of their CA ACF2 for VM selection (from most specific to most general). The DECOMP subcommand displays your rules this way.

It is a good idea to use a file to build rules. If you had compiled (saved) your rule directly and then found an error, you would have to start at the beginning and enter the entire rule set again. If that rule set is in a file, edit the file and recompile the rule.

Changing an Access Rule

To modify an existing access rule set with the ACF command, follow these three steps:

1. Decompile the rule with the ACF DECOMP subcommand. (You can also use ACFDCMP-an advanced CA ACF2 for VM command explained in the chapter “Advanced CA ACF2 for VM Commands” in this guide.)
2. Edit the CMS file or the online version of the rule set to reflect the changes.
3. Compile the rule set using the ACF COMPILE subcommand. (You can also use ACFCOMP-an advanced CA ACF2 for VM command explained in the chapter “Advanced CA ACF2 for VM Commands” in this guide.)

To decompile the rule into a CMS file, follow the steps shown below:

```
ACF
decomp tlcams into(tlcams)

ACFDCM556I-ACCESS rule TLCAMS stored by TLCPJM on 01/03/98-11:21
ACFDCM551I Total record length=301 bytes - 7 percent utilized

ACF
end

xedit tlcams rule a
 ...
```

Edit the rule as you would any other CMS file, then compile the rule.

```
acf
ACF
compile tlcams
ACF compiler entered
...
Total record length=240 bytes - 5 percent utilized
Rule TLCAMS replaced
```

CA ACF2 for VM displays the entire rule set.

Using ACF DELETE to Delete an Access Rule

A security administrator can delete access rules from the Rule database with the DELETE subcommand. The syntax of the DELETE subcommand is:

```
DELete { ruleid       }
       { RULE(ruleid) }
```

ruleid

Deletes one specific access rule set only if you are in RULE mode.

RULE(ruleid)

Identifies the individual access rule set you want to delete if you are in either RULE mode or ACF command mode.

Using ACF TEST to Test an Access Rule

Use the TEST subcommand to test access rule sets. This testing tells you whether the rule set validates all attempts to access data. While the TEST subcommand is active, CA ACF2 for VM interprets only access rules. This testing does not take into account any site-specific system options or attributes of the logonids being tested. The syntax for the TEST subcommand is:

```
TEST { *      }
     { ruleid }
```

*****

Indicates that you want to test the previously compiled access rule set.

(no parameter)

When you specify TEST without a parameter, the TEST subcommand operates the same as when you specify an asterisk.

ruleid

Identifies the $KEY value of the access rule set.

TEST Subcommand Keywords

A period indicates you have activated the TEST subcommand. You can enter any of the following keywords with appropriate values to specify a test access environment. Separate each keyword with blank characters. You can specify one or more input lines.

DATE(date)

Specifies when access to a rule is tested. (This date can be in the format mm/dd/yy, dd/mm/yy, or yy/mm/dd.) The appropriate format is defined in the DATE field of the OPTS VMO record. The TEST subcommand uses the current date as the default.

DIR(sfsdirectory)

Specifies an actual SFS directory name and indicates the rule you are testing is for an SFS file. If you specify the DIR keyword, you must also specify a value for FILEPOOL. A null value indicates that this SFS file was defined without any directory levels.

DSN(dsn)

Specifies the actual filename for testing access. Refer to the FNFT keyword for information about SFS file filenames and filetypes.

FILEPOOL(filepool)|FPOOL(filepool)

Specifies an actual VM Shared File System (SFS) filepool and indicates the rule you are testing is for a SFS file or SFS directory. A null value, FILEPOOL(), turns off SFS testing and returns to CMS or OS/390 file testing.

FNFT(filename.filetype)

This keyword is a synonym of the DSN keyword. We include it to ease entering a SFS file's filename and filetype. If you specified FILEPOOL(filepool) to do SFS testing, then you can enter the filename and filetype of a SFS file. If you specify a null value, FNFT(), you test a SFS directory access. If you specify any value, you must separate the SFS filename and filetype with a dot. Additional levels are invalid. For example:

FNFT(PROFILE.EXEC)

Tests access to PROFILE EXEC

FNFT()

Tests access to a SFS directory

FNFT(PROFILE)

Invalid for SFS testing

FNFT(PROFILE.EXEC.A)

Invalid for SFS testing

LIBRARY(libmask)

Specifies a library you want to test access from. CA ACF2 for VM validation skips rule entries with LIBRARY specified. You can use this keyword to test OS/390 access rule entries.

LID(lid)

Specifies the logonid to obtain the UID. You must have access authority to the specified logonid record.

PGM(pgmmask)

This is an OS/390 keyword. If you specify this keyword in a rule entry, CA ACF2 for VM rule validation skips it. You can use this keyword to test OS/390 access rule entries.

SOURCE(sourceid)

Specifies the logical name of the input source or source group.

TIME(hhmm)

Specifies the time, in hours (hh) and minutes(mm), CA ACF2 for VM uses to test the access.

UID(uid)

Specifies the actual UID. This keyword does not require access to the logonid record.

VOLUME(vol)

Specifies the pattern volume set or set of volumes where the data must reside to be valid. VM validation uses the VOL keyword for attach rules only.
CA ACF2 for VM validation skips rule entries with the VOL keyword specified in all other rules. You can use this keyword to test OS/390 access rule entries.

Testing SFS File IDs

Specify the FILEPOOL keyword to activate the testing of VM Shared File System (SFS) files and directories. Specify directory names with the DIR keyword. Specify filenames and filetypes with the FNFT or DSN keywords.

When testing SFS accesses, CA ACF2 for VM does not process the PGM, VOLUME, LIB, and DDNAME values.

TEST Subcommand Results

The results of the TEST subcommand show whether read, write, allocate, and execute access to the volume or file is “A” (for ALLOW), “L” (for LOG), or “P” (for prevent). For a SFS rule entry, CA ACF2 for VM only displays the read and write accesses.

A

ALLOW Access is allowed.

L

LOG Access is allowed but logged.

P

PREVENT Access is explicitly prevented.

If no rule entry specifically applies to the test access environment, CA ACF2 for VM displays the following message:

```
ACFpgm74CI No rule applies, access would be denied
```

Subsequent Keyword Input

After a result is displayed, you can make another entry of keywords to test another rule set environment. Remember, after you enter TEST subcommand keywords, the values that you specify remain in effect until you change them. CA ACF2 for VM assumes that nearly all values you do not specify are completely masked, by default. To terminate the TEST subcommand, enter:

```
END
```

If you enter END with other valid TEST keywords, CA ACF2 for VM performs the access test, displays the results, and terminates the TEST subcommand.

Chapter 13: About Resource Rules

CA ACF2 for VM also provides protection for system resources. CA ACF2 for VM defines some of these resources, but you can also define your own resources. This protection is provided through resource rules that are similar to access rules, but belong in their own category. Resource rules are stored on the Infostorage database and have a storage class value of "R."

This chapter contains information on:

- Types of resource rules
- Resident type list support
- Masking resource rule keys
- How CA ACF2 for VM sorts rules
- Resource rule set components
- Setting the resource rule mode
- The CA Standard Security Facility (CAISSF).

This section contains the following topics:

Types of Resource Rules

With CA ACF2 for VM, you can protect VM account support, the AUTOLOG and XAUTOLOG commands, access to group user IDs, the DIAL command, IUCV, and VMCF resources. To protect a resource, you must write and compile a resource rule. Then, indicate the type of resource to CA ACF2 for VM with a type code. The CA ACF2 for VM-supplied type codes for resource rules are:

ACT

For resource rules that protect CA ACF2 for VM account validation

ALG

For resource rules that protect the target of the AUTOLOG command

DIA

For resource rules that protect the target of the DIAL command

DSP

For resource rules that protect VM dataspaces

GRP

For resource rules that protect logging onto group machines

IUC

For resource rules that protect the Inter-User Communication Vehicle (IUCV)

PGR

For resource rules that protect POSIX

VMC

For resource rules that protect the Virtual Machine Communications Facility (VMCF).

These type codes can be different at your site. Locally-defined type code values must contain three alphanumeric characters. You must define the values for these resource rules. To implement the rules, define the resources that are protected and issue CA ACF2 for VM access validation requests from your locally-written application programs through the System Request Facility (SRF).

Names of Resource Rules

There are individual resource names in each resource type code, each one unique from others of the same type code. Each name identifies an actual resource to be protected through a resource rule. Resources can have names from 1- to 40-characters long. The length of the resource name depends on the application. For example, CA ACF2 for VM-defined resource names for the type codes are generally one- to eight-characters long.

Resident Type List Support for Resource Rules

A resident type list lets you mask a resource rule set's $KEY value so you can validate more than one resource against the rule. An explanation of resource rule set syntax, including masking the $KEY value, follows shortly.

There are two ways to build resident type lists for resources:

- Use the TYPES field of the RESTYPE VMO record to automatically build a permanent list for up to 64 resource types.
- Use the ACFSERVE RELOAD RESOURCE command to manually build a list not defined in the TYPES field of the RESTYPE VMO record.

By default, the TYPES field of the RESTYPE VMO record automatically builds resident type lists for resources with type codes ACT, ALG, DIA, GRP, IUC, and VMC. These values are the default values for the type codes CA ACF2 for VM defines. Resident type lists are automatically built, by default, for all resource rules with type lists that CA ACF2 for VM defines. This includes account support, autolog, group logon, dial, and IUCV and VMCF resource rules.

Masking Resource Rule Keys

If you built a resident type list (or directory) for the resource type code through the TYPES field of the RESTYPE VMO record or the ACFSERVE RELOAD RESOURCE command, you can mask the resource name in the $KEY value.

A mask for a resource rule name follows the conventions of logonid masking, except that you cannot use a dash (-) as a masking character. (See the “About the Logonid Record” chapter for information about logonid masking.) The asterisk is the only character that you can use for resource masking. A resource name mask can contain a dash imbedded between other characters because dashes are treated literally as characters and not a mask. With masking, you can write a resource rule set for a group of resources and still write a unique rule set for a single resource in that group. For example, $KEY(G*******) applies to all resources of a given type code whose names are from one-to eight- characters and begin with G. $KEY(GET) applies specifically to the resource GET. When validating access to this resource, CA ACF2 for VM first checks the resource rule with the $KEY value GET (versus G*******), because GET is a more specific resource name and CA ACF2 for VM always sorts rules from most-specific to most-general.

Masking is especially useful when you want to create your own resource types. You can standardize the names of resources of a certain type so certain names fit a particular mask, letting one rule set control access to those resources. Masking a resource name in a resource rule follows the conventions of UID masking in access rules, (see the “About Access Rules” chapter for more information.) except that you cannot use a dash (-) as a masking character.

How CA ACF2 for VM Sorts Rules

The rule compiler converts the rule set input into a form that the rule interpreter can use during verification. The compiler also orders the rules according to the following criteria:

1. UID patterns, from most specific to most general
2. SOURCE parameters in alphabetical order, with “none specified” last
3. SHIFT parameters in alphabetical order, with “none specified” last
4. UNTIL dates, from earliest to latest.

The first active rule whose environment matches the actual access attempt determines the access permission.

Sample Resource Rule Set

CA ACF2 for VM uses the type code (the TYPE value) and resource name (the $KEY value) to identify a resource rule set. Here is an example of a resource rule set:

```
RESOURCE

compile
ACFCMP510I ACF compiler entered

$KEY(ABCD) TYPE(123)
 UID(*****TLCAMS) ALLOW

ACFCMP551I Total record length=152 bytes - 3 percent utilized

RESOURCE
```

This rule set is interpreted as follows:

$KEY(ABCD)

Indicates that the rule set applies to the resource ABCD.

TYPE(123)

Defines the type code of the rule set. This type code is the default CA ACF2 for VM or your site defined to validate site-specific resource rules.

UID(***tlcams)**

Identifies a user with the UID mask TLCAMS for validation against the resource ABCD with a type code of 123.

ALLOW

Indicates that TLCAMS can access the resource ABCD of type code 123, without being logged.

Most resource rule sets are usually larger than this sample.

Resource Rule Set Components

A resource rule set consists of the following:

1. Control statements (identified by the $ or % symbols in the first column).
2. Comment statements that are also rule set statements (identified by the * symbol in the first column).
3. Rule entries (must all follow the control statements). When you enter a rule as input from a terminal or create or update a file used as compiler input, the rule input text **must** begin in the second column. If a rule entry is too long for one line, enter a blank and a dash at the end of the line to indicate that this rule entry is continued on the next line. The continuation line must begin in the second column.

A resource rule set is similar to an access rule set except that it is associated with a type code and has some structural differences. The complete syntax for a resource rule set is:

```
$Key(resource)
$Type(typecode)
 [ $Nosort                          ]
 [ $Prefix(prefix)                  ]
 [ $Recname(record-name)            ]
 [ $Userdata(text)                  ]
 [ %Change uidmask1,uidmask2,...    ]
 [ %RChange uidmask1,uidmask2,...   ]
 [ *comment                         ]
 [ rsrcmask                         ]
 [ UId(uidmask)                     ]
 [ SOurce(sourceid)                 ]
 [ SHift(shiftname)                 ]
 [ ACtive(date)                     ]
 [ UNtil(date)|FOR(days)            ]
 [ SERVICE(READ|UPDATE|ADD|DELETE)] 
 [ Data(text)                       ]
 [ Nextkey(nextkey)                 ]
 [ Reccheck(recordid)               ]
 [ Verify                           ]

  {Allow|Log|Prevent}
```

- The lines beginning with $ and % show the syntax for the various types of control statements.
- All $ and % control statements must precede rule entries.
- The entries beginning in column two show the syntax for a single rule entry. There are several such entries in a rule set. Rule set text can be variable format or fixed format 80-byte records (with the sequence field as the last eight characters in the record).

Any CMS file containing rule text exceeding 80 characters is changed into continuation lines. This makes editing and printing the file easier. The compiler accepts input files that already exist with record lengths up to 256 characters.

The components of this resource rule set are defined below.

$KEY(resource)

Indicates the resource name being compiled. This name can be up to 40 characters, depending on the resource rule application. For example, resource names for the CA ACF2 for VM-defined type codes generally are one-to eight-characters long and often represent a value, such as a logonid. The TYPE value of the $KEY control statement specifies the three-character type code that identifies the type of resource being protected.

$TYPE(typecode)

Specifies an alternate way to indicate the resource type. The resource type is generally specified on the $KEY control statement, as explained previously. The CA ACF2 for VM-defined type codes are located in the TYPES field of the RESTYPE VMO record. These default type codes are listed in Types of Resource Rules section.

You can also use the ACFSERVE RELOAD RESOURCE subcommand to reload and create specific resource types. See the “Using the ACFSERVE Commands” chapter for detailed information about how to use this subcommand. Even though CA ACF2 for VM supplies these type codes, you can customize them to identify different type codes.

$NOSORT

Specifies that CA ACF2 for VM will not sort this resource rule set if you define the NOSORT parameter in the RULEOPTS VMO record.

$PREFIX(prefix)

Specifies a value that overrides the rule set key as a prefix to all resource names in this rule set. You can specify up to 24 characters. If you specify $PREFIX(), the prefix is set equal to the $KEY entry and the $PREFIX control statement is not generated when the rule is decompiled. CA ACF2 for VM issues a warning message indicating that the $PREFIX specified is null and is ignored (optional).

$RECNAME(recname)

Used for OS/390 record-level protection. CA ACF2 for VM does not use this information. $RECNAME specifies the name of a RECORD definition record. For more information, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guide.*

Note: CA ACF2 for VM cannot maintain RECORD definition records.

$USERDATA(text)

Specifies any text string, up to 64 characters. Unlike the text in comment statements that is removed from the rule set at compilation time, the text on the $USERDATA control statement is stored with the rule set. You can also use the DATA parameter for a resource rule entry to store information in a rule entry.

%CHANGE uidmask1,uidmaskn,...

Specifies a mask (or masks) for the UIDs of users who can store and delete this rule set. The user or users given change authority can further delegate this authority to other users. Do not specify a dash (-) as a masking character at the end of a %CHANGE UID mask as CA ACF2 for VM interprets it as a continuation character. There is no need to specify this dash because all UID values are padded out to their full length.

%RCHANGE uidmask1,uidmaskn,...

Specifies who has **restricted** CHANGE authority over the rule set. The designated users can change individual rule entries, but not control statements. They cannot delegate change authority or delete the rule set. If the same user matches entries in %CHANGE and %RCHANGE, %CHANGE takes effect (optional).

Comment Statements

Indicates comment statements that begin with an asterisk (*) in column one and can appear anywhere in the input. They let you place any text inside an uncompiled rule set. This text is lost when the rule set is compiled. To retain comments when the rule set is compiled, use the $USERDATA control statement or the DATA parameter of a resource rule entry.

rsrcmask

Specifies additional qualifiers in the resource name. You must specify the first-level qualifier in the $KEY control statement to specify additional qualifiers for the rsrcmask parameter. CA ACF2 for VM places a period between the first and second level qualifiers.

You must specify rsrcmask as the first parameter in the rule entry. A resource name can be up to 256 characters, including the characters specified in the $KEY. You can use the dash masking character (just as in access rules) to represent any number of characters up to the 256 character limit. For example:

```
$KEY(PAYT) TYPE(CKC)
 EMPLOYEE.EMERGENCY.DATA.- UID(PAYUSR1) SERVICE(UPDATE)
 EMPLOYEE.- UID(PAYUSR) SERVICE(READ)
```

See the Access Rule Masking section in the “About Access Rules” chapter for more details on masking.

You cannot enclose the value you supply for rsrcmask in single quotes.

UID(uidmask)

Specifies a UID or mask of UIDs that identify the users attempting access. If you omit this parameter, it includes all users. UID masking in resource rules follows the same standards as UID masking in access rules.

SOURCE(sourceid)

Specifies the name of a source or source group where the access attempt is being made. If you omit this parameter, it includes all sources.

SHIFT(shiftname)

Specifies the one- to eight-character name of the SHIFT record limiting the time, date, and day of the week when a user can access this resource. If you omit this parameter, it applies to any time or day.

ACtive(date)

Specifies a Gregorian date in the form mm/dd/uu, yy/mm/dd, or dd/mm/yy, (depending on the site option) that is the first date when this rule is considered valid. **Note:** See the DATE field of the OPTS VMO record for the site option information. Years specified as 70-99 assume a date in the 20th century (1970-1999). Years specified as 00-69 assume a date in the 21st century (2000-2069). This parameter is valid only when the VMO RULEOPTS RULELONG parameter is set.

UNTIL(date)

Specifies the last date an access can take place under this rule; that is, for the rule to apply, the access attempt must be on or before the UNTIL date. This date must be in the format specified in the DATE operand of the OPTS VMO record (mm/dd/yy, yy/mm/dd, or dd/mm/yy).

FOR(days)

Specifies the number of days from the compilation date of the resource rule set when a user can access under this rule. The minimum number that you can specify is zero (the compilation date). The maximum is 365. CA ACF2 for VM converts all values specified in this parameter to an UNTIL value when the rule is compiled.

SERVICE(READ|ADD|UPDATE|DELETE)

Specifies the type of access associated with the request. You can specify one or more keywords, separated by blank characters or commas. If you omit this parameter, all types of access are allowed. Keywords are:

READ

Read-only access

ADD

Access to add new records

UPDATE

Access to modify existing records

DELETE

Access to delete records.

The SERVICE keyword is valid only for site-specific resource rules and CA ACF2 for VM data space validation. For the resource rules specific to CA ACF2 for VM, it is ignored.

DATA(text)

Specifies up to 64 characters of data for your own use. This string is kept with the rule entry and displayed when the rule set is decompiled. Your site can have standards concerning the format of this character string. The data is passed to the site exits and back to the calling application subsystem with the global $USERDATA field. This data can indicate further checking or contain some control information.

VERIFY

Requests password validation for any access attempts made under this rule. The application subsystem requesting access to a resource is informed of the request for password validation. The VERIFY keyword is valid only for site-specific resource rules. For the resource rules specific to CA ACF2 for VM, it is ignored.

NEXTKEY(nextkey)

Specifies the key of an alternate rule set that CA ACF2 for VM should check if access to this resource is **denied** based on this rule set. The NEXTKEY parameter functions the same in resource rules as it does in access rules. For more information, see the Using NEXTKEY section in the “About Access Rules” chapter.

Reccheck(recid)

Used for OS/390 record-level protection. Specifies the name of an EXPRESSN record that you want CA ACF2 for VM to use for this validation. The EXPRESSN record defines a Boolean expression that CA ACF2 for VM evaluates to determine whether a user can access a record based on the contents of a field. For more information about EXPRESSN records and record-level protection, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guid*e.

You cannot use CA ACF2 for VM to maintain EXPRESSN records.

ALLOW

Specifies that access to the resource is allowed (without creating a logging record).

LOG

Specifies that access to the resource is allowed, but logged. CA ACF2 for VM writes an SMF record to log the event.

PREVENT

Specifies that access to the resource is prevented. CA ACF2 for VM writes an SMF record to log the event.

If you do not specify ALLOW, LOG, or PREVENT, PREVENT (the default) is assumed. The following sections explain the control statements and rule entry parameters.

Rule Set Control Statements

The control statements identify the resource rule set and determine some of the rule set characteristics. The $KEY control statement is the only statement that is required. If you do not specify the TYPE value, the $TYPE control statement is required.

Each control statement must begin in column one. You can specify any number of $ or % control statements in the compiler input. If you enter the same type of $ control statement more than once, CA ACF2 for VM uses only the last statement of that type. You can specify more than one of each type of % control statement.

You can use a dash (-) as the last nonblank character on the line to continue all input to the compiler on multiple statements. A continuation is recognized unconditionally. For instance, if a %CHANGE statement is continued,
CA ACF2 for VM treats the next line as a continuation of this control statement even if that line has the format of another control statement.

Individual Rule Entries

You must specify individual resource rule entries after the control statements in the rule set. These entries identify the environment and access permissions when you can access a resource. A rule entry can extend up to 72 positions. Get into the practice of starting rule entries in the second column to avoid treating a rule entry as a comment when that entry begins with an asterisk. Like control statements, a rule entry continues unconditionally from one statement to the next when a blank space followed by a dash (-) appears at the end of a line. If a dash appears at the end of a rule entry and the next line contains a comment, CA ACF2 for VM assumes the comment is a continuation of the rule entry.

Setting Resource Rule Mode

You can create, display, and maintain resource rules with the ACF subcommands. You can process the rules after you establish the RESOURCE setting (with an appropriate type code) of the ACF command. CA ACF2 for VM processes all resource records under the RESOURCE setting of the ACF command. The type code must be three alphanumeric characters. Use the SET subcommand to establish this setting. The syntax for the SET subcommand is:

```
     { RESOURCE(ACT)  }
     { RESOURCE(ALG)  }
     { RESOURCE(DIA)  }
SEt  { RESOURCE(DSP)  }
     { RESOURCE(GRP)  }
     { RESOURCE(IUC)  }
     { RESOURCE(VMC)  }
     { RESOURCE(PGR)  }
     { RESOURCE(typ)  }
```

The parameters for the SET subcommand are defined below.

ACT

Specifies account support

ALG

Specifies autolog resource rules

DIA

Specifies dial resource rules

DSP

Specifies VM dataspace resource rules

GRP

Specifies group logon resource rules

IUC

Specifies IUCV resource rules

PGR

Specifies POSIX resource rules

VMC

Specifies VMCF resource rules

typ

Specifies site-defined resource rules, a one- to three-alphanumeric character resource type code.

The type code values (ACT, ALG, GRP, DIA, IUC, and VMC) are the CA ACF2 for VM-supplied values specified in the TYPES field of the RESTYPE VMO record. The value you set must agree with its respective operand definition in this record. After you establish the RESOURCE setting and appropriate type code, you can issue various ACF subcommands to process resource rule sets. For a list of valid ACF subcommands that you can use under the RESOURCE setting, see the "Using the ACF Command" chapter.

What is CAISSF?

The CA Standard Security Facility (CAISSF) is the CA-CIS security layer for all CA products. It controls and validates access to all system and application resources. Other CA products and applications can ask CAISSF if access is allowed for a specific user. CAISSF provides such benefits as centralizing resource access control and reducing administrative costs for training and day-to-day administration.

CAISSF Components

CAISSF consists of the following components:

- Product-supplied CAISSF calls
- CAISSF facilities
- CA ACF2 for VM security interfaces
- CA ACF2 for VM security authorizations

- VMO resource translation records
- CA ACF2 for VM rules.

These components are described in the following sections.

Product-supplied CAISSF Calls

Most CA products have their own built-in security schemes. This can cause problems because of the administrative overhead to maintain separate authorizations. CAISSF solves this problem and gives you the ability to centralize this function.

You can ask CA ACF2 for VM to create SMF records for CAISSF requests. You also have the ability to review the CA ACF2 for VM databases and cross-reference users to resources.

Every CA product with an interface to CAISSF publishes a *Security Administrator Guid*e. This guide describes:

- Which CAISSF facilities are used
- Which security interfaces are required
- Which CA ACF2 for VM security authorizations are required
- Which resource types are used
- Which resource names are protected

To activate CAISSF security in a product, read the requirements in the *Security Administrator Guid*e and set them up as described. When you have completed all the tasks and have tested the rules, activate CAISSF security for the product. After you have activated CAISSF, CA ACF2 for VM validates all access attempts and passes access recommendations back to the product. The **product** then allows or denies the access.

CAISSF Facilities

CAISSF provides facilities for the other vendor products and for CA security products. The products use the facilities to ask CAISSF for validation. CAISSF asks CA ACF2 for VM for recommendations. The facilities invoke a number of CA ACF2 for VM services. These services include:

- Password validation and password change
- Logon access control
- Resource access validation
- Data access validation
- User information extract.

The CA ACF2 for VM services each product uses are entirely dependent on the product and vary based on product needs. For example, products you can dial into can use logon access control, but if a user is logged onto VM, the product can use the password validation service. (The *Security Administrator Guid*e lists all vendor products with the CA ACF2 for VM services needed.)

CA ACF2 for VM Security Interfaces

CAISSF is a multilayered security facility. Internally, the layers in the CA ACF2 for VM environment include:

- Product CAISSF calls
- CAISSF router
- CA ACF2 for VM CAISSF translator
- System Request Facility (SRF) modules
- CA ACF2 for VM processing and validation routines.

Each product that uses CAISSF has calls to the CAISSF router for the services it needs to invoke. The CAISSF router determines if a security product is installed, what release is running, and then loads the appropriate CAISSF translator.

The CAISSF router is a CMS module called CAS9SEC. The CA ACF2 for VM CAISSF translator is also a module. Its name depends on the release you are running. The naming convention for the translator is CAS9pcrr), where pc is the product code and rr is the release. For CA ACF2 for VM, the product code is M9 and the current release is 4.3. That yields a translator module name of CAS9M943.

In most cases, the CA ACF2 for VM CAISSF translator converts CAISSF calls into SRF calls. For password validations without a new password, CA ACF2 for VM uses diagnose A0 subcode 4. The CA ACF2 for VM CAISSF translator makes calls to various programs that are part of the SRF and reside in the ACFSRF LOADLIB. In turn, SRF uses the built-in processing and validation routines installed with CA ACF2 for VM. These include the routines in the control program (CP) of VM and the CA ACF2 for VM service machine.

CA ACF2 for VM Security Authorizations

CAISSF uses the CA ACF2 for VM SRF facilities to ask for access recommendations. In most cases, this requires that the product's service machine be authorized to make SRF calls. Some of the required CA ACF2 for VM authorizations required are:

- @SRF definition in the ACFFDR
- VMSAF logonid record privilege

- SRF logonid record privilege.

(You can find the CA ACF2 for VM security authorizations for each product in each product's *Security Administrator Guid*e.)

VMO Resource Translation Records

Each product's CAISSF calls are made against different resources. For example, SUBMIT, CACMD (CA command), and so on. In a CA ACF2 for VM environment, these resources are translated to three character resource types. CA ACF2 for VM has a default VMO record called SSFTYPE that defines what the translation values are.

The SSFTYPE VMO record translates CAISSF resources to CA ACF2 for VM resource types. The CA ACF2 for VM resource type is the value you need to specify in the $TYPE statement in any resource rule for the product. For example, the SUBMIT resource has a default resource type of SUB. (The resources that each product calls is defined in the product's *Security Administrator Guid*e.) With CA ACF2 for VM, you can set a unique resource type for each SYSID. This allows you to have separate resource rules in a multi-CPU, shared database environment. You can issue the ACF SHOW STATE subcommand and view the RESOURCE CLASS LIST values to determine the active resource translation values.

CA ACF2 for VM Rules

Most products ask for access recommendations that are defined in CA ACF2 for VM rules. Users can request accesses that CA ACF2 for VM validates against access rules or resource rules. In most cases, CAISSF requests are against resource rules. The name of the resource is in the following form:

```
PRODUCT NAME.CLASS.FUNCTION.OPERAND
```

For example, a CA-Director ADMIN function of ADDGROUP might have the resource name DIRECTOR.ADMIN.ADDGROUP, which you would specify as the $KEY value of a resource rule.

With CA ACF2 for VM, you can control the level of rule writing. That is, you can determine how general or how specific you want a rule to be. To do this, mask the name of the resource. For example, you could specify one resource rule to protect all CA-Director ADMIN functions with a $KEY of DIRECTOR.ADMIN.********.

With masking, you can write as many or as few rules as you want. Continuing with the above example, you could mask all CA-Director ADMIN functions in one rule, DIRECTOR.ADMIN.******** and have a specific rule that controls CA Director DIRGEN functions with a rule like DIRECTOR.ADMIN.DIRGEN. You could also protect all CA Director functions in one rule: CA-Director.**********.

If you mask any rule keys, you must have a resource directory for the resource type. Resource directories are explained in the topic RESTYPE Record-Resident Type List Support in the chapter “Defining Structured Infostorage Records” in this guide. Issue the ACF SHOW STATE subcommand to determine the active resource directories.

Chapter 14: Maintaining Resource Rules with the Full-Screen Feature

This chapter provides information for maintaining resource rules through the full-screen feature.

After you finish this chapter, you will be able to use the full-screen feature to:

- Add a resource rule
- Create a resource rule set (including rule entries and control statements)
- Display an existing resource rule
- Modify a current resource rule
- Delete a resource rule (and verify its deletion)
- Test a resource rule set before it is stored on the Infostorage database.

This section contains the following topics:

Writing Resource Rules

To get to this screen, enter the full-screen feature described in the chapter “Using the Full-Screen Feature” and select option 3, Resource Control, from the Primary Option Menu. Use this screen to select the type of resource access control you want to do. Enter your selection on the option line.

```
M9HE-3000      Resource Control (3)                    CA ACF2 for VM
OPTION ===> ____________________________________________________________
                                                            TIME 12:43
      0 Change options or PF keys

      1 Add a Resource Ruleset

      2 Display Resource Ruleset(s)

      3 Change Resource Ruleset(s)

      4 Delete Resource Ruleset(s)

      5 Test a Resource Ruleset

      Option may be followed by “.type.rulekey” to select
      the type and ruleset to which the option is to apply.


PF1=Help        2=Print       3=Quit       4=Return      5=        6=
PF7=            8=            9=           10=           11=       12=Retrieve
```

You can also enter a number and a type.rulekey to indicate the rule set to be affected. The type indicates the CA ACF2 for VM or site-defined type code. It can be up to three characters long.

Valid options are:

0 Change options or PF keys

Displays the current values of the PF keys and lets you change them.

1 Add a Resource Ruleset

Lets you add new resource rule sets.

2 Display Resource Ruleset(s)

Lets you display current resource rule sets. You cannot update (or change) rules with this option.

3 Change Resource Ruleset(s)

Lets you change currently existing resource rule sets.

4 Delete Resource Ruleset(s)

Lets you delete current resource rules. Use caution with this option. If you delete a resource rule set accidentally, you must recreate it.

5 Test a Resource Ruleset

Lets you test a resource rule set before it is compiled and stored in the Infostorage database.

Changing Options and PF Keys

You see this screen if you select option 0, Change Options or PF Keys, from the previous screen. You can also select 3.0 from the Primary Option menu and go directly to this screen. Use this screen to change the options or PF keys for the **resource rules** screens only.

```
M9HE-0010     Change Options or PF Keys          CA ACF2 for VM
COMMAND ==> _____________________________________________________
                                                      TIME 12:44

  Options:
  ==>Y   Time(12/24) ==> ___
  ==>N   Language  ==> ___

           Description  Command
 PF1  ==>  Help         HELP
 PF2  ==>  Print        PRINT
 PF3  ==>  Quit         QUIT
 PF4  ==>  Return       RETURN
 PF5  ==>  Execute      EXECUTE
 PF6  ==>  _________    _________
 PF7  ==>  Backward     BACKWARD
 PF8  ==>  Forward      FORWARD
 PF9  ==>  Director     DIRECTOR
 PF10 ==>  Previous     PREVIOUS
 PF11 ==>  Next         NEXT
 PF12 ==>  Retrieve    RETRIEVE


PF1=Help        2=Print       3=Quit       4=Return      5=        6=
PF7=            8=            9=           10=           11=       12=Retrieve
```

Yes

Indicate how you want yes values displayed. Enter 1 (to indicate binary) or Y (to indicate alpha) in this field.

No

Indicate how you want no values displayed. Enter 0 (to indicate binary) or N (to indicate alpha) in this field.

Time(12/24)

Enter whether you want the time reported in a 12-hour (enter 12) or 24-hour (enter 24) clock.

Language

Enter the five-character code for the language you want to use for screens. CA supplies the following valid options:

AMENG

American English

UCENG

Upper case English.

Your site may have added additional language support. Contact your security administrator for this information.

To change the values for the PF keys, enter the value you want to display on the screen for each PF key in the Description column. Enter the actual command names (in capital letters) in the Command column.

Adding a Resource Rule

This section explains how to add a resource rule. When you have finished this section, you will know how to use the full-screen feature to:

- Write resource rules
- Add resource rule entries
- Delegate authority to other CA ACF2 for VM users to change resource rules that they do not own.

Creating a Resource Rule Set

To see this screen, select option 1, Add a Resource Ruleset, from the Resource Control (3) screen. You can also enter 3.1 from the Primary Option Menu and go directly to this screen.

```
M9PA-3110 Add Resource Ruleset Control Information (3.1.1) CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 12:46
Resource type ==> ___
Rulekey       ==> ________________________________________
Prefix        ==> ________________________
Recname       ==> ________________________
Userdata      ==> ____________________________________________________
Sort Option   ==> _


Use the 'Next' function to see more rule data.









PF1=Help      2=Print       3=Quit      4=Return    5=Execute     6=
PF7=          8=            9=          10=Previous 11=Next       12=Ret
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC. For information on these types codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

Prefix

Specifies a value that overrides the rule set key as a prefix to all resource names in this rule set. You can specify up to 24 characters.

Recname

Used for MVS record-level protection. CA ACF2 for VM does not use recname. Specifies the name of a RECORD definition record. For more information, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guid*e.

Note: CA ACF2 for VM cannot maintain RECORD definition records.

Userdata

Use this space for any comments you care to enter. This information does not affect the rule set. This field can contain up to 64 characters. Unlike comment statements that are removed when the rule set is compiled, CA ACF2 for VM stores the information you enter in this field with the rule set.

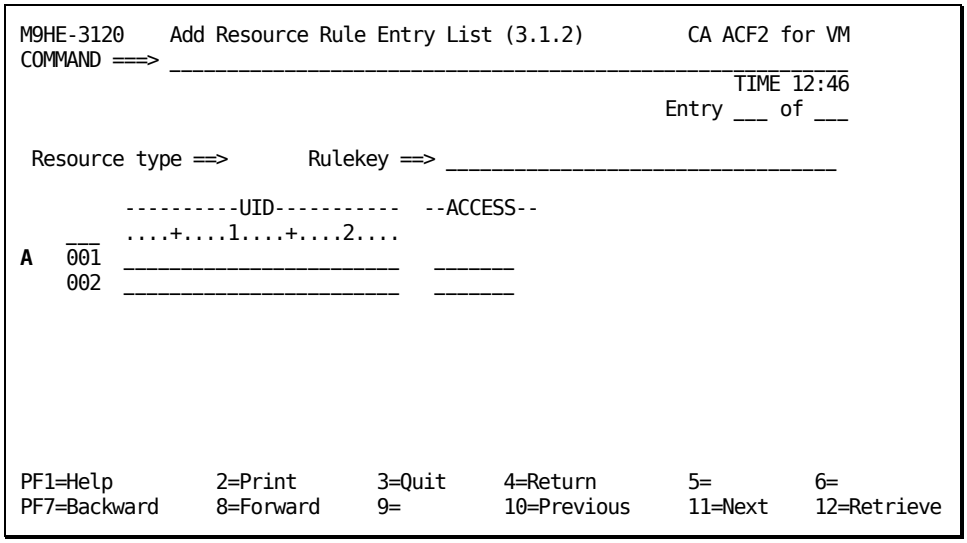
Sort option

Specifies how CA ACF2 for VM sorts the rule set. If you enter 1 or Y in this field, CA ACF2 for VM does not sort the rule set from most specific to most general. This field does not appear if sort option is not active at your site.

Press PF11 (or your designated NEXT key) to add more rule data.

Adding Resource Rule Entry Lists

To see this screen, press PF11 (or your designated NEXT key) after you have viewed the Add Resource Ruleset Control Information screen. Use this screen to add rule entries.



A

Indicates the prefix area for this screen. The prefix area is where you select rules or enter information.

This screen contains only summary information of the different rule entries. Enter an S (for SELECT) in the prefix area to go to the next screen and see all of the details for rule you selected.

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC For more information, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

UID

Specifies the User Identification string of the user that the rule was written for.

ACCESS

Specifies the access privileges that this user has. Valid access privileges are:

ALLOW

Allow access to the resource.

LOG

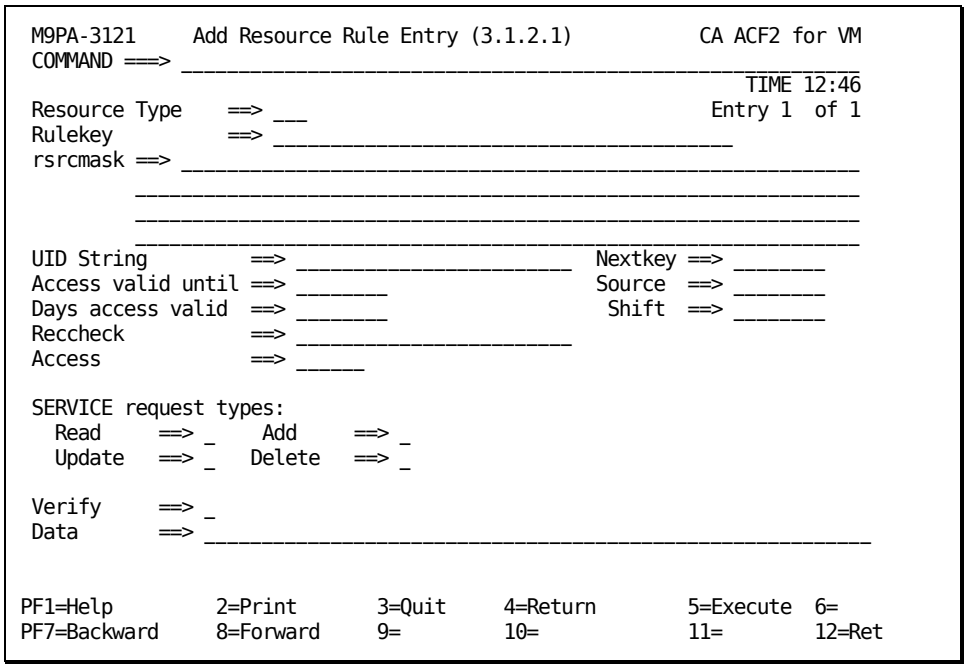
Allow access, but record the action.

PREVENT

Do not allow access.

Adding Resource Rule Entries

You see this screen if you entered an S to select a specific rule entry from the previous screen, Add Resource Rule Entry List (3.1.2). This screen lets you see all the details for the resource rule entries.



Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC. For more information, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

rsrcmask

Specifies additional qualifiers in the resource name. You must specify the first-level qualifiers in the rulekey if you want to specify additional qualifiers for the rsrcmask parameter. CA ACF2 for VM places a period between the first and second level qualifiers.

You must specify rsrcmask as the first parameter in the rule entry. A resource name can be up to 256 characters, including the characters specified in the rulekey. You can use the dash masking character (just as in access rules) to represent any number of characters up to the 256 character limit.

See the Access Rule Masking section in the “About Access Rules” chapter for more details on masking.

You cannot enclose the value you supply for rsrcmask in single quotes.

UID

Specifies the User Identification string of the user (or set of users) that this rule entry applies to. If you omit this field, the rule applies to all users (optional).

Nextkey

Specifies the key of an alternate rule set that CA ACF2 for VM should check if access to this resource is **denied** based on this rule set. The NEXTKEY parameter functions the same in resource rules as it does in access rules. For more information, see the Using NEXTKEY section in the “About Access Rules” chapter.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access this rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times.

Access valid until

Specifies the last date that this rule entry is valid. Valid dates are in the Gregorian date format (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending upon the DATE parameter of the OPTS VMO record (optional).

Days access valid

Specifies the number of days that this rule is considered valid, starting from the day that the rule set was compiled. The minimum number that you can specify is zero (today's date), and the maximum number is 365 (optional).

Reccheck

Used in MVS record-level protection. Specifies the name of an EXPRESSN record you want CA ACF2 for VM to use for this validation. The EXPRESSN record defines a Boolean expression that CA ACF2 for VM evaluates to determine whether a user can access a record based on the contents of a field. For more information about EXPRESSN records and record-level protection, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guid*e.

You cannot maintain EXPRESSN records from CA ACF2 for VM.

Access

Specifies the type of access to give to this user. Valid options are ALLOW (allow access to the resource), LOG (allow access, but record the action), and PREVENT (do not allow access). PREVENT is the default.

SERVICE request types

Specifies the type of access associated with the request. If you have defined local resources at your site, you must complete this field. CA ACF2 for VM-defined resources do not use this field.

In the SERVICE request types field, enter a single character response following each service permission. For example, to allow TLCAMS read access to a specific volume, but prevent all other permissions, enter a Y after the ==> in the Read field of the SERVICE request types field like this:

SERVICE request types:

Read ==> Y
Add ==> N
Update ==> N
Delete ==> N

Verify

Requests password validation for any access attempts made under this rule. This field is valid only for site-defined resource rules.

Data

Enter up to 64 characters in this comment field. This field is kept with the rule entry and displayed when you decompile the rule. This data can indicate further checking or contain some control information.

Adding Rule Set %CHANGE Information

To get to this screen, press Enter after you have completed the Add Resource Rule Entry (3.1.2.1) screen. Use this screen to grant the necessary authority to users so that they can make future maintenance changes to specified rule sets.

```
M9PA-3130    Add Ruleset %Change Information (3.1.3)   CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                            TIME 12:46
Resource type ==> ___  Rulekey ==> ____________________________________

UIDs allowed to change full ruleset (%CHANGE):      Entry 1  of 0
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________

UIDs allowed to change rule entries only (%RCHANGE):   Entry 1  of 0
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________
==> ____________________  ==> ____________________


PF1=Help         2=Print       3=Quit     4=Return       5=Execute  6=
PF7=Backward     8=Forward     9=         10=Previous    11=Next    12=Ret
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP DIA, IUC, and VMC. For more information on these type codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

UIDs allowed to change full ruleset (%CHANGE)

Specifies the UIDs of users with full change authority. These users can make changes to the rule set. This control statement indicates who can replace a particular set of rules. A security administrator can compile a rule set with only $KEY and %CHANGE control statements and establish a base rule set to distribute rule writing permissions (optional).

Do not specify a dash (-) as a masking character at the end of a %CHANGE UID mask as CA ACF2 for VM interprets it as a continuation character. There is no need to specify this dash because all UID values are padded out to their full length.

UIDS allowed to change rule entries only (%RCHANGE)

Specifies the UIDs of users who have limited change authority. These users can create rule entries, but cannot delegate the same authority to anyone else. The designated users cannot change control statements. They cannot further delegate change authority or delete the rule set. If the same user matches entries in both %CHANGE and %RCHANGE, the higher authority (%CHANGE) is in effect for that user (optional).

Entry ___ of ___

Specifies the %CHANGE UIDs listed first for this screen and the total number of %CHANGE entries found.

Displaying a Resource Rule

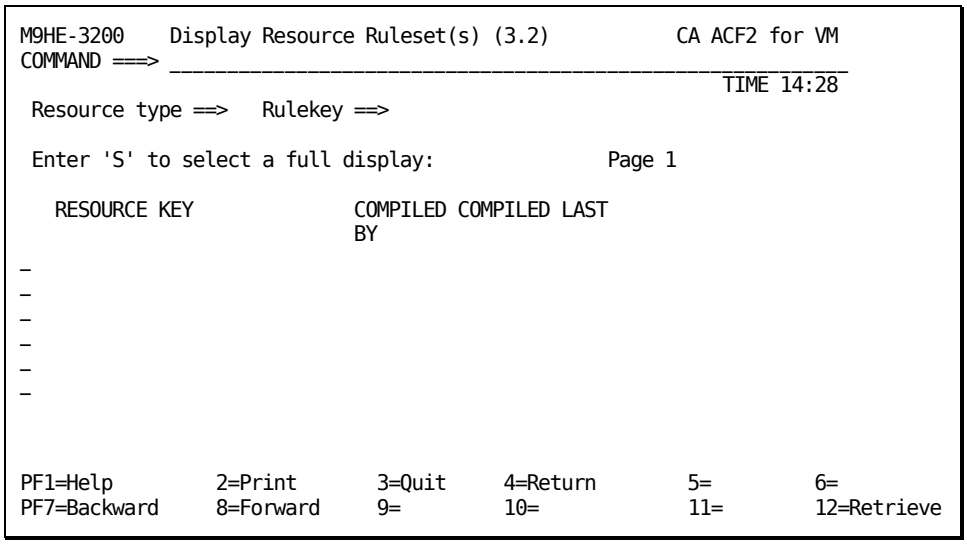
This section explains how you can display information about CA ACF2 for VM users. You must have the necessary authority to display user information.

When you finish this section, you will know how to use the full-screen feature to display:

- Summary and detailed resource rule information
- Who can change particular resource rules.

Displaying Resource Rule Sets

To get to this screen, select option 2, Display Resource Ruleset(s) from the Resource Control (3) screen. This screen displays summary information about resource rule sets and lets you select a rule for complete display.



This screen is not displayed if you passed an unmasked rule key from the menu screen or if the key had no masking. If you did not specify a rule key on the initial display of this screen, lines 6 through 21 are not displayed.

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.

CA ACF2 for VM-defined type codes are ACT, ALG, GRP DIA, IUC, and VMC. For more information on these type codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

Resource key

Specifies the $KEY of the resource rule set.

Compiled by

Specifies the logonid of the user that last compiled the rule set.

Compiled last

Specifies the date and time that the rule set was last compiled.

To see more information on any of the displayed rule sets, enter an S (for SELECT) on the field before the RESOURCE KEY and press Enter.
CA ACF2 for VM displays another screen with more detailed information.

Displaying Resource Rule Set Control Information

This screen displays more detailed information on the rule set that you selected on the Display Resource Ruleset(s) screen. This is a display-only screen.

```
M9PA-3210 Display Resource Ruleset Control Info. (3.2.1)    CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 14:29
Resource type      : ___
Rulekey            : ________________________________________
Prefix             : ________________________
Recname            : ________
Rule length        : ___
% Utilization      : __
Total Rule Entries : ___
Userdata     : ____________________________________________________
Sort Option  : _

Use the 'Next' function to see more rule data.









PF1=Help         2=Print       3=Quit     4=Return       5=         6=
PF7=             8=            9=         10=Previous    11=Next    12=Ret
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY of the resource rule set.

Prefix

Specifies a value that overrides the rule set key as a prefix to all resource names in this rule set. You can specify up to 24 characters.

Recname

Used for MVS record-level protection CA ACF2 for VM does not use recname. Specifies the name of a RECORD definition record. For more information, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guide*.

Note: CA ACF2 for VM cannot maintain RECORD definition records.

Rule length

Specifies the length of the compiled rule set. This field contains the number of bytes (from 1-496) in the rule set.

% Utilization

Specifies the amount of the database used.

Total rule entries

Specifies the number of rule entries currently in the rule.

Userdata

Use this space for any comments you care to enter. This information does not affect the rule entry. This field can contain up to 64 characters. Unlike comment statements that are removed when the rule set is compiled, information you enter in this field is stored with the rule set (optional).

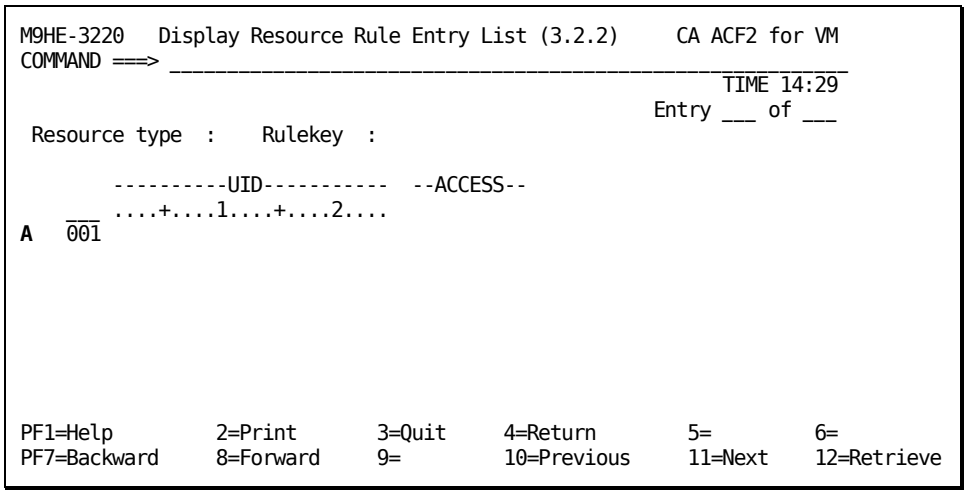
Sort option

Specifies how CA ACF2 for VM sorts this rule set. If you enter 1 or Y in this field, CA ACF2 for VM does not sort the rule set from most specific to most general. This field does not appear if sort option is not active at your site.

Use your backward and forward PF keys to see more rule data.

Displaying Resource Rule Entry Lists

This screen displays summary information for the rulekey you specified. This is a display-only screen. For detailed information on any of the rule entries, enter an S (for SELECT) in the prefix area and press Enter. Another screen is displayed.



A

Indicates the prefix area for this screen. The prefix area is where you select rules or enter information.

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY of the resource rule set.

UID

Displays the User Identification string of the user that the rule applies to.

ACCESS

Specify the type of access permission to give to this user in this field. Valid access permissions are:

ALLOW

Allow access to the specified resource.

LOG

Allow access, but record the action.

PREVENT

Do not allow access.

Entry _____ of ____

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Displaying Resource Rule Entries

You see this screen if you entered an S to select a specific rule entry from the previous screen, Display Resource Rule Entry List (3.2.2). This screen displays more detailed information about the rule you selected.

```
M9PA-3221    Display Resource Rule Entry (3.2.2.1)       CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                          TIME 14:30
Resource type   : ___                Entry 1  of 1
Rulekey         : ________________________________________
rsrcmask  : ____________________________________________________________
        ____________________________________________________________
        ____________________________________________________________
        ____________________________________________________________
UID String : _____________________  Nextkey  : ________
Access valid until : ________          Source  : ________
Days access valid  : ________          Shift   : ________
Reccheck           : _____________________
Access             : _______

SERVICE request types:
  Read      : _   Add       : _
  Update    : _   Delete    : _

Verify   : _
Data     : ____________________________________________________________

PF1=Help         2=Print       3=Quit      4=Return        5=         6=
PF7=Backward     8=Forward     9=          10=             11=        12=Ret
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY of the resource rule set.

rsrcmask

Specifies additional qualifiers in the resource name. You must specify the first-level qualifiers in the rulekey if you want to specify additional qualifiers for the rsrcmask parameter. CA ACF2 for VM places a period between the first and second level qualifiers.

You must specify rsrcmask as the first parameter in the rule entry. A resource name can be up to 256 characters, including the characters specified in the rulekey. You can use the dash masking character (just as in access rules) to represent any number of characters up to the 256 character limit.

See the Access Rule Masking section in the “About Access Rules” chapter for more details on masking.

You cannot enclose the value you supply for rsrcmask in single quotes.

Nextkey

Specifies the key of an alternate rule set that CA ACF2 for VM should check if access to this resource is denied based on this rule set. The NEXTKEY parameter functions the same in resource rules as it does in access rules. For more information, see the Using NEXTKEY section in the “About Access Rules” chapter.

UID

Specifies the User Identification string of the user (or set of users) that this rule entry applies to. If you omit this field, the rule applies to all users (optional).

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access this rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times.

Access valid until

Specifies the last date that this rule entry is valid. Valid input is in the Gregorian date format (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending upon the DATE parameter of the OPTS VMO record (optional).

Days access valid

Specifies the number of days that this rule is considered valid, starting from the day that the rule set was compiled. The minimum number that you can specify is zero (today's date), and the maximum number is 365 (optional).

Reccheck

Used in MVS record-level protection. Specifies the name of an EXPRESSN record you want CA ACF2 for VM to use for this validation. The EXPRESSN record defines a Boolean expression that CA ACF2 for VM evaluates to determine whether a user can access a record based on the contents of a field. For more information about EXPRESSN records and record-level protection, rever to the CA ACF2 for VM for OS/390 Administrator Guide.

You cannot maintain EXPRESSN records from CA ACF2 for VM.

Access

Specifies the type of access to give to this user. Valid options are ALLOW (allow access to the resource), LOG (allow access, but record the action), and PREVENT (do not allow access). PREVENT is the default.

SERVICE request types

Specifies the type of access associated with the request. If you have defined local resources at your site, you must complete this field. CA ACF2 for VM-defined resources do not use this field.

In the SERVICE request types field, enter a single character response following each service permission. For example, to allow TLCAMS read access to a specific volume, but prevent all other permissions, enter a Y after the ==> in the Read field of the SERVICE request types field like this:

```
SERVICE request types:
  Read   ==> Y
  Add    ==> N
  Update  ==> N
  Delete  ==> N
```

Verify

Requests password validation for any access attempts made under this rule. This field is valid only for site-defined resource rules.

Data

Enter up to 64 characters in this comment field. This field is kept with the rule entry and displayed when you decompile the rule. This data can indicate further checking or contain some control information.

Displaying Resource Rule Set %CHANGE Information

This screen displays users who can change the rule set you specified. This is a display-only screen.

```
M9PA-3230   Display Ruleset %CHANGE Information (3.2.3)  CA ACF2 for VM
COMMAND ===> _____________________________________________________________
                                                              TIME 14:30
Resource type  : ___  Rulekey  : ____________________________________

UIDs allowed to change full ruleset (%CHANGE):           Entry 1  of 0
 : ________________________   : ________________________
 : ________________________   : ________________________
 : ________________________   : ________________________
 : ________________________   : ________________________
 : ________________________   : ________________________
 : ________________________   : ________________________

UIDs allowed to change rule entries only (%RCHANGE):     Entry 1  of 0
 : ________________________   : ________________________
 : ________________________   : ________________________
 : ________________________   : ________________________
 : ________________________   : ________________________
 : ________________________   : ________________________
 : ________________________   : ________________________


PF1=Help         2=Print       3=Quit     4=Return       5=         6=
PF7=Backward     8=Forward     9=         10=Previous    11=Next    12=Ret
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY of the resource rule set.

UIDs allowed to change full ruleset (%CHANGE)

Specifies the UIDs of users with full change authority. These users can make any and all changes to the rule set. This control statement indicates who can replace a particular set of rules. A security administrator can compile a rule set with only $KEY and %CHANGE control statements and establish a base rule set to distribute rule writing permissions (optional).

%RCHANGE uidmask1,uidmask2,...,uidmaskn

Specifies who has **restricted** CHANGE authority over the rule set. The designated users can change individual rule entries, but not control statements. They cannot delegate change authority or delete the rule set. If the same user matches entries in %CHANGE and %RCHANGE, %CHANGE takes effect (optional).

Entry _____ of _____

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Use your designated backward and forward PF keys to add additional rule entries.

Changing a Resource Rule

This section explains how to use the full-screen feature to:

- Change resource rule sets
- Change rule set information
- Change rule entries in existing rule sets
- Redefine who can change specific resource rule sets.

Changing Resource Rule Sets

CA ACF2 for VM displays this screen if you selected option 3, Change Resource Ruleset(s) from the Resource Control (3) screen. This screen lets you change compiled rule sets. It is not displayed if you passed an unmasked rule key from the menu screen, or if the key had no masking. If you did not specify a rulekey on the initial display of this screen, lines 6 through 21 are not displayed.

```
M9HE-3300   Change Resource Ruleset(s) (3.3)           CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                          TIME 14:30
 Resource type ==>   Rulekey ==>

 Enter 'S' to select a full display:                          Page 1

   RESOURCE KEY              COMPILED COMPILED LAST
                             BY
_
_
_
_
_
_



PF1=Help         2=Print       3=Quit     4=Return       5=         6=
PF7=Backward     8=Forward     9=         10=Previous    11=Next    12=Retrieve
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC. For more information on these type codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

Resource key

Specifies the $KEY of the rule set.

Compiled by

Specifies the logonid of the user that last compiled the rule set.

Compiled last

Specifies the date and time that the rule set was last compiled.

To see more information on any of the displayed rule sets, enter an S (for SELECT) on the Resource Key field and press Enter. CA ACF2 for VM displays another screen with more detailed information.

Changing Resource Rule Set Control Information

This screen displays more detailed information about the rule set that you selected on the Change Resource Ruleset(s) screen and lets you change information.

```
M9PA-3310 Change Resource Ruleset Control Information (3.3.1) CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                              TIME 14:31
 Resource type  : ___
 Rulekey        : ________________________________________
 Prefix       ==> ________________________
 Recname      ==> ________________________
 Userdata     ==> ____________________________________________________
 Sort Option  ==> _



 Use the 'Next' function to see more rule data.



PF1=Help         2=Print       3=Quit     4=Return       5=Execute  6=
PF7=             8=            9=         10=Previous    11=Next    12=Ret
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY of the resource rule set.

Prefix

Specifies a value that overrides the rule set key as a prefix to all resource names in this rule set. You can specify up to 24 characters.

Recname

Used for MVS record-level protection CA ACF2 for VM does not use recname. Specifies the name of a RECORD definition record. For more information, see the CA ACF2 for VM for z/OS and OS/390 Administrator Guide.

Note: CA ACF2 for VM cannot maintain RECORD definition records.

Userdata

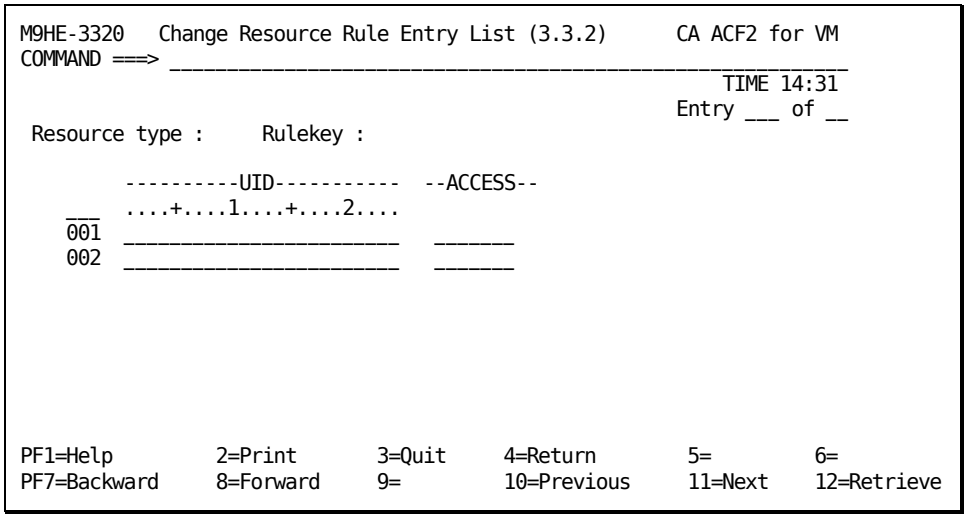
Use this space for any comments you want to enter. This information does not affect the rule entry. This field can contain up to 64 characters. Unlike comment statements that are removed when the rule set is compiled, information you enter in this field is stored with the rule set (optional).

Sort option

Specifies how CA ACF2 for VM sorts this rule set. If you enter Y or 1
CA ACF2 for VM does not sort the rule set from most specific to most general. This field does not appear on this screen if sort option is not active at your site.

Changing Resource Rule Entry Lists

With this screen, you can change rule entries in a rule set that already exists. It contains only summary information of the rule entries. Enter an S on the field beside the rule that you want to select. This brings you to another screen where you can see more details on the rule.



Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY of the resource rule set.

UID

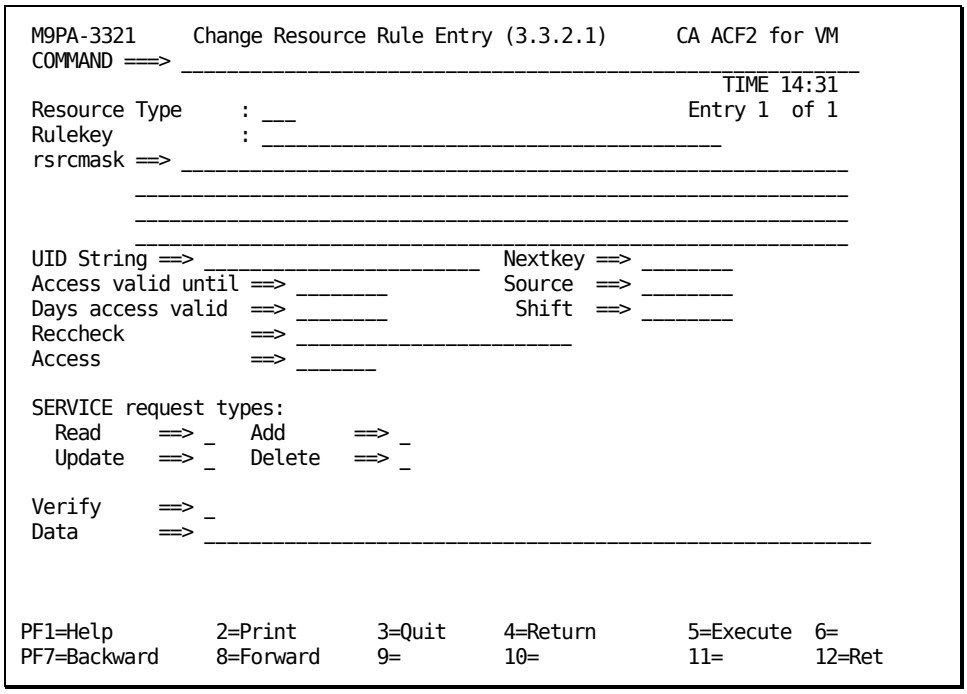
Specifies the User Identification string of the user (or group of users) that this rule applies to.

ACCESS(ALLOW|LOG|PREVENT)

Specifies the type of access to give to this user. Valid options are ALLOW (allow access to the resource), LOG (allow access, but record the action), and PREVENT (do not allow access). PREVENT is the default.

Changing Resource Rule Entries

You see this screen if you entered an S to select a specific resource rule entry from the previous screen, Change Resource Rule Entry List (3.3.2). This screen lets you see more detailed information and change the rule that you selected.



Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY of the resource rule set.

rsrcmask

Specifies additional qualifiers in the resource name. You must specify the first-level qualifiers in the rulekey if you want to specify additional qualifiers for the rsrcmask parameter. CA ACF2 for VM places a period between the first and second level qualifiers.

You must specify rsrcmask as the first parameter in the rule entry. A resource name can be up to 256 characters, including the characters specified in the rulekey. You can use the dash masking character (just as in access rules) to represent any number of characters up to the 256 character limit.

See the Access Rule Masking section in the “About Access Rules” chapter for more information about masking.

You cannot enclose the value you supply for rsrcmask in single quotes.

UID

Specifies the User Identification string of the user (or set of users) that this rule entry applies to.

Nextkey

Specifies the key of an alternate rule set that CA ACF2 for VM should check if access to this resource is denied based on this rule set. The NEXTKEY parameter functions the same in resource rules as it does in access rules. For more information, see the Using NEXTKEY section in the “About Access Rules” chapter.

Source

Specifies an input source or source group name where this rule should apply. For example, you can specify the ID of a terminal. The access is allowed only if the user logs onto the specific terminal. If you do not specify a source, any input source is valid. Ask your security administrator for a list of valid source group names (optional).

Shift

Specifies the name of the shift record on the Infostorage database that applies to this rule entry. It defines valid days, dates, and times that this rule entry is in effect. If you do not specify this parameter, any access this rule indicates is appropriately allowed, logged, or prevented for all days, dates, and times.

Access valid until

Specifies the last date that this rule entry is valid. Valid dates must be in the Gregorian date format (mm/dd/yy, dd/mm/yy, or yy/dd/dd), depending upon the DATE parameter of the OPTS VMO record (optional).

Days access valid

Specifies the number of days that this rule is considered valid, starting from the day that the rule set was compiled. The minimum number that you can specify is zero (today's date), and the maximum number is 365 (optional).

Reccheck

Used in MVS record-level protection. Specifies the name of an EXPRESSN record you want CA ACF2 for VM to use for this validation. The EXPRESSN record defines a Boolean expression that CA ACF2 for VM evaluates to determine whether a user can access a record based on the contents of a field. For more information about EXPRESSN records and record-level protection, rever to the CA ACF2 for VM for OS/390 Administrator Guide.

You cannot maintain EXPRESSN records from CA ACF2 for VM.

Access(ALLOW|LOG|PREVENT)

Specifies the type of access to give to this user. Valid options are ALLOW (allow access to the resource), LOG (allow access, but record the action), and PREVENT (do not allow access). PREVENT is the default.

SERVICE request types

Specifies the type of access associated with the request. If you have defined local resources at your site, you must complete this field. CA ACF2 for VM-defined resources do not use this field.

In the SERVICE request types field, enter a single character response following each service permission. For example, to allow TLCAMS read access to a specific volume, but prevent all other permissions, enter Y after the ==> in the Read field of the SERVICE request types field like this:

```
SERVICE request types:
  Read   ==> Y
  Add    ==> N
  Update  ==> N
  Delete  ==> N
```

Verify

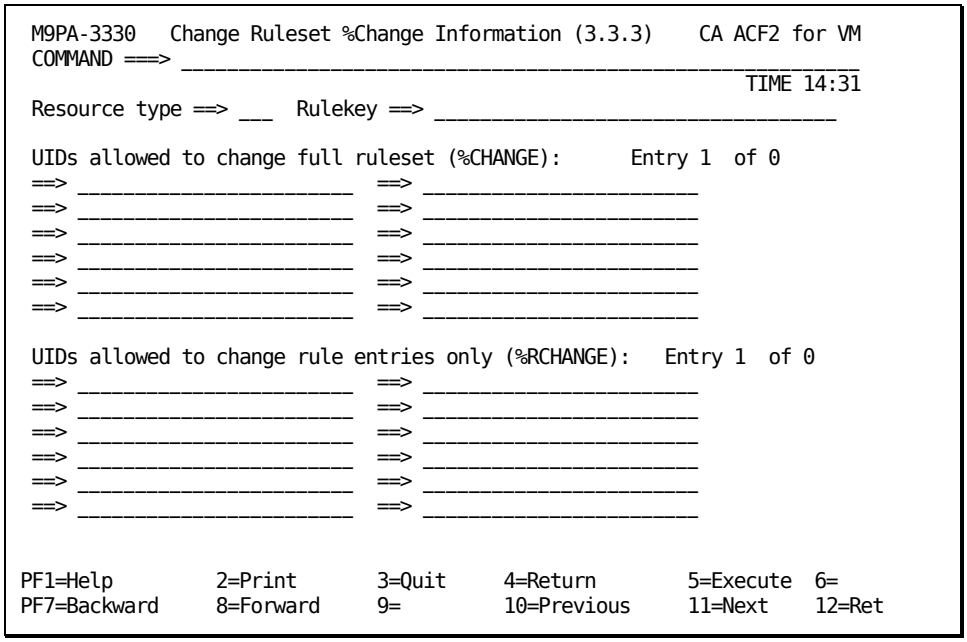
Requests password validation for any access attempts made under this rule. This field is valid only for site-defined resource rules.

Data

Enter up to 64 characters in this comment field. This field is kept with the rule entry and displayed when you decompile the rule. This data can indicate further checking or contain some control information.

Changing Resource Rule Set %CHANGE Information

This screen lets you change the users who can modify specified rule sets. To remove users, erase their UIDs from this list.



Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC. For more information on these type codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

UIDs allowed to change full ruleset (%CHANGE)

Specifies the UIDs of users with full change authority. These users can make any and all changes to the specified rule set. This control statement indicates who can replace a particular set of rules. A security administrator can compile a rule set with only $KEY and $CHANGE control statements and establish a base rule set to distribute rule writing permissions (optional). Do not specify a dash (-) as a masking character at the end of a %CHANGE UID mask as CA ACF2 for VM interprets it as a continuation character. There is no need to specify this dash because all UID values are padded out to their full length.

%RCHANGE uidmask1,uidmask2,...,uidmaskn

Specifies who has **restricted** CHANGE authority over the rule set. The designated users can change individual rule entries, but not control statements. They cannot delegate change authority or delete the rule set. If the same user matches entries in %CHANGE and %RCHANGE, %CHANGE takes effect (optional).

Entry ___ of ___

Specifies the number of this rule entry and the total number of rule entries in this rule set.

Deleting a Resource Rule

This section provides information on using the full-screen feature to:

- Delete resource rule sets
- Help prevent you from accidentally deleting a resource rule set.

Deleting Resource Rule Sets

CA ACF2 for VM displays this screen if you selected option 4, Deleting Resource Ruleset(s) from the Resource Control (3) screen. This screen lets you delete compiled rule sets. It is not displayed if you passed an unmasked rule key from the menu screen. If you did not specify a rulekey on the initial display of this screen, lines 6 through 21 are not displayed.

```
M9HE-3400    Delete Resource Ruleset(s) (3.4)           CA ACF2 for VM
COMMAND ===> _____________________________________________________________
                                                              TIME 14:31
 Resource type ==>   Rulekey ==>

 Enter 'S' to select a full display:                            Page 1

   RESOURCE KEY             COMPILED COMPILED LAST
                            BY
_
_
_
_
_
_


PF1=Help         2=Print       3=Quit     4=Return       5=         6=
PF7=Backward     8=Forward     9=         10=            11=        12=Retrieve
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC. For more information on these type codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

Resource key

Specifies the $KEY of the rule set.

Compiled by

Specifies the logonid of the user that last compiled the rule set.

Compiled last

Specifies the date and time that the rule set was last compiled.

Verifying a Resource Rule Set Deletion

This screen lets you delete the resource rule set you selected on the Delete Resource Ruleset(s) screen. It lets you verify that you really want to delete the rule before CA ACF2 for VM actually deletes it from the Infostorage database. If you delete a resource accidentally, you must recreate it.

```
M9PA-3410  Delete Resource Ruleset Control Info. (3.4.1)   CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                            TIME 14:32
Resource type   : ___
Rulekey         : ________________________________________
Prefix          : ________________________
Recname         : ________
Userdata        : ____________________________________________________
Sort Option     : _









PF1=Help        2=Print      3=Quit     4=Return       5=Execute  6=
PF7=            8=           9=         10=Previous    11=Next    12=Ret
```

Resource type

Specifies the three-character type code that defines the type of resource this rule set protects.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC. For more information on these type codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

Prefix

Specifies a value that overrides the rule set key as a prefix to all resource names in this rule set. You can specify up to 24 characters.

Recname

Used for MVS record-level protection. CA ACF2 for VM does not use recname. Specifies the name of a RECORD definition record. For more information, see the CA ACF2 for VM for z/OS and OS/390 Administrator Guide.

Note: CA ACF2 for VM cannot maintain RECORD definition records.

Rule length

Specifies the length of the compile rule set. This field contains the number of bytes (from 1-496) in the rule set.

% Utilization

Specifies the percentage of the database used.

Total rule entries

Specifies the total number of rule entries to be deleted from this rule set.

Userdata

Use this space for any comments you care to enter. Information you enter in this field does not affect the rule entry. This field can contain up to 64 characters. Unlike comment statements that are removed when a rule is compiled, the information you enter in this field is stored with the rule set (optional).

Sort option

Specifies how CA ACF2 for VM sorts this rule set. If you enter Y or 1,
CA ACF2 for VM does not sort the rule set from most specific to most general. If sort option is not active at your site, this field does not appear on your screen.

To delete the resource rule set, press your designated EXECUTE PF key. CA ACF2 for VM deletes the rule set from the Infostorage database.

Testing a Resource Rule

This section explains how you can test resource rules before you store them. When you have finished this section, you will know how to use the full-screen feature to test resource rules to be sure that they do what you want them to do.

Testing Resource Rules

You see this screen if you selected option 5, Test a Resource Ruleset, from the Resource Control (3) screen. This screen lets you select resource rule sets for testing. To use this screen, enter the KEY (or masks) of the rule sets you want to test.

```
M9HE-3500    Test Resource Ruleset(s) (3.5)              CA ACF2 for VM
COMMAND ===> ____________________________________________________________
                                                          TIME 17:12
 Resource type   ==>

 Rulekey         ==>

 Enter 'S' to select a Rule to Test:                        Page 1

  RESOURCE KEY COMPILED BY    COMPILED COMPILED BY  %UTIL

 ==> __
 ==> __
 ==> __









PF1=Help         2=Print       3=Quit     4=Return       5=         6=
PF7=Backward     8=Forward     9=Director 10=            11=        12=Retrieve
```

Resource type

Specifies the three-character type code that defines the type of resource protected by this rule set.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC. For more information on these type codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

The displayed information for a list of rule sets is described below:

Resource key

Specifies the key of the resource rule set

Compiled by

Specifies the logonid of the user that last compiled the rule set

Last compiled

Specifies the date and time the rule set was last compiled

%UTIL

Specifies the percentage of the database used.

Testing a Resource Rule Set

CA ACF2 for VM displays this screen if you selected option 5, Test a Resource Ruleset, from the Resource Control (3) screen. This screen lets you select resource rules for testing. To see this screen, enter the $KEY of the rule set that you want to test.

```
M9HE-3510    Test a Resource Ruleset (3.5.1)          CA ACF2 for VM
COMMAND ===> _____________________________________________________
                                                          TIME 17:13
What is being accessed:
 Resource type   ==>
 Rulekey         ==>

Who is attempting the access:
 Logonid         ==> ________ or UID  ==> ______________________

When and Where is this access occuring:
 Date ==> ________ Time ==> _____ Source ==> ________

What SERVICE is requested:
   Read    ==>
   Add     ==>
   Update  ==>
   Delete  ==>

Test results:
 Access     :          By Rule Entry  :


PF1=Help         2=Print       3=Quit     4=Return        5=         6=
PF7=             8=            9=         10=             11=        12=Retrieve
```

What is Being Accessed

Resource type

Specifies the three-character type code that defines the type of resource protected by this rule set.

Rulekey

Specifies the $KEY value of the rule set. The $KEY control statement supplies the name of the resource that this rule was written for. A resource rule name can represent many different objects, depending on the resource TYPE.
CA ACF2 for VM-defined type codes are ACT, ALG, GRP, DIA, IUC, and VMC. For more information on these type codes, see the “About Resource Rules” chapter. For example, for DIA, the $KEY rule ID is the ID of a user in the VM directory that another user wants to dial to. For ACT, the $KEY rule ID is an eight-character account code that you define. The rule ID can be up to 40 characters long. However, for most CA ACF2 for VM-defined resource types, only the first eight characters are significant. You can mask this field with standard CA ACF2 for VM masking characters.

Who is Attempting the Access

Logonid

Specifies the logonid of the user attempting the access.

or

UID

Specifies a pseudofield concatenation of selected information from the logonid record that can include information from user-defined fields, such as department, job function, and the logonid field.

Note: Pick only one of these fields. If you select both fields, CA ACF2 for VM uses the logonid field.

When and Where is This Access Occurring

Date

Specifies the date (in the format mm/dd/yy, dd/mm/yy, or yy/dd/mm, depending upon your preference) that the simulated access is attempted.

Time

Specifies the time of the access.

Source

Specifies the source of the simulated access.

What SERVICE is Requested

SERVICE request types

Specifies the type of access associated with the request. If you have defined local resources at your site, you must complete this field. CA ACF2 for VM-defined resources do not use this field.

In the SERVICE request types field, enter a single character response following each service permission. For example, to allow TLCAMS read access to a specific volume, but prevent all other permissions, enter Y after the ==> in the Read field of the SERVICE request types field like this:

```
SERVICE request types:
  Read    ==> Y
  Add     ==> N
  Update  ==> N
  Delete  ==> N
```

Test Results

The access permission is displayed following the access type. Valid options are ALLOW (allow access to the resource), LOG (allow access, but record the action), and PREVENT (do not allow access). The By rule entry field indicates the rule entry line number for the access.

Chapter 15: Maintaining Resource Rules with the ACF Command

This chapter provides information for using the ACF command to process resource rules. The ACF command lets you process resource rules through line commands instead of the full-screen panels. Whatever method you choose, you must have the necessary privileges in your logonid record to process resource rules.

After you complete this chapter, you will be able to use the ACF subcommands to:

- Create a resource rule
- Decompile a resource rule
- Display an existing resource rule
- Change a resource rule
- Store a resource rule
- Delete a resource rule (and verify its deletion)
- Test a resource rule before it is stored on the Infostorage database.

This section contains the following topics:

Beginning Resource Rule Processing

Before you can start processing resource rule sets with the ACF command, you must first tell CA ACF2 for VM that you want to process resource rule sets. To do this, enter:

```
SEt RESOURCE(type)
```

The type variable is the CA ACF2 for VM-supplied type code you must enter that tells CA ACF2 for VM what kind of resource rule set to process. For a list of these type codes, see the “About Resource Rules” chapter. For information on POSIX type codes, see the “OpenExtensions VM Support” chapter.

Using ACF COMPILE to Build Resource Rules

The COMPILE subcommand creates a set of resource rules. The syntax for the COMPILE subcommand is:

```
        { *   } [List|NOList ]
COmpile { fn  } [Store|NOStore]
                [Force|NOForce]
```

*****

Indicates that the subsequent text is input to the compiler. Using the COMPILE subcommand without parameters is equivalent to specifying an asterisk.

fn

Indicates a CMS filename where the rule compiler input resides. The filetype must be RULE. The filename cannot be one of the subcommand operands (LIST|NOLIST, STORE|NOSTORE, FORCE|NOFORCE). The ACF command cannot distinguish between a file named FORCE or an operand named FORCE. But you can have a rule with a key of FORCE. You must build the FORCE rule in a file with a filename of FORCE1, for example.

List|NOList

The LIST parameter displays the input to the compiler on the terminal screen when you compile the rule set. NOLIST causes no such display. LIST is the default when compiling from a CMS file. Otherwise, NOLIST is the default.

Store|NOStore

The STORE parameter automatically stores the rule set at compilation time. NOSTORE causes no automatic storing of the rule set, you must issue the STORE subcommand to store the rule set from an interactive compile. STORE is the default when compiling from a CMS file. Otherwise, NOSTORE is the default.

Force|NOForce

The FORCE parameter stores the resource rule set, regardless of whether it currently exists. NOFORCE only stores the resource rule set if it did not already exist. FORCE is the default. When you use FORCE|NOFORCE as parameters of the COMPILE subcommand, they apply only to the COMPILE subcommand you are issuing. When you use FORCE|NOFORCE as parameters of the SET subcommand, they are in effect until you change them or END the ACF command. They are not affected by changes in the ACF command setting.

CA ACF2 for VM provides two ways of compiling a resource rule set, from a CMS file or directly at the terminal. The following pages explain these two methods.

Compiling Resource Rules from a CMS File

You can create a resource rule set by first entering the control statements and rule entries in a CMS file. The file must have a filetype of RULE. Each rule entry must be on a separate line.

For example, assume that the following text was in a file named RSRCRULE with a filetype of RULE.

```
$KEY(ABCD) TYPE(123)
 UID(*****TLCAMS) ALLOW
 UID(*****TLCPJM) UNTIL(11/25/90) ALLOW
```

After entering the text into the CMS file, invoke CA ACF2 for VM and compile the resource rule set with the COMPILE subcommand under the RESOURCE(123) setting. Issue the COMPILE subcommand with the CMS filename.

```
set resource(123)
RESOURCE(123)

COMPILE RSRCRULE
ACFCMP510I ACF compiler entered

$KEY(ABCD) TYPE(123)
 UID(*****TLCAMS) ALLOW
 UID(*****TLCPJM) UNTIL(01/25/90) ALLOW
ACFCMP551I Total record length=184 bytes - 4 percent utilized
ACFCMD729I Resource ABCD stored

RESOURCE(123)
```

Indicate only the filename. The filetype is assumed to be RULE. The rule set is compiled and stored (by default). After compiling the resource rule set, you can use the TEST subcommand to determine if it performs attempted access validations as you expect.

Compiling Resource Rules from a Terminal

You can also enter a resource rule set directly from a VM CMS terminal by first entering the COMPILE subcommand without parameters under the RESOURCE(type) setting. Begin by entering the $KEY (with TYPE) control statement on the first line. Enter the other control statements, then the rule entries, each on a separate line. After entering each line of information, press **Enter**.

```
RESOURCE(123)

COMPILE
ACFCMP510I ACF compiler entered

$KEY(ABCD) TYPE(123)
 UID(*****TLCAMS) ALLOW
 UID(*****TLCPJM) UNTIL(11/25/99) ALLOW

ACFCMP551I Total record length=184 bytes - 4 percent utilized

RESOURCE(123)
```

When you are finished adding entries to the rule set, press **Enter** without adding any text. The COMPILE subcommand automatically ends. After compiling the resource rule set, you can use the TEST subcommand to ensure that the resource rule set performs the validation for the resource access you want. To add the rule set to the database, issue the STORE subcommand.

Using ACF DECOMP to List Resource Rules

The DECOMP subcommand, under the RESOURCE setting, decompiles a resource rule set that has been previously compiled and stored. Use this subcommand to examine a resource rule set. You can decompile a resource rule set at the terminal or into a CMS file to keep for later reference. The syntax for the DECOMP subcommand is:

```
        { *                }
DEComp  { rsrcname         }  [INTO(fn)]
        { LIKE(rsrcmask)   }
```

*****

Decompiles the last individual resource rule set processed since you established the current RESOURCE setting and type code.

rsrcname

Specifies the individual resource rule set to be decompiled or listed.

LIKE(rsrcmask)

Specifies a mask of resource names for decompiling or listing a group of resource rule sets.

INTO(fn)

Specifies a CMS file where the decompiled rule set is placed. You do not need to specify the filetype because a filetype of RULE is always assumed.

```
DECOMP ABCD INTO(RSRCRULE)
 ACFDCM557I Resource rule <ABCD> stored by TLCAMS on 11/11/97-19:01
 ACFDCM551I Total record length=184 bytes - 4 percent utilized

RESOURCE
```

In this example, the ABCD resource rule is decompiled into a CMS file RSRCRULE RULE.

Using ACF LIST to Display Resource Rules

The LIST subcommand functions like the DECOMP subcommand. The minimum CA ACF2 for VM privilege you must have is AUDIT. It lists all resource rules that match a rule ID mask. The syntax for the LIST subcommand is:

```
      { *              }
List  { (resource)     }
      { LIKE(rsrcmask) }
```

*****

Identifies the last resource rule you referenced in the ACF command session.

resource

Identifies a specific resource name ($KEY) to be listed.

LIKE(rsrcmask)

Lists multiple resource rules at one time. CA ACF2 for VM lists all resource rules that match the rsrcmask.

Using ACF DECOMP to Change Resource Rules

Use the DECOMP subcommand to modify an existing resource rule set. The syntax for this subcommand is:

```
       { *              }
DEComp { rsrc           } [INTO(fn)]
       { LIKE(rsrcmask) }
```

*****

Lets you decompile the last resource rule set you referenced since you established the current RESOURCE setting and type code.

rsrc

Specifies the individual resource rule set that you are decompiling.

LIKE(rsrcmask)

Specifies a mask of rule IDs for decompiling a group of resource rule sets.

INTO(fn)

Specifies the file into which you are decompiling the indicated resource rule set.

Using ACF STORE to Store Resource Rules

The STORE subcommand, under the RESOURCE setting, stores the previously compiled set of resource rules. CA ACF2 for VM can reject this operation if you do not have the authority to store the rule set. The syntax for the STORE subcommand is:

```
STore
```

The STORE subcommand has no operands. If you issued the SET NOFORCE command, CA ACF2 for VM stores the resource rule set only if it does not already exist. You receive a message if CA ACF2 for VM does not store the resource rule.

Using ACF DELETE to Remove Resource Rules

The DELETE subcommand, under the RESOURCE setting, deletes a resource rule set. Only users with the SECURITY privilege can delete resource rule sets. You can restrict this authority through scoping. The syntax for the DELETE subcommand is:

```
DELete { resource       }
       { LIKE(rsrcmask) }
```

The parameters for the DELETE subcommand are defined below.

resource

Specifies the resource rule you want to delete.

LIKE(rsrcmask)

Specifies a mask of resource names for a group of rule sets you want to delete.

The following example shows how the DELETE subcommand deletes the resource rule set for the resource named TERM1

```
DELETE TERM1
ACFCMD797I RESOURCE TERM1 DELETED

RESOURCE
```

Using ACF TEST to Test Resource Rules

The TEST subcommand interactively tests a compiled resource rule set. Test the resource rule to determine if the rule validates access to the resource that you want. While the TEST subcommand is active, only resource rule interpretation is done. This testing does not consider any site-specific system options or attributes of the logonids being tested. It also does not consider any exits that you have developed. The TEST subcommand is simplified because of the unlikelihood of testing all possible exit combinations. The syntax for the TEST subcommand is:

```
TEST  { * }
      { resource }
```

*****

Indicates that you want to test the previously compiled resource rule set.

(no parameter)

Operates the same as when you specify an asterisk.

resource

Identifies the resource rule that you want to test.

TEST Subcommand Keywords

After you issue the TEST subcommand and any of its parameters, the TEST subcommand is active. You can specify a test access environment by entering any of the following keywords with the appropriate values. You must separate each keyword by blank characters. You can specify keywords on one or more input lines.

DATE(date)

Specifies the access date. This date can be in the format mm/dd/yy, dd/mm/yy, or yy/mm/dd. The appropriate format is specified in the DATE field of the VMO OPTS record. The TEST subcommand uses today's date as the default.

LID(lid)

Specifies the logonid used to obtain the user ID. You must have access to the specified logonid record.

RSRCNAME(rsrcname)

Specifies the resource name that you want to test access for. CA ACF2 for VM places the $KEY value before the RSRC value unless you place the RSRC value in single quotes. CA ACF2 for VM does not use RSRCNAME when validating VM accesses. You can use this keyword to test OS/390 resource rule entries or accesses that site-written routines support.

SERVICE(READ,UPDATE,ADD,DELETE)

Specifies the type of access to test. This access type can be ADD, DELETE, READ, or UPDATE. Separate multiple access types by blank characters or commas. If you do not specify the SERVICE keyword, the TEST subcommand tests for combined ADD, DELETE, READ, and UPDATE authority.

SOURCE(sourceid)

Specifies the logical name of the input source or source group.

TIME(hhmm)

Specifies the time in hours (hh) and minutes (mm). This keyword is used to test a SHIFT specified in a rule entry.

UID(uidmask)

Specifies a mask of the UIDs you want to test and identifies the users whose access you want to test. You do not need access to the logonid records of the users whose access you want to test.

An example of how to use the TEST subcommand is shown below.

```
RESOURCE

COMPILE
ACFCMP510I ACF compiler entered

$KEY(ABCD) TYPE(123)
 UID(*****TLCAMS) ALLOW
 UID(*****TLCPJM) UNTIL(11/25/99) ALLOW

ACFCMP551I Total record length=184 bytes - 4 percent utilized
```

When the period (.) is displayed, the TEST subcommand is active. You can enter any of the TEST subcommand keywords to specify the particular environment that you want to test. The keyword UID, for example, tests whether a resource rule set allows a certain user to access a resource. In the next screen example, we are testing the previously compiled resource rule set for the resource name ABCD to see if user TLCAMS can access this resource.

```
RESOURCE

TEST *
.
UID(*****TLCAMS)
The following parameters are in effect:
 Date=11/11/97, time=1904, UID=*****TLCAMS, source=********

 Access would be ALLOWED
.

The following parameters are in effect:
 Date=11/11/97, time=1904, UID=*****TLCAMS, source=********

 Access would be ALLOWED
.
end

RESOURCE
```

The system displays all of the current values of the environment being tested. At the bottom of the display is a message indicating whether access to the resource is allowed, logged, or prevented. From the previously compiled rule set, user TLCAMS is allowed access to resource ABCD.

After a result is displayed, you can make another entry of keywords and values to specify another environment for testing. After you enter TEST command keywords, the values you specify remain in effect until you change them. Furthermore, as shown in the previous example, CA ACF2 for VM assumes that nearly all values we did not specify are completely masked by default. For instance, if you specify no UID keyword, the subcommand tests whether all UIDs are allowed access. To terminate the TEST subcommand, enter:

```
END
```

TEST Subcommand Results

The results of the TEST subcommand show whether access to the resource is allowed, logged, or prevented, as follows:

ALLOWED

Access is allowed

LOGGED

Access is allowed, but logged

PREVENTED

Access is prevented.

If no rule entry specifically applies to the test access environment, CA ACF2 for VM displays the following message:

```
ACFpgm74CI No rule applies, access would be denied
```

Chapter 16: Protecting Special Resources

In CA ACF2 for VM, there are eight CA ACF2 for VM-supplied type codes. These type codes are classified in storage class “R” for special kinds of resource rules. These resources and their default type codes are:

VM account support (ACT default)

Specifies that if a VM account number is assigned, you can change it with the SET ACCOUNT command whenever you issue the LOGON or AUTOLOG commands. Resource rules and VMACCT logonid values provide account number validation.

AUTOLOG resource rules (ALG default)

Validate AUTOLOG commands automatically. This is necessary because CA ACF2 for VM lets you execute the AUTOLOG command without a password in predefined instances. This default also applies to the XAUTOLOG command.

Group machine logon resource rules (GRP default)

Validate group logon resource rules when you log onto a group virtual machine with the GRPLOGON privilege. This ensures that only authorized individuals who are specifically defined through resource rules can log onto group machines.

DIAL resource rules (DIA default)

Validate DIAL resource rules automatically unless the target user ID has the DIALBYP privilege in his logonid record. This ensures that only authorized individuals who are specifically defined through resource rules can dial into machines that have been secured for DIAL validation.

IUCV resource rules (IUC default)

Provide a fine degree of audit and control in establishing and terminating Inter User Communication Vehicle (IUCV) paths for transferring data.

APPC/VM resource rules (IUC default)

Provide a fine degree of audit and control in establishing and terminating Advanced Program-to-Program Communication/VM paths for transferring data.

POSIX resource rules (PGR default)

Specifies the groups, both primary and supplemental, that each system user can use.

ESA dataspace resource rules (DSP default)

Validates access to ESA dataspaces at permit time. This ensures that only authorized individuals who are specifically defined through resource rules can access dataspaces.

VMCF resource rules (VMC default)

Provide a fine degree of audit and control in establishing and terminating Virtual Machine Communication Facility (VMCF) paths for transferring data.

This chapter describes the types of resource rules specific to CA ACF2 for VM and how they work with other CA ACF2 for VM controls.

This section contains the following topics:

Account Support through CA ACF2 for VM

With CA ACF2 for VM, you can assign a virtual machine to an account number whenever you issue the LOGON or AUTOLOG commands. You can change the account number when you issue the SET ACCOUNT command. CA ACF2 for VM account validation is integrated into standard CA ACF2 for VM system entry processing and also assumes control whenever you issue the SET ACCOUNT command. This coincides with account support under native VM that occurs during the same instances.

The following terms apply to native VM account support:

Account control statement

An optional entry in the CP directory for a virtual machine that defines primary and alternate account numbers where costs can be charged for the machine.

Primary account number

A one- to eight-character account number in a virtual machine CP directory account statement. During system entry, CA ACF2 for VM automatically assigns this number to the machine by default.

Alternate account numbers

An optional list (up to seven account numbers, one-to eight- characters long) in the virtual machine CP directory account statement. CA ACF2 for VM assigns the machine an alternate account number only if it is explicitly specified during system entry (through the ACCOUNT operand) or if it is changed during the machine's session (through the SET ACCOUNT command).

The CA ACF2 for VM method for employing account validation is significantly different from that of native VM. Specifically, CA ACF2 for VM account validation takes place through resource rules and VMACCT logonid values that replace the CP directory ACCOUNT statements. With CA ACF2 for VM account support, the following features are available:

- You can use three validation modes for implementing account support.
- You can extend the number of accounts available to a single user ID beyond the limit of eight native VM imposes.
- You can assign administrative authority over specific accounts to specific users.

Account Controls

CA ACF2 for VM provides the following controls for account support:

Account resource rules

CA ACF2 for VM definitions that list account numbers and the virtual machines that can use them to charge costs. These rules work with VMACCT logonid values to functionally replace the native VM implementation ACCOUNT directory control statements provide.

ACCTVLD

An operand in the OPTS VMO record that establishes the account mode setting. You can specify one of three modes of operation: NO, FULL, and LID.

ACCOUNT

An operand in the RESCLASS VMO record that defines the three-character resource type code for account resource rule sets. ACT is the default.

VMACCT

An eight-byte logonid field that holds the default account number for a virtual machine. The VMACCT values work with account resource rules to functionally replace VM account directory control statements.

VLDVMACT

A logonid bit field you can turn on for selected users so they undergo CA ACF2 for VM account validation under the LID account mode setting.

Implementing Account Controls

During implementation, set SET ACCTVLD to NO, turning off account validation. Do not turn on full account security (FULL setting) or partial security (LID setting) until you have compiled and stored your account resource rule sets.

There are five steps to implementing VM account support:

1. Determine your account mode setting. Issue the SHOW STATE subcommand to ensure that your ACCTVLD setting is set to NO. The ACCTVLD operand of the OPTS VMO record establishes the account mode. There are three possible settings: FULL, LID, and NO.

   ACCTVLD(FULL)

   Indicates that full CA ACF2 for VM account support is in effect. (This is the default). CA ACF2 for VM does not use the CP directory for accounting purposes except at system initialization time. We recommend this setting for CA ACF2 for VM account validation.

   ACCTVLD(LID)

   Indicates an alternative to full account support. CA ACF2 for VM account support is in effect, but only for specified machines. The CP directory is not used for accounting except at system initialization time.

   ACCTVLD(NO)

   Indicates that CA ACF2 for VM account support is disabled. Directory account validations are performed exactly as on a native VM system.

2. Assign the VLDVMACT logonid privilege. This is necessary only if you use the ACCTVLD=LID setting. If you use full account security (ACCTVLD=FULL), skip to Step 3, Assign the VMACCT Logonid Privilege.

   Under the ACCTVLD=(LID) account mode setting, the VLDVMACT logonid bit field defines virtual machines for CA ACF2 for VM account validation. Machines without VLDVMACT do not undergo account resource rule validation. These machines are automatically assigned the account numbers kept in their VMACCT logonid fields. If they do not have VMACCT values, they are denied system entry.

   To turn on the VLDVMACT logonid attribute for a virtual machine, a user with the appropriate privilege must enter the CHANGE subcommand under the ACF LID mode setting.

3. Assign the VMACCT logonid privilege. VMACCT defines an eight-byte logonid field that is the default account number for a virtual machine. Virtual machines to undergo CA ACF2 for VM account validation need this privilege if they are to be assigned default accounts. You can turn on the VMACCT logonid attribute for a virtual machine in one of two ways:

   - Execute the ACFCVACT utility that prepares a file of ACF CHANGE subcommands for creating the VMACCT field values for logonids in the directory. This utility also creates account resource rules from the CP directory.
   - A user with the appropriate privilege can issue the CHANGE subcommand under the ACF LID mode setting. With this subcommand, you can assign this attribute to users on an individual basis.

4. Establish account resource rule sets. Remember, when full account validation is in effect, CA ACF2 for VM automatically validates an account resource rule during system entry or when you issue the SET ACCOUNT command. The CA ACF2 for VM access mode setting and CA ACF2 for VM SECURITY privilege have no bearing on the account resource rule validation process.

   A separate rule is required for each account, but you can use standard CA ACF2 for VM masking conventions. In every rule, the account number is specified as the $KEY value. The UID portion of a rule entry is the machine to use that account number. You can create account resource rules in two ways:

   - Execute the ACFCVACT utility that converts your account CP directory statements into resource rules. This utility also prepares a file of ACF CHANGE subcommands for creating the VMACCT field values for logonids in the directory.
   - Issue the COMPILE subcommand under the RESOURCE setting. With this subcommand, you can store individual resource rules or many rules from a file.

5. Set the account mode (as determined earlier). Set the ACCTVLD operand in the OPTS VMO record to FULL or LID. Issue the SHOW STATE subcommand of the ACF command to check that your ACCTVLD setting is correct.

For more detailed information on the ACFCVACT utility, see the in the *Reports and Utilities Guid*e.

Account Resource Rules and VMACCT Field Values

When you use CA ACF2 for VM account resource rules, you can use the account operand and SET ACCOUNT command to dynamically impact account validation. For example:

ACCOUNT operand

An optional operand of the LOGON and AUTOLOG CP commands. You can issue it during system entry to assign a virtual machine an account number (up to eight characters) that is not its default value. This alternative account number is temporary. CA ACF2 for VM recognizes it only for the duration of the session, or until you issue the SET ACCOUNT command. CA ACF2 for VM checks the account number against an account resource rule that must allow the assignment for system entry.

SET ACCOUNT command

A CP command that changes the account number (up to eight characters) of a virtual machine. This new account number is temporary. CA ACF2 for VM recognizes it only for the duration of the session (unless you issue the SET ACCOUNT command again with another account number). CA ACF2 for VM checks the account number against an account resource rule that allows the change.

Account resource rule validation is entirely optional with CA ACF2 for VM. It is enforced with an account mode setting of FULL (validation for all virtual machines) or LID (validation only for designated virtual machines). The
CA ACF2 for VM access mode setting and CA ACF2 for VM SECURITY privilege level have no bearing on the account resource rule validation process. An account rule validation occurs in the following way:

- The account number assigned or changed is validated as a rule set $KEY value.
- The user ID of the virtual machine whose account number is assigned or changed is validated as a UID value of an entry in the rule set.
- The rule set typecode is ACT by default. You can modify this value through the TYPES field of the RESTYPE VMO record.

Whenever a logging or invalid access occurs from account resource rule validation, CA ACF2 for VM records it in the Resource Event Log (ACFRPTRV).

Account Number to Undergo Validation

Resource rule account validation takes place under the appropriate settings (FULL or LID). Specifically, if you execute:

- The LOGON command (issued at system entry)
- The AUTOLOG command (issued at system entry)
- The SET ACCOUNT command (issued during a session).

The account number that is validated as the rule set $KEY value depends on the scenario. There are three possibilities:

Scenario 1

During system entry, CA ACF2 for VM automatically selects the user default account number (in his VMACCT logonid field) to undergo account resource rule validation. This occurs by default (whenever you issue the LOGON or AUTOLOG commands without the ACCOUNT operand).

Scenario 2

During system entry, you specify an account number other than the default value to undergo account resource rule validation. Use the ACCOUNT operand of the LOGON or AUTOLOG commands.

Scenario 3

During the machine's session, you specify any account number to undergo account resource rule validation. Use the SET ACCOUNT command for this purpose.

For example, assume that the TLCAMS virtual machine has the value ACT001 in its VMACCT logonid field. During system entry for TLCAMS, CA ACF2 for VM selects this number by default (Scenario 1). It then validates this account number against an ACCOUNT resource rule with the same $KEY value. For the validation to succeed, a rule set similar to the following is needed:

```
$KEY(ACT001) TYPE(ACT)
 UID(TLCAMS) ALLOW
```

Assume that another rule set lets TLCAMS issue another account number, ACT002, for example:

```
$KEY(ACT002) TYPE(ACT)
 UID(TLCAMS) ALLOW
```

During system entry, this number is assigned to the TLCAMS virtual machine if you specify it through the ACCOUNT operand of the LOGON or AUTOLOG command (Scenario 2). Or, during the session of TLCAMS, the user could change this number through the SET ACCOUNT command (Scenario 3). In both of these situations, the account number ACT002 is temporary. CA ACF2 for VM recognizes it as TLCAMS's account for the duration of the session.

ACCOUNT Resource Rules and VMACCT Values; Replacing ACCOUNT Directory Validation

ACCOUNT resource rules and the VMACCT logonid field values replace the ACCOUNT CP directory statements in native VM.

For example, the following two user definitions are in the CP directory:

```
USER TLCAMS TLCAMS 2M 8M G
ACCOUNT ACT001

USER AUTOLOG1 AUTOLOG1 2M 8M G
ACCOUNT ACT001
```

In native VM, the first ACCOUNT statement assigns the account ACT001 to the TLCAMS virtual machine. The second ACCOUNT statement assigns the same account to the AUTOLOG1 machine. In CA ACF2 for VM, the ACCOUNT directory statements are ignored. Instead, CA ACF2 for VM enforces ACCOUNT resource validation. For account validation to occur in the manner the above directory defines, both TLCAMS and AUTOLOG1 need the VMACCT field value defined in their logonid records.

The following resource rule is also needed:

```
$KEY(ACT001) TYPE(ACT)
 UID(TLCAMS) ALLOW
 UID(AUTOLOG1) ALLOW
```

The benefits ACCOUNT resource rules provide are twofold. Rules let you specify more than eight accounts for a given user; the ACCOUNT statement does not. You can assign administrative authority over specific accounts to specific users through the %CHANGE control statement in individual resource rule sets. CA ACF2 for VM account support also provides three account mode settings, as explained in the next section.

We provide a REXX exec called ACFCVACT to help you create ACCOUNT resource rules from the CP directory. This exec also prepares a file of ACF CHANGE subcommands for creating the VMACCT field values for logonids in the directory. These values are the default account numbers for virtual machines defined for account validation. You can run the ACFCVACT EXEC during CA ACF2 for VM installation for VM or whenever you implement account validation.

Important Information

You should keep in mind the following information about VM account support through CA ACF2 for VM when creating account resource rules:

- The ACCOUNT field of the RESCLASS VMO record defines the type code required for account resource rule validation. The default specification is ACT, implementing the typecode ACT. You can modify this value.
- By default, you can implement account number masking for ACCOUNT resource rules. Masking lets you use asterisks (*) as $KEY masking characters, but not the dash (-).

  The ACCOUNT field of the RESTYPE VMO record lets you mask the account numbers in the $KEY. This macro must contain the same value specified in the ACT operand of the TYPES field of the RESTYPE VMO record for masking to occur. The default TYPES field of the RESTYPE VMO record includes type code ACT.
- Specifying a resource rule's type code in the RESTYPE VMO record also makes the resource rule directory resident for the rule set's type code. This improves system performance.

- CA ACF2 for VM ignores the SERVICE and VERIFY keywords in ACCOUNT resource rules.
- Normal source and shift controls apply to the validation process for ACCOUNT resource rules.

  During system IPL, the system operator and any user ID specified in the AUTOLOG operand list of the VMXAOPTS macro (in HCPAC0) bypass CA ACF2 for VM account validation. However, when the ACF2 service machine startup completes, CA ACF2 for VM validates and initializes the account numbers of all users already on the system. Any user accessing the system with an unauthorized or invalid account number is automatically forced off.

  Any user ID with the VMD4TARG logonid privilege bypasses account validation when logging on.

  With CA ACF2 for VM, you can command limit the LOGON command. Preventing or logging the ACCOUNT operand of the LOGON command depends on the CA ACF2 for VM account mode setting:

- If account support is turned on (ACCTVLD=(FULL) or ACCTVLD=(LID)) in the OPTS VMO record, you cannot prevent or log the ACCOUNT operand of the LOGON command through command limiting.
- If account support is turned off (ACCTVLD=(NO)), you prevent or log the ACCOUNT operand of the LOGON command through command limiting.

VM Native AUTOLOG Support

Operands for entering the AUTOLOG and XAUTOLOG commands on native VM systems are also available with CA ACF2 for VM installed:

Without Password Suppression:

```
AUTOLOG target password {operands}
XAUTOLOG target PASSWORD password {operands}
```

With Password Suppression:

```
AUTOLOG target {operands}
XAUTOLOG target {operands}
```

Although the same AUTOLOG or XAUTOLOG operands are available on native VM and on systems with CA ACF2 for VM installed, these systems differ in who can issue the AUTOLOG or XAUTOLOG operands.

For AUTOLOG

In native VM, only CP privilege class A and B users can issue the command. This is not changed with CA ACF2 for VM.

For XAUTOLOG

In native VM only CP privilege class A and B users can enter passwords with the PASSWORD and PROMPT operands, whereas CA ACF2 for VM lets class A, B, and G users enter passwords with these operands. By allowing class G users to enter passwords for validation on CA ACF2 for VM systems, you benefit over native VM.

AUTOLOG or XAUTOLOG Validation

The following terms describe CA ACF2 for VM AUTOLOG support:

Initiator machine

Indicates the virtual machine (or user of that machine) that issues the AUTOLOG or XAUTOLOG command.

Target machine

Indicates the virtual machine autologged through the AUTOLOG or XAUTOLOG command.

Password suppression

Indicates that when you must supply a password with AUTOLOG, XAUTOLOG, or LOGON, this option (by default) requires that you enter the password following a prompt. CA ACF2 for VM does not display the password when you enter it; it automatically suppresses it. If you enter a clear text password on the same line as the XAUTOLOG command, CA ACF2 for VM rejects the command if password suppression is in effect. CA ACF2 for VM cannot recognize a password entered on the same line as the AUTOLOG command.

AUTOLOG and XAUTOLOG CP Directory Statement

Indicates the method native VM uses to authorize class G virtual machines to autolog other virtual machines.

The special CA ACF2 for VM logonid attributes for the AUTOLOG or XAUTOLOG command include:

AUTOALL

Indicates that a virtual machine with this privilege can autolog any virtual machine without specifying a password, no matter what privileges the machines might have. Like AUTONOPW, AUTOALL has no effect on the logon process. The user of a machine with this privilege can log on through normal logon procedures.

AUTONOPW

Indicates that a virtual machine with this privilege can be autologged without specifying a password. AUTONOPW has no effect on the logon process. The user of a machine with this privilege can log on through normal logon procedures.

AUTOONLY

Indicates that a virtual machine with this attribute cannot be logged onto a device, even if the machine is a group virtual machine.

The process CA ACF2 for VM uses on VM systems to validate the AUTOLOG and XAUTOLOG command consists of the following steps:

1. AUTOLOG or XAUTOLOG command limiting validation
2. Conditional password validation
3. Autolog resource validation
4. Propagating user IDs for group logon.
5. A brief explanation of each step follows.

Step 1: AUTOLOG or XAUTOLOG Command Limiting Validation

Whenever you issue the AUTOLOG or XAUTOLOG command, CA ACF2 for VM validates the command limiting rules. This is optional for AUTOLOG or XAUTOLOG, just as it is for any other CP command. Limiting the AUTOLOG or XAUTOLOG commands lets you:

- Prevent specific users from executing the AUTOLOG or XAUTOLOG commands up-front, avoiding the overhead of going through the other validation steps.
- Control the AUTOLOG or XAUTOLOG command operands that are allowed, logged, or prevented for individual users.

CA ACF2 for VM records all loggings or invalid accesses from limiting the AUTOLOG or XAUTOLOG command in the Command Limiting Journal (ACFRPTCL).

Step 2: Conditional Password Validation

After optional command limiting validation, CA ACF2 for VM validates conditional passwords for VM systems.

For XAUTOLOG

CA ACF2 for VM for checks if you entered the PROMPT or PASSWORD operands of the XAUTOLOG command. These operands are available to CP privilege class A, B, or G users on VM systems CA ACF2 for VM protects. They have the following definitions:

PROMPT

Prompts for the password of the target virtual machine being autologged. CA ACF2 for VM issues a message, asking you for your password.

PASSWORD password

Requires the password (in clear text form on the command line) of the target virtual machine being autologged. CA ACF2 for VM rejects the PASSWORD option if password suppression is active. CA ACF2 for VM issues an error message telling you the XAUTOLOG command was rejected because the command format is not valid. CA ACF2 for VM does not log this invalid access attempt. It is regarded as a syntax error. Issue the XAUTOLOG command again, but use the PROMPT operand instead.

If you specify PROMPT or PASSWORD, CA ACF2 for VM validates the password, resulting in:

- If you supplied the correct password, the validation continues to Step 3, Resource Rule Validation.
- If you supplied an incorrect password, CA ACF2 for VM denies the AUTOLOG, displays a message telling you that the password was incorrect, and records this in the Invalid Password/Authority Log (ACFRPTPW).

If you do not specify the PROMPT or PASSWORD options in the XAUTOLOG command, CA ACF2 for VM determines if you must supply a password:

- If the logonid of the target machine has the AUTONOPW privilege or if the logonid of the initiator machine has the AUTOALL privilege, you do not have to enter a password. The system entry process continues to Step 3, Resource Rule Validation.
- However, if the AUTONOPW and AUTOALL conditions are not met, CA ACF2 for VM issues a syntax error message to tell you that the XAUTOLOG failed and you must enter a password. CA ACF2 for VM does not log this invalid access attempt. It is treated as a syntax error. Reenter the XAUTOLOG command with the PROMPT or PASSWORD option.

For AUTOLOG

When CA ACF2 for VM requires a password, you must enter the password following a prompt. You can never enter the password on the command line. If you enter a password on the same line as the AUTOLOG command, CA ACF2 for VM considers it console input data. Consider this example:

```
AUTOLOG target [console input data]
```

If you enter a password on the command line, it appears in clear text and passes to the target machine as a console command. When password suppression is not in effect, control returns to VM native and depends on the SET PASSWORD command. The SET command specifies that

- You cannot enter the password on the command line. (This is the default mode when the system is IPLed.) Use the SET and AUTOLOG commands, SET PASSWORD AUTOLOG SEPARATE.

  Enter:

  ```
  AUTOLOG target [console input data]
  ```

- You can enter the password on the command line. Use the SET and AUTOLOG commands, SET PASSWORD AUTOLOG INCLUDE.

  Enter:

  ```
  AUTOLOG target [password][console input data].
  ```

Step 3: Resource Rule Validation

CA ACF2 for VM next validates AUTOLOG resource rules, regardless of the CA ACF2 for VM access mode setting and any special privileges you might have (for example, SECURITY). When checking AUTOLOG resource rules,
CA ACF2 for VM:

- Validates the user ID of the target machine (the machine being autologged) as a rule set $KEY value
- Validates the user ID of the initiator machine (the machine performing the autolog) as a UID value of an entry in the rule set
- Validates that the rule set typecode is ALG, by default. You can modify this value through the TYPES field of the RESTYPE VMO record.

For example, the following rule set lets OPERATOR, MAINT, or SYSMGR (the initiators) autolog TLCAMS (the target):

```
$KEY(TLCAMS) TYPE(ALG)
 UID(OPERATOR) ALLOW
 UID(MAINT) ALLOW
 UID(SYSMGR) ALLOW
```

During validation of the AUTOLOG resource rule, CA ACF2 for VM also checks the logonid of the initiator machine (the machine performing the autolog) for the AUTOALL privilege. This is a super AUTOLOG privilege similar to NON-CNCL for data access. If AUTOALL is specified, the autolog occurs, even if there is no rule explicitly allowing the request. CA ACF2 for VM records this override, and all other loggings and invalid accesses, in the Resource Event Log (ACFRPTRV).

Autolog Resource Rules; Replacing XAUTOLOG Directory Validation

AUTOLOG resource rules replace the AUTOLOG or XAUTOLOG CP directory statements in native VM. For example, the following two user definitions are in the CP directory:

```
USER TLCAMS TLCAMS 2M 8M G
AUTOLOG RON BOB TIM

USER AUTOLOG1 AUTOLOG1 2M 8M G
XAUTOLOG JANE ANN MARY
```

In native VM, the first AUTOLOG statement lets Class G users (RON, BOB, and TIM) autolog the target virtual machine (TLCAMS) using XAUTOLOG. The second statement lets class G users (JANE, ANN, and MARY) autolog AUTOLOG1 using XAUTOLOG. These directory statements apply only to the XAUTOLOG command.

CA ACF2 for VM ignores AUTOLOG or XAUTOLOG directory statements. Instead, autolog resource validation is enforced. To autolog as the above directory defines, you need the following resource rules:

```
$KEY(TLCAMS) TYPE(ALG)
 UID(RON) ALLOW
 UID(BOB) ALLOW
 UID(TIM) ALLOW

$KEY(AUTOLOG1) TYPE(ALG)
 UID(JANE) ALLOW
 UID(ANN) ALLOW
 UID(MARY) ALLOW
```

We provide a REXX EXEC called ACFCVALG to help you create AUTOLOG resource rules from the CP directory. You can run this exec while installing CA ACF2 for VM on VM after a system IPL.

Step 4: Propagating User IDs for Group Logon

By this step, the user who issued the AUTOLOG or XAUTOLOG command already has authorization to autolog the target machine. This step preserves the individual accountability of group machines (machines with the GRPLOGON privilege) for audit purposes.

CA ACF2 for VM checks the logonid of the target machine for the GRPLOGON privilege. If GRPLOGON is specified, the name of the initiator machine becomes associated with the name of the target machine. For example, the initiator appears in all CA ACF2 for VM reports as the logonid. The target appears as the jobname. This corresponds with how CA ACF2 for VM generates reports for group virtual machines.

For any other machines that this group machine subsequently autologs, the name of the original initiator (the machine that first autologged the group machine) is still identified as the logonid in the CA ACF2 for VM reports. The original initiator of the group machine is accountable for all of that machine's subsequent actions.

An autologged target group machine goes through autolog resource validation, but not group logon resource validation. Group logon resource validation applies only when group machines are trying to gain access to the system through logon.

Important AUTOLOG and XAUTOLOG Information

Read the following important information about VM AUTOLOG or XAUTOLOG support through CA ACF2 for VM:

- Standard CA ACF2 for VM command limiting rules apply for AUTOLOG and XAUTOLOG. They include:
  - The LIMIT field of the CMDLIM VMO record that specifies the commands CA ACF2 for VM validates.
  - Command limiting rules that allow, log, or prevent command access attempts.
  - The MODE field of the CMDLIM VMO record that specifies the mode for command limiting validation.
- Information for creating AUTOLOG resource rules includes:
  - The AUTOLOG field of the RESCLASS VMO record defines the typecode required for autolog resource rule validation. The default specification is AUTOLOG (ALG), implementing the typecode ALG and implies AUTOLOG or XAUTOLOG commands. You can modify this value.
  - By default, you can implement resource name masking for AUTOLOG resource rules. Masking lets you use asterisks (*) as $KEY masking characters, but not the dash (-). The TYPES field of the RESTYPE VMO record lets you mask the target machines in the $KEY.
  - Specifying a resource rule's type code in the TYPES field of the RESTYPE VMO record makes the resource rule directory resident for the rule set's type code. This helps improve system performance.
  - CA ACF2 for VM ignores the SERVICE and VERIFY keywords in AUTOLOG resource rules.
  - Normal source and shift controls apply to the validation process for AUTOLOG resource rules.

- Password suppression is a CA ACF2 for VM option with the AUTOLOG and XAUTOLOG commands on VM systems. The PSWDSUP operand of the VMXAOPTS macro in HCPAC0 defines password suppression. When CA ACF2 for VM requires a password during AUTOLOG and XAUTOLOG validation, there are two possible reasons for entering your password, as the PSWDSUP option dictates:
  - PSWDSUP=YES specifies that password suppression is turned on. This is the default. CA ACF2 for VM rejects any password you enter in clear text with the PASSWORD operand. You cannot enter the password on the command line for AUTOLOG. You must wait for the prompt.

    For XAUTOLOG, you must specify the XAUTOLOG PROMPT option so you can enter the password following the prompt. CA ACF2 for VM automatically suppresses the password display.
  - PSWDSUP=NO specifies password suppression is turned off. You can enter the password in clear text with the XAUTOLOG PASSWORD option only if you issued the following class B command for the system since the last IPL.
- CA ACF2 for VM validates data and resource access for a machine that is successfully autologged (the target) against it and not the user who issued the autolog (the initiator).
- CA ACF2 for VM validates data and resource access for a group machine that is successfully autologged (the target) not the user who issued the autolog (the initiator). The initiator is identifiable in the reports corresponding with how CA ACF2 for VM generates reports for group virtual machines.

AUTOLOG or XAUTOLOG Implementation

Three steps are required to implement VM AUTOLOG support:

1. Establish AUTOLOG and XAUTOLOG command limiting controls, if necessary. Decide on whether to implement AUTOLOG and XAUTOLOG command limiting to restrict unauthorized attempts to execute this command. XAUTOLOG is a CP privilege class A, B, and G command. With CA ACF2 for VM, the command syntax for XAUTOLOG is identical for class A, B, and G users. For complete information on how to use command limiting, to the *Command and Diagnose Limiting Guide*.

2. Assign the appropriate logonid privileges for AUTOLOG and XAUTOLOG execution. Determine when you can execute the AUTOLOG or XAUTOLOG commands without a password. Assign the AUTONOPW privilege to the appropriate target machines and the AUTOALL privilege to the appropriate initiator machines. Also, decide which target virtual machines should have the AUTOONLY privilege so that nobody can log onto them (they can only be autologged). Assign this attribute appropriately.

   If you are installing CA ACF2 for VM for the first time, assign the AUTOALL privilege to the AUTOLOG1 virtual machine. This lets it autolog subsequent machines after coming up, generating SMF logging records in the process. You can write AUTOLOG resource rules to control the precise nature of how you want to autolog this machine.

   You can also assign the AUTOALL privilege to include the IBM VMBATCH machine and the System Center VMSCHEDULE machine.

   To turn on the AUTONOPW, AUTOALL, or AUTOONLY privilege bits for a virtual machine, a user with the appropriate privilege must enter the ACF CHANGE subcommand.

3. Establish AUTOLOG resource rule sets. Every time someone tries to autolog a virtual machine, CA ACF2 for VM automatically validates an AUTOLOG resource rule. The CA ACF2 for VM access mode setting and CA ACF2 for VM SECURITY privilege have no bearing on the AUTOLOG resource rule validation process.

   You need a separate rule for each machine that is autologged. In each rule, you must specify the logonid of the target machine as the $KEY value. The UID portion of the rule entry is the initiator machine.

   You can execute the ACFCVALG utility that converts your AUTOLOG CP directory statements into resource rules to create AUTOLOG resource rules. For complete details on this utility, see the *Reports and Utilities Guid*e. You can also issue the COMPILE subcommand of the RESOURCE setting to store individual resource rules or many rules from a file.

The GRPLOGON Privilege: Logging onto Group Machines

It is sometimes necessary to have a virtual machine that several users can access. Suppose that several database administrators from different departments maintain a centralized database. They could perform the necessary maintenance from one virtual machine. To access the machine, all of the administrators enter the same logonid and password of the machine itself. Only one person has access to the machine at a time. Although using a virtual machine like this serves a valuable purpose, it poses a possible security breach. From an auditing standpoint, there is no way to determine who is using the machine (other than someone logged on). Individual accountability is lost.

CA ACF2 for VM defines a virtual machine to have the special GRPLOGON privilege to solve this problem. This is a bit field in the Privileges group of the logonid record. A machine with this privilege is called a group virtual machine. Many people can use such a machine, with one person having access at any given time. They all can be identified through auditing.

This group logon feature minimizes password proliferation. To log onto a group machine, you do not need to know other passwords because the users logging onto the group machine use their own passwords. You can log onto a group virtual machine (a virtual machine with the GRPLOGON privilege) by logging on as a group user. You can also autolog a group virtual machine like any other virtual machine.

The following terms describe CA ACF2 for VM group logon support:

Group user

Indicates an individual who uses a group virtual machine.

Group machine

Indicates a virtual machine defined with the GRPLOGON privilege in the Logonid database. Many users can use it and still be identifiable to the system.

Logging on as a Group User

This is the common implementation when using a group virtual machine.

```
LOGON maint
ACFpgm263R Enter your CA-ACF2 logonid
userlgn
ACFpgm244R Enter CA-ACF2 password
userpwd
```

After entering LOGON MAINT to use the MAINT group machine, the group user supplies his own logonid and password, not MAINT's. After he accesses the system, CA ACF2 for VM validates all accesses as if the group virtual machine were logged on as a regular user. That is, it validates normal data and resource accesses against the group machine, not the group user. CA ACF2 for VM creates SMF records for the group machine. SMF records contain information that identifies the group user. The user who logged onto the group virtual machine is identified in the CA ACF2 for VM reports. This enforces individual accountability.

System Access through Resource Rules

When you try to log onto a group virtual machine, CA ACF2 for VM automatically validates a resource rule to see if you can use the machine, regardless of the CA ACF2 for VM data access mode setting and any special privileges you might have. CA ACF2 for VM:

- Validates the user ID of the group logon machine (the machine with the GRPLOGON privilege) as a rule set $KEY value. You can specify the resource rule type code in the TYPES field of the RESTYPE VMO record to mask this value.
- Validates the user ID that is entered during the logon process after the following prompt as a UID value of an entry in the rule set:

  ```
  ACFpgm263R Enter your ACF2 Logonid
  ```

  Standard UID masking applies for this value.

  Validates that the rule set type code is GRP by default. This type code is defined in the GRPLOGON field of the RESTYPE VMO record.

For example, the following rule set lets any group user with the logonid mask TLC- log onto the MAINT group machine.

```
SET RES(GRP)

RESOURCE

COMPILE
ACFCMP510I ACF compiler entered

$KEY(MAINT) TYPE(GRP)
 UID(*****TLC-) ALLOW
 UID(*****TLCAMS) PREVENT

ACFCMP551I Total record length=168 bytes - 4 percent utilized

RESOURCE
```

The PREVENT rule entry makes it explicitly clear how the MAINT group machine is used in this example. Users with the logonid TLC- mask can supply their own logonids and passwords to log onto the machine. TLCAMS cannot log on as the group machine. Because each user must supply his own logonid and password, no password sharing takes place. Whenever an invalid access attempt occurs through this kind of resource rule validation, CA ACF2 for VM records it in the Resource Event Log (ACFRPTRV).

Important Group Logon Information

Important details apply to the system entry process for logging onto group virtual machines.

- You must assign a group virtual machine the GRPLOGON privilege.
- You cannot log onto a machine with the AUTOONLY privilege, regardless if that machine is a group machine. CA ACF2 for VM issues the following message when you try to log onto a machine with the AUTOONLY privilege:

  ```
  ACFpgm279E Logon not allowed - Logonid has AUTOONLY attribute
  ```

  CA ACF2 for VM records all unsuccessful system access attempts in the Invalid Password/Authority Log (ACFRPTPW).

  The GRPLOGON field of the RESCLASS VMO record defines the typecode required for resource rule validation for group machines. The default specification is GRPLOGON=(GRP), implementing type code GRP.
- You can specify the resource rule type code in the RESTYPE VMO record to implement $KEY resource name masking for group logon resource rules. This type code must be the same value specified in the GRPLOGON field of the RESCLASS VMO record. You can use standard UID masking in individual rule entries.
- CA ACF2 for VM ignores the SERVICE and VERIFY keywords for group logon resource rules.
- Normal source and shift controls apply to the validation process for resource rule sets for group virtual machines.
- Password suppression is an IBM security feature implemented in the PSUPRS operand of the SYSJRL macro. It forces you to enter your password on a separate line from your logonid when you log on. CA ACF2 for VM does not display your password when you enter it. If you attempt to enter a password for a logonid with the group logon attribute, you receive error messages, but can proceed.

  ```
  LOGON MAINT MAINTPW
  ACFpgm278E Logon for groupid - password entered will be ignored
  ACFpgm263R Enter your CA-ACF2 logonid
  ```
- When you are logging onto a group machine as a group user, you do not need to be defined as a user ID in the VM directory. The only prerequisite for a group user is that the user is defined with a logonid record in the Logonid database. You must always define the group machine with a logonid record in the Logonid database and a user ID in the VM directory. This applies not only when you are logging on as a group user, but also when you are logging on as the group machine itself.

Protecting the DIAL Command

The DIAL command lets you dial into another virtual machine. Typically, you can dial into a production or test OS/390 or VSE machine and then log onto an application, such as TSO, CICS, IMS, or ICCF. You can control and audit the DIAL command to protect virtual machines from unauthorized users who dial into the machines and potentially access their resources.

The following terms describe CA ACF2 for VM DIAL support:

DIAL secured machines

By default, all machines are dial secured. CA ACF2 for VM validates logonid, password, and resource accesses unless the machine has the DIALBYP logonid attribute.

Dialer

Indicates the user requesting DIAL access to a target virtual machine.

Target Virtual Machine

Indicates the virtual machine a dialer can access.

CA ACF2 for VM DIAL Validation Process

Because the CP DIAL command lets you dial into another virtual machine with no restrictions, CA ACF2 for VM provides carefully controlled and audited DIAL command access validation to protect secured virtual machines from unauthorized users. With CA ACF2 for VM, all machines are secured from DIAL command access by default. Use the DIALBYP logonid privilege at your discretion. This privilege indicates bypass DIAL command validation. For example:

- If the DIALBYP logonid privilege is specified for the target machine, CA ACF2 for VM does not prompt the dialer for a logonid and password. The dialer is granted DIAL access without going through standard DIAL command validation.
- If the DIALBYP logonid privilege is not specified for the target, CA ACF2 for VM prompts the dialer for a logonid and password and validates the DIAL.

Standard CA ACF2 for VM Validation

CA ACF2 for VM validates the DIAL before DIAL access to the secured target to provide security for DIAL commands issued in the VM operating system environment. In addition, CA ACF2 for VM creates $DIAL and $DROP SMF records. The steps involved for DIAL access include:

1. Validating dialer's logonid and password
2. Validating resource rules
3. Validating DIAL command limiting (optional)
4. Creating a $DIAL SMF record for VM systems when the DIAL command succeeds, and creating a $DROP SMF record when the DIAL is severed.

For more information on $DIAL and $DROP pseudocommands, the Logging DIALs and DIAL Drops section. If the DIAL command does not pass each validation step, access is denied and an SMF logging results. DIAL access is allowed to the secured target only when the DIAL command passes each validation.

Implementing DIAL Protection

The DIALBYP privilege bypasses DIAL command validation. When this logonid privilege is assigned to a target machine, CA ACF2 for VM does not validate a DIAL when you issue it to access the target virtual machine.

Because the DIALBYP logonid record privilege replaces the DIALSEC list in VMXAOPTS of HCPAC0, you should give all users specified in this list the DIALBYP privilege. For masked DIALSEC entries, use CHANGE LIKE to update all matching logonid records.

The steps involved in implementation are listed below.

1. Add the DIALBYP privilege (optional)
2. Modify the resource type code in the TYPES field of the RESTYPE VMO record (optional)
3. Write DIAL resource rules
4. Test DIAL resource rules.

See the *Command and Diagnose Limiting Guide* if you need to implement DIAL command limiting. A brief explanation of these steps follows.

Adding the DIALBYP Privilege

To let a virtual machine bypassDIAL validation, add the DIALBYP logonid privilege to the target machine's logonid record. To change an existing logonid to reflect the DIALBYP privilege, use the LIST subcommand, then issue the CHANGE subcommand. An example follows.

```
acf

ACF

list TLCAMS

TLCAMS  TLCMGRGTLCAMS ANN SMITH EXT 200
PRIVILEGES ACCOUNT AUDIT SCPLIST(ACCTMGR)
 .
 .
```

Next, issue the CHANGE subcommand:

```
acf

ACF

change TLCAMS DIALBYP

TLCAMS  TLCMGRGTLCAMS ANN SMITH EXT 200
PRIVILEGES ACCOUNT AUDIT DIALBYP SCPLIST(ACCTMGR)
 .
 .
```

Ann's virtual machine has the DIALBYP logonid privilege. When this virtual machine is the target of a DIAL command, CA ACF2 for VM will not validate the DIAL. To list all virtual machines with the DIALBYP logonid privilege, enter the LIST IF(dialbyp) subcommand.

Modifying the DIAL Resource Type Code

You can use a site-definedresource type code for DIAL resource rules. To define this type code:

```
acf

ACF

SET CONTROL(VMO)
CHANGE RESTYPE DIA
```

The default value is DIAL=(DIA).

When you modify this operand, you must also modify the TYPES field of the RESTYPE VMO record to reflect this change. These resource type codes must be consistent in both the RESCLASS and RESTYPE records.

Modifying the RESTYPE Record

An example of a modified RESTYPE VMO record is RESTYPE TYPES(ABC). The default type code for DIAL is DIA.

Writing DIAL Resource Rules

An example of a DIAL resource rule is shown below.

```
ACF

set resource(dia)

RESOURCE(DIA)

compile
ACFCMP510I ACF compiler entered

$key(TLCAMS) type(dia)
 uid(*****tlc) allow
 uid(*****TLCPJM) prevent

ACFCMP551I Total record length=168 bytes - 4 percent utilized

RESOURCE
```

Testing DIAL Resource Rules

An example of testing a DIAL resource rule follows.

```
test *
.
uid(*****TLCPJM)
The following parameters are in effect:
 DATE=12/27/97 TIME=1721 SOURCE=******** UID=*****TLCPJM

 ACCESS WOULD BE PREVENTED
.
uid(*****TLCAMS)
The following parameters are in effect:
 DATE=12/27/97 TIME=1721 SOURCE=******** UID=*****TLCAMS

 ACCESS WOULD BE ALLOWED
.
end

RESOURCE
```

Logging DIALs and DIAL Drops

CA ACF2 for VM logs all DIALs and DIAL drops in command limiting SMF records. This uses the pseudocommands $DIAL and $DROP. These pseudocommands introduce a consistent format that always contains the real address of the device you are dialing to, and the virtual address of the target machine you dialed to.

$DIAL Pseudocommand

CA ACF2 for VM always logs the $DIAL pseudocommand through command limiting SMF records for secured target virtual machines. The $DIAL logging always provides the dialed virtual address on the target virtual machine. This $DIAL logging does not occur if the DIAL fails after it passes the standard validations.

The format of the $DIAL SMF record is:

```
$DIAL terminal targetid vaddr
```

For example:

```
$DIAL DISP0420 MVSXA 0817
```

$DIAL is logged under the dialer's logonid and UID.

$DROP Pseudocommand

When CA ACF2 for VM drops a dialed device because of a CP RESET command, an I/O error, or the target machine logged off, it records the $DROP pseudocommand in the command limiting SMF records.

The format of the $DROP SMF record is:

```
$DROP terminal targetid vaddr
```

For example:

```
$DROP DISP0420 MVSXA 0817
```

$DROP is logged under the dialed virtual machine's logonid and UID. Because the $DROP SMF record is identical to the $DIAL record, it is easy to pair these two records together by target ID and virtual address to report total dialed time.

CA ACF2 for VM does not log the $DIAL and $DROP pseudocommands if the target machine has the DIALBYP privilege.

IUCV, APPC/VM, and VMCF Validation and Logging

The Inter User Communication Vehicle (IUCV), Advanced Program-to-Program Communication/VM (APPC/VM), and the Virtual Machine Communication Facility (VMCF) define communication paths for the transfer of data between two processes (virtual machines and CP system services). These methods are types of Interprocess Communications (IPC).

You can establish a fine degree of audit and control over the IPC paths through CA ACF2 for VM resource rules. CA ACF2 for VM validates these resource rules to establish a one-way IPC path connection and logs the establishment and termination of a one-way IPC path connection. These functions make a user accountable for his actions during IUCV, APPC/VM, and VMCF interprocess communications.

The following terms describe CA ACF2 for VM support for IUCV, APPC/VM, and VMCF:

Initiator machine

Invokes a request to perform a data or message transfer.

Target machine

The recipient of a request to perform a data or message transfer. This recipient is not necessarily a virtual machine, but can be a CP system service or an application-define resource ID.

These three communication paths are similar in function. They differ in the amount of data that you can transfer between the initiator and target virtual machines and in their path structures. In the case of APPC/VM, the initiator and target machines can be on physically separate VM systems through the Transparent Services Access Facility (TSAF).

To implement IUCV, APPC/VM, and VMCF validation, you must turn on the appropriate option in the OPTS VMO record (IUCVVLD for an IUCV or APPC/VM path connection, or VMCFVLD for a one-way VMCF path connection). Then, modify the RESCLASS VMO record and, if necessary, the RESTYPE VMO record and the COMSEC operand. This operand is in the VMXAOPTS macro in HCPAC0. After you make these modifications, CA ACF2 for VM validates them through resource rules.

System Access through Resource Rules

When you issue a request to establish or terminate a one-way IPC path connection, CA ACF2 for VM optionally validates the action through a resource rule.

You can define target machines in the COMSEC operand of the SRVMOPTS macro to secure them from unauthorized accesses. The default value is COMSEC=(INCLUDE,-), securing all virtual machines, CP services, and resource IDs for IPC resource rule validation.

When you define the target machines, you can explicitly name the machines, CP system services, and resource IDs in the INCLUDE= option or name the target machines that are not secured in the EXCLUDE= option. You can specify COMSEC=(EXCLUDE,-) in the VMXAOPTS macro to totally disable it.

CA ACF2 for VM then checks the resource rule (if implemented through COMSEC). Each rule entry defines a one way IPC path connection from an initiator virtual machine to a target machine.

CA ACF2 for VM:

- Validates the logonid, appropriate service machine name, CP system service, or resource ID as the $KEY value for resource rules. It validates the target machine (the recipient of a request to perform a data or message transfer) as a rule set $KEY value. You can specify the resource rule's type code in the TYPES field of the RESTYPE VMO record to mask this value.
- Validates the user ID of the initiator machine (the invoker of a request to perform a data or message transfer) as a UID value of an entry in the rule set. Standard UID masking applies for this value.
- The rule set type codes are IUC (to validate IUCV and APPC/VM) and VMC (to validate VMCF) by default. These type codes are defined in the IUCV and VMCF fields of the RESCLASS VMO record.

IUCV Communication Path

Initiate IUCV communication through the IUCV CONNECT function and terminate it through the IUCV SEVER function. The following rule initiates a one way path connection, letting TLCAMS issue an IUCV CONNECT to communicate with VMSYSU The initiator, TLCAMS, must be specified in the UID of the rule entry; the target, VMSYSU, must be specified as the $KEY value:

```
RESOURCE

SET RESOURCE(IUC)

RESOURCE(IUC)

COMPILE
ACFCMP510I ACF compiler entered

$KEY(VMSYSU) TYPE(IUC)
 UID(*****TLCAMS) ALLOW

ACFCMP551I Total record length=144 bytes - 3 percent utilized

RESOURCE
```

Establishing and terminating the IUCV communication path is logged only if the LOG access permission is specified in the rule entry. If this happens,
CA ACF2 for VM creates an audit record during the IUCV CONNECT function (to indicate when the IUCV path was initiated) and during the IUCV SEVER function (to indicate when the IUCV path was terminated).

APPC/VM Communication Path

Initiate APPC/VM communication through the APPC/VM CONNECT function and terminate it through the APPC/VM SEVER function. Because there is no difference between IUCV and APPC/VM, the previous IUCV example also applies to APPC/VM. The rule allows TLCAMS to issue an APPC/VM CONNECT to communicate with a resource ID known as VMSYSU.

Establishing and terminating the APPC/VM communication path is logged only if you specify LOG in the rule entry. If this is the case, CA ACF2 for VM creates an SMF record during the APPC/VM CONNECT function (to indicate when the APPC/VM path was initiated) and during the APPC/VM SEVER function (to indicate when the APPC/VM path was terminated).

The VMSAF LID attribute is required for APPC/VM CONNECT with password validation that is done for APPC security level SECURITY(PGM). This includes an APPC/VM VTAM support (AVS) service machine. See the IBM manuals for more information on APPC/VM and AVS.

VMCF Communication Path

Initiate VMCF communication through the VMCF AUTHORIZE function and terminate it through the VMCF UNAUTHORIZE function. The path is controlled in exactly the same way as IUCV communication.

The following rule initiates a one way path connection, letting TLCPJM issue a VMCF AUTHORIZE to communicate with VMTARG. The initiator, TLCPJM, must be specified in the UID of the rule entry; the target, VMTARG, must be specified as the $KEY value:

```
RESOURCE

SET RESOURCE(vmc)

RESOURCE(VMC)

compile
ACFCMP510I ACF compiler entered

$KEY(VMTARG) TYPE(VMC)
 UID(*****TLCPJM) ALLOW

ACFCMP551I Total record length=144 bytes - 3 percent utilized

RESOURCE
```

Establishing and terminating the VMCF communication path is logged only if the LOG access permission is specified in the rule entry. If this happens,
CA ACF2 for VM creates an audit record during the VMCF AUTHORIZE function (to indicate when the VMCF path was initiated) and during the VMCF UNAUTHORIZE function (to indicate when the VMCF path was terminated).

Reporting IUCV and VMCF

Whenever an invalid path attempt or logging occurs through IUCV and VMCF resource rule validation, CA ACF2 for VM records it in the Resource Event Log (ACFRPTRV).

Important Information

Important details apply when creating IPC resource rules:

- The CA ACF2 for VM service machine and the VM operating system are allowed unconditional IUCV and VMCF access. You do not need to write IPC resource rules for the CA ACF2 for VM service machine and the VM operating system.
- An CA ACF2 for VM front-end feature to IPC resource rules lets you mediate which target machines are secured for validation. We provide this feature with the COMSEC operand of the VMXAOPTS macro in HCPAC0. The default COMSEC specification is (INCLUDE,-), securing all virtual machines and CP services for IUCV and VMCF resource rule validation.

  The COMSEC=(INCLUDE,ALL) option does not secure all virtual machines, CP services, and resource IDs for IPC validation. ALL is a valid target name for VMCF only that you can specify as a $KEY value for VMCF resource validation. ALL indicates that CA ACF2 for VM validates all virtual machines and CP services for VMCF, not those for IUCV or APPC/VM. (INCLUDE,ALL) does not perform resource rule validation for IUCV or APPC/VM.
- IUCV, APPC/VM, and VMCF resource rules initiate a one way path connection. If you want virtual machines to initiate the communication path in either direction, you need two rules.

  ```
  $KEY(VMSYSU) TYPE(IUC)
   UID(TLCAMS) ALLOW

  $KEY(VMTARG) TYPE(IUC)
   UID(TLCPJM) ALLOW
  ```

  The first rule lets TLCAMS issue an IUCV CONNECT to VMSYSU. The second rule lets TLCPJM issue an IUCV CONNECT to VMTARG. TLCAMS is the initiator in the first rule. TLCPJM is the initiator in the second rule.

- The IUCV and VMCF fields of the RESCLASS VMO record define the type codes required for IUCV and VMCF resource rules, respectively. The IUCV field also defines the type code for APPC/VM resource rules. IUCV and APPC/VM always share the same type code. The defaults are IUCV(IUC) and VMCF(VMC). These values implement type codes IUC (to validate IUCV and APPC/VM) and VMC (to validate VMCF).
- You can specify the resource rule's type code in the RESTYPE VMO record to implement $KEY resource name masking for IPC resource rules. This type code must be the same value that is specified in the IUCV and VMCF operands of the RESCLASS VMO record. You can use standard UID masking in individual rule entries.
- You can specify the percent sign (%) in IPC rules to indicate a literal asterisk (*) of a target you want to protect.

```
COMPILE
ACFCMP510I ACF compiler entered

$KEY(%MSG) TYPE(IUC)
 UID(TLCAMS) ALLOW

ACFCMP551I Total record length=144 bytes - 3 percent utilized

RESOURCE
```

  The target of this IUCV rule is *MSG, a CP system service. The percent sign (%) lets you indicate that *MSG is the literal name of the target to protect, not MSG prefixed by some character. To indicate *MSG in the SRVMOPTS COMSEC list, specify %MSG, just like in the $KEY value.
- You should never use the following resource IDs for APPC/VM rules: ANY, ALLOW, or SYSTEM. Also, a resource ID should not be identical to a user ID.
- The TSAF virtual machine issues APPC/VM CONNECT and SEVER on behalf of remote initiator machines. This also applies to any user-written APPC/VM communications servers. The APPC/VM communications server virtual machine is validated for access, not the originator who issued the request to the server.
- The access permissions specified in rule entries perform the following functions:

  ALLOW

  Validates an IPC communication path. CA ACF2 for VM does not create audit records.

  LOG

  Validates an IPC communication path. CA ACF2 for VM creates audit records when the IPC path is established and terminated. This access permission is important for systems running in a C2-rated environment, as the National Computer Security Center (NCSC) defines.

PREVENT

Stops the IPC communication path. CA ACF2 for VM creates audit records for the invalid access attempt. The user and the operator receive CA ACF2 for VM error messages.

CA ACF2 for VM ignores the SERVICE and VERIFY keywords for resource rules for IUCV, APPC/VM, and VMCF communications.

Normal source and shift controls apply to the validation process.

Information for Shared Databases

In a shared database environment, the implementation procedure for IUCV, APPC/VM, and VMCF security is the same. However, there are special considerations that give you increased flexibility when designing IUCV, APPC/VM, and VMCF security for shared databases. This flexibility translates into increased control of resources through the resource type codes. By modifying these codes, you can separate operating system resources for each CPU to meet your unique security requirements.

If you run two VM systems, you can establish separate rule sets for individual systems to implement more control. For example, you can separate the rule sets for IUCV, APPC/VM, and VMCF through the type code specified in the CONTROL(VMO) record and resource rules.

First, modify the resource type code in the RESCLASS VMO record:

```
SET CONTROL(VMO)
```

For SYSTEMA:

```
CHANGE RESCLASS IUCV(ISA) VMCF(VSA)
```

Next, modify the RESTYPE VMO record.

```
CHANGE RESTYPE TYPES(ISA,VSA)
```

After you modify the resource type codes, write resource rules to use the new type codes.

```
$KEY(VMSYSU) TYPE(ISA)
UID(TLCAMS) ALLOW
```

For more information about shared databases, see the *Systems Programmer Guid*e.

POSIX Supplemental Group Validation

With CA ACF2 for VM, you can assign a virtual machine to a primary or default POSIX group. A user can change the current POSIX group to a supplemental group with the POSIX setgid( ) or setegid( ) function or with the shell newgrp command.

The CA ACF2 for VM method for employing POSIX group validation is significantly different from native VM. Specifically, CA ACF2 for VM POSIX group validation takes place through resource rules, GROUP logonid values, and GROUP and USER Profile values that replace the CP directory POSIXGROUP and POSIXINFO statements.

POSIX Controls

CA ACF2 for VM provides the following controls for POSIX support:

POSIX group resource rules

CA ACF2 for VM definitions that list POSIX GROUP names and the virtual machines that can use them. These rules work with GROUP logonid values to functionally replace the native VM implementation POSIXGROUP and POSIXINFO directory control statements provide.

POSIXDB

An operand in the OPTS VMO record that specifies that CA ACF2 for VM provides POSIX database management.

NOPOSIXDB

An operand in the OPTS VMO record that specifies that CA ACF2 for VM does not provide POSIX database management. This is the default.

MAXPGRPS

An operand in the OPTS VMO record that defines the maximum number of POSIX groups allowed for each user. The default is 32, which is also the minimum value. The maximum is 125.

POSIXGRP

An operand in the RESCLASS VMO record that defines the three-character resource type code for POSIX supplemental group resource rule sets. PGR is the default.

GROUP

An eight-byte logonid field that holds the primary or default group for a virtual machine. The GROUP value works with POSIX Group resource rules and Group and User Profile records to functionally replace VM POSIX directory control statements.

USER Profile records

These records define POSIX environment information to CA ACF2 for VM for each user that runs POSIX. The key to the record is the user's logonid name.

GROUP Profile records

These records relate group names to POSIX GIDs (Group Identification numbers). The key to the record is the group name.

Primary logon groups

A user can belong to more than one group. The user's primarygroup is set in their LIDREC in the GROUP field. You can identify all other groups that a user can access (setgid()|setegid()) through resource rules.

Resource(PGR) group rules

Users are allowed to belong to multiple groups in the OpenExtensions VM environment. A user's primary or default group is defined in the GROUP field of his logonid.

Control(VMO) RESTYPE record

You must define POSIX Supplemental Group resource rules in the resident resource rule cache. The resident resource cache can have several resource types in it.

Control(VMO) OPTS record

You can maintain the OpenExtensions VM Database information in the VM directory or with an External Security Manager (ESM) such as CA ACF2 for VM. The POSIXDB option in the OPTS VMO record identifies whether CA ACF2 for VM activates OpenExtensions VM Database support.

ACFSERVE commands

Whenever you create, change, or delete a USER or GROUP Profile record, you must issue the appropriate ACFSERVE command to rebuild the directories for the Profile records so CA ACF2 for VM recognizes these changes.

CA ACF2 for VM diagnose limiting

CP allows users to inquire on the contents of the OpenExtensions VM Database through new CP diagnose calls. The new diagnose calls extract OpenExtensions VM Database and runtime information or issue setid() requests. See the *Command and Diagnose Limiting Guid*e for more information on these diagnose calls.

See the “OpenExtensions VM Support” chapter for additional information on Open Edition VM support.

VM Dataspace Security

CA ACF2 for VM validates all VM dataspace accesses. This validation occurs at PERMIT time, not at actual access time.

CA ACF2 for VM activates dataspace protection by default at installation time. You can turn it off by changing the OPTS VMO record to NODSPVLD. To turn it back on, change the OPTS VMO record to DSPVLD.

The resource rule type for dataspace validation is the DSPACE(xxx) keyword of the RESCLASS VMO record. The default DSPACE resource rule type for dataspaces is DSP.

Like IUCV validation, the owner of a dataspace must be defined in the COMSEC list before dataspace validation can occur. Unlike IUCV validation rules, dataspace rules must distinguish between READ and WRITE access to the dataspace. Therefore, dataspace rules use the SERVICE keyword of resource rules to distinguish between READ access (which is read-only) and UPDATE access (both READ and WRITE).

Following is a sample dataspace access rule that permits TLCAMS read access to any dataspaces TLCJAM created, and lets TLCMEG have write access:

```
$KEY(TLCJAM) TYPE(DSP)
 UID(TLCAMS) SERVICE(READ) ALLOW
 UID(TLCMEG) SERVICE(UPDATE) ALLOW
```

CA ACF2 for VM issues the following message to the issuer of the ADRSPACE MACRO PERMIT function for dataspace violations:

```
Dataspace WRITE access denied for userid TLCAMS
```

CA ACF2 for VM issues the following message to the system operator and the target user ID, TLCAMS, at the same time:

```
Dataspace WRITE access to userid TLCJAM denied for userid TLCAMS
```

Since a long-running server usually grants dataspace permission on behalf of a user, CA ACF2 for VM does not increment the violation count of the server issuing the dataspace permit. This is so CA ACF2 for VM does not log off the server.

Chapter 17: Protecting Tapes

This chapter explains how to enforce tape volume security with CA ACF2 for VM. When you finish reading this chapter, you will know:

- How to protect tape volume access through the ACFMOUNT command
- How to limit the use of the ACFMOUNT command
- How the ACFMOUNT command is validated
- The procedures the system operator must follow to protect tapes.

This section contains the following topics:

Protecting Access to Tape Volumes

You can protect access to tape volumes through CA ACF2 for VM validation and system operator procedures. To access a tape volume:

1. Enter the ACFMOUNT command specifying the volume serial number of the tape. You can enter the virtual device address of the tape drive where the tape will be mounted and the type of operation performed (read or write).
2. If you implement command limiting for the ACFMOUNT command, the ACFMOUNT command limiting rule is validated to see if you have the authority to issue the command. ACFMOUNT command limiting is not required to implement CA ACF2 for VM tape volume access validation. ACFMOUNT command limiting enhances auditing.
3. CA ACF2 for VM performs ACFMOUNT command validation that is not related to command limiting.
4. A tape volume access rule is validated to see if you can read from or write to the specified tape volume.
5. If access is allowed, CA ACF2 for VM prompts the system operator to mount the indicated volume.

Using the ACFMOUNT Command

The ACFMOUNT command lets you access a tape volume. The syntax for the ACFMOUNT command is:

```
ACFMOUNT volser [vadr access]
```

The parameters for the ACFMOUNT command are defined below.

volser

Represents the one- to six-alphanumeric character volume serial identification of the mounted tape. You must issue this parameter. There is no default.

vadr

Represents the virtual device address of the tape drive where the tape is mounted. If you specify this parameter and CA ACF2 for VM grants access to the requested tape volume after performing validation, CA ACF2 for VM informs the operator to mount the tape on the tape drive's real device address. If you omit this parameter and CA ACF2 for VM grants access to the requested tape volume after performing validation, CA ACF2 for VM tells the operator to mount the tape without specifying the device. This parameter is optional. There is no default.

access

Indicates the optional requested operation to be performed, READ (the default) or WRITE. You cannot abbreviate this parameter.

Command Limiting ACFMOUNT

ACFMOUNT command limiting is not a requirement for tape volume access validation. If you use ACFMOUNT command limiting, command entries that are logged or prevented are reported in the Command Limiting Journal (ACFRPTCL).

To implement command limiting for ACFMOUNT, the COMMANDS operand of the CMDLIM VMO record must indicate ACFMOUNT. If you use an INCLUDE list, you must specify ACFMOUNT. If you use an EXCLUDE list, do not specify ACFMOUNT. To write an ACFMOUNT command limiting rule, use the ACF COMPILE subcommand of the CMDLIM setting. See the *Command and* Diagnose *Limiting Guid*e for more information. The following rule set lets any user with the UID mask beginning with TLC enter the ACFMOUNT 767305 181 WRITE command. Normal masking for command limiting rules applies.

```
$KEY(ACFMOUNT)
 767305 - UID(*****TLC) LOG
```

Validating the ACFMOUNT Command

CA ACF2 for VM performs ACFMOUNT command validation that is not related to command limiting. CA ACF2 for VM checks that you entered the command syntax correctly. If you entered the virtual device address of the tape drive where the tape is mounted, CA ACF2 for VM validates whether the device is attached. CA ACF2 for VM then checks if the device is a tape drive. If the command is not accepted, CA ACF2 for VM sends you a message. The message explains the reason for the invalid request. For example, you could see the following message:

```
ACFpgm658E Volser of tape invalid or missing.
```

This means that you did not enter the volume serial identification or entered it incorrectly. An example of an invalid entry for a volser is 1234567; consisting of one to six alphanumeric characters. The return code is 008. The other messages that CA ACF2 for VM might send you are explained below.

```
ACFpgm654E DEVICE NOT ATTACHED
```

The virtual device address of the requested tape drive is not defined. The return code is 012.

```
ACFpgm655E DEVICE NOT A TAPE DRIVE
```

The virtual device address of the requested device is defined, but not as a tape drive. The return code is 016.

```
ACFpgm656E 'parameter' IS AN INVALID PARAMETER
```

An invalid parameter after the volume serial identification was entered. The return code is 020.

```
ACFpgm657E SYSTEM OPERATOR IS NOT RECEIVING-MOUNT MESSAGE NOT ISSUED
```

The system operator is not receiving messages. The system operator console may be logged off or disconnected without a secondary console. The return code is 024.

Validating Access to Tape Volumes

CA ACF2 for VM validates tape volume access, checking if you can read from or write to the specified tape volume. Rule validations that are logged and prevented are reported in the Data Set Access Journal (ACFRPTDS). To validate tape volume access, you must specify the volume serial label (volser) for the volume in the VOLMASK field of the TAPEVOLS VMO record.

System Operator Procedures

If CA ACF2 for VM grants access to the requested tape volume after performing validation, a message prompts the system operator to mount the tape. Possible messages and the action to take are:

```
ACFpgm650A MOUNT TAPE VOLUME 'volser' ON 'raddr' WITH A RING
```

Mount the tape volume volser onto a tape drive with the real address radr. The message also specifies to include a write ring on the tape. This message occurred because you specified the ACFMOUNT command with the vadr parameter and indicated WRITE access.

```
ACFpgm652A MOUNT TAPE VOLUME 'volser' ON 'raddr' WITHOUT A RING
```

Mount the tape volume volser onto a tape drive with the real address radr. The message also specifies that the tape does not need a write ring. This message occurred because you specified ACFMOUNT with the vadr parameter and indicated READ access.

```
ACFpgm651A MOUNT TAPE VOLUME 'volser' FOR USERID 'lid' WITH A RING
```

Mount the tape volume volser onto any tape drive for lid. The message specifies to include a write ring on the tape. This message occurred because you specified ACFMOUNT and indicated WRITE access

```
ACFpgm653A MOUNT TAPE VOLUME 'volser' FOR USERID 'lid' WITHOUT A RING
```

Mount the tape volume volser onto any tape drive lid. The message specifies that the tape does not need a write ring. It occurred because you specified ACFMOUNT and indicated READ access.

These messages do not necessarily mean that tape volume access validation has occurred. They also appear if you correctly enter ACFMOUNT and specify a volume that is not secured in the TAPEVOLS VMO record.

The main concern for a system operator is to mount a tape only when CA ACF2 for VM prompts. The message gives explicit instructions for doing this.

You may want the system operator to be responsible for enforcing additional controls. The system operator must be aware of tape volume (object) reuse considerations and implement the appropriate protection measures.

1. When you notify the operator that a tape is no longer needed (it becomes a scratch tape), the operator must erase the data on the tape or demagnetize the tape volume to remove its data. Use the ACFERASE utility for this. (You can find information about using the ACFERASE utility in the *Reports and Utilities Guide.*
2. The operator must archive tape volumes in a secure location. Preferably, you should keep them in the same room with the system console. (This is an important procedure for any site.)

Protecting Tape Reuse

The ACFERASE utility erases all of a tape's data, protecting against unauthorized data access to implement tape volume reuse protection. This utility is a standalone program. It executes in the CMS transient area and uses the CMS tape macros. Give the person responsible for the protection of tape volumes authority to use this utility. This person is usually the system operator. You should never give this privilege to any other user. You can find more information about the ACFERASE utility and the ACFERASE commands in the *Reports and Utilities Guid*e.

Chapter 18: Maintaining Entry Source and Source Group Records

Source entry records define individual input sources. They are stored on the Infostorage database and have a storage class value of “E.”

You can use entry records to:

- Identify individual input sources (such as terminals or card readers) for CA ACF2 for VM validation. Input source records translate physical input sources to logical input sources. CA ACF2 for VM recognizes both.
- Identify groups of input sources for CA ACF2 for VM validation. These groups can be made up of logical or physical input source names.

When you finish this chapter, you will know:

- The SRC and SGP source entry records
- The name conventions used for physical units

How to use the SET ENTRY setting and ACF subcommands to create, change, delete, and display records

- How to activate entry records with the ACFSERVE command.

This section contains the following topics:

The XREF Source Group Record

The XREF setting lets you use the XREF Source Group Record (X-SGP) in place of the Entry Source Group Record (E-SGP) for grouping sources. Here are considerations:

- You can use masking characters. For example, suppose your site has four departments: payroll, personnel, operations, and programming. All source names begin with the first three letters of the department, such as PAY for payroll and PER for personnel. You could reduce the number of entries in a source group to one using a mask, for example PAY-.

  However, using masking can increase the time CA ACF2 for VM takes to validate an access attempt. CA ACF2 for VM uses a binary search to locate a source name entry in an unmasked record. However, because of the ability to use source names or masks, CA ACF2 for VM uses a modified sequential search to locate a source name in a masked record. If the masked record contains many source names, CA ACF2 for VM takes longer to process the masked record.

- You can exclude sources from a source group. For example, you might want to grant write access to users in the payroll department who log on at terminals PAY01 through PAY15, but prevent them from writing if they log on from the group terminal, PAY07. You could specify INCLUDE(PAY-) and EXCLUDE(PAY07) in the X-SGP record for the PAY source group.

- You can define source groups and sets of source group records separately.

- You can use standard CA ACF2 for VM scope records for X-SGP records, whereas the E-SGP records require that you use a pseudo data set name to obtain a scoping function.

Entry records that define single sources (the E-SRC records) will continue to be supported in future releases of CA ACF2 for VM. An advantage of E-SGP records is the ability to use the NEWDATA, VERDATA, and OLDDATA parameters to change all records with a particular source entry. For example, suppose an employee moves from the payroll department to the personnel department. If his old source was named PAY57, and his new source was named PER33, you could change every instance of PAY57 to PER33 using a single CHANGE subcommand:

```
CHANGE LIKE(-) OLDDATA(PAY57) NEWDATA(PER33)
```

For more information about X-SGP records, see the chapter “Maintaining Cross-Reference Records” chapter.

Types of Source Entry Records

In storage class “E,” there are two different type codes: SRC (for source records) and SGP (for source group records).

Sources (SRC)

The source entry records identify the individual input sources for CA ACF2 for VM validation. This record translates a physical input source to a logical input source name. For example, you can translate a terminal, GRAF0494, to a logical source, ROOM001. When listed, this source entry record looks like this:

```
TYPE: SRC  ENTRY: GRAF0494  1 DATA ITEM
ROOM001
```

In the above example, the type code SRC identifies this record as an input source record. GRAF0494 is the name of the entry record, and also the physical name that identifies a terminal. The data item, ROOM001, defines a logical name for the terminal. The user has chosen this logical name because it is more meaningful than the name GRAF0494. CA ACF2 for VM expects only one logical source for each source entry record. If you specify more than one, CA ACF2 for VM ignores the others.

You must put the logical name of the input source in the logonid record to protect sources using the SRC type code.

Source groups (SGP)

The source group entry records identify groups of input sources for CA ACF2 for VM validation. A source group can contain physical or logical input source names and logical records that are themselves groups of sources. The source group name is a logical name that maps many input sources to a specific source group. Source groups let you write rules for groups of sources rather than each individual input source. For example, you can define the PAYROLL department as a source group and specify various input sources, GRAF0480, GRAF0481, and GRAF0482, in that group. (You can also use a logical name, like ROOM001, that was defined in a SRC entry in the source group.) When listed, this source group entry record looks like this:

```
TYPE: SGP  ENTRY: PAYROLL 3 DATA ITEMS
GRAF0480
GRAF0481
GRAF0482
ROOM001
```

In the above example, the type code SGP identifies this record as a source group record. PAYROLL is the name of the entry record, and is a logical name that identifies all terminals in the PAYROLL department. This record contains three data items that identify the individual terminals or groups that make up the source group.

If you are using eight-character names, a source group record can contain a maximum of 442 entries. If you are using less than eight characters, the number of entries in the source group record will be greater.

You can define an individual input source simultaneously in more than one source group. To avoid confusion, a physical device name cannot be the same as an SRC entry record name in an SGP entry record.

As stated before, a source group can also contain logical records that are themselves groups of sources. This lets you define groups in larger groups.

For example, suppose you have three groups of terminals, some users are permitted to log on to all three groups, and the other users can log on to only one of the three groups. In this case, you want to define each of the three individual groups of terminals, and one larger group that consists of all three individual groups.

To do this, you define a total of four X-SGP records, one for each individual group of terminals, and one for the larger group that consists of all three of the individual terminal groups. For each X-SGP record ID (recid), you use the name of the terminal group or the name of the set of terminal groups that the X-SGP record defines. You may have X-SGP records with record IDs such as TERMA, TERMB, TERMC, and GROUP1, where GROUP1 consists of TERMA, TERMB, and TERMC.

After you have defined all of the X-SGP records, specify one of the individual terminal groups (TERMA, TERMB, or TERMC) or the set of all three terminal groups, GROUP1 in the SOURCE field of the users' logonid records.

Specify TERMA in the SOURCE field of the logonid records for those users who are permitted to log on to only TERMA terminals.

- Specify TERMB in the SOURCE field of the logonid records for those users allowed to log on to only the terminals in TERMB.
- Specify TERMC in the SOURCE field of the logonid records for those users permitted to log on to only TERMC terminals.
- Specify GROUP1 in the SOURCE field of the logonid records for those users permitted to log on to TERMA, TERMB, and TERMC terminals.

You can also have a set of X-SGP records cross-reference another set of X-SGP records. Extending the scenario just described, suppose that you want to permit some users to have access to a system from the set of terminal groups indicated in the X-SGP GROUP1 record and from another set of terminal groups defined in the record X-SGP GROUP2. In this case, you could define an X-SGP record called GROUPA. GROUPA consists of GROUP1 and GROUP2.

You can use this X-SGP to X-SGP cross-reference ability as an indexing process to define up to 25 levels.

Fields in Each Type of Entry Record

The fields of an entry record are referred to as data items. For each type of entry record, the following table summarizes the type code and what information is permitted for the record name and data items:

Type Code	Record Name	Data Item
SRC	Physical input source name	Logical input source name to represent the physical input source name (one data item only).
SGP	Source group name	Physical or logical names of input sources in the group, or names of source groups contained in the group.

Naming Source or Source Group Records

Use care when naming source and source group records. CA ACF2 for VM considers all E-SGP record names source group names. It also considers all entries in an E-SGP record as individual sources. The exception is an entry in an E-SGP record that is also the name of another E-SGP record. In this case,
CA ACF2 for VM considers the entry a group name.

For example, suppose you create an E-SGP record named ACCT with the following entries: AAA, BBB, CCC, DDD, and ZZZ. CA ACF2 for VM considers each of the entries an individual source. Later you create another E-SGP record named ZZZ with the following entries: MMM, LLL, YYY, and XXX. CA ACF2 for VM considers ZZZ to be the name of a source group. This means that all the entries in ZZZ will become part of the ACCT group. The ACCT group now includes sources MMM, LLL, YYY and XXX, but ZZZ is no longer considered as an input source.

Physical Unit Naming Conventions

When creating a source (SRC) or source group (SGP) record, you must follow specific naming conventions. Naming conventions for graphics, logical devices, and VTAM devices are explained in the following sections.

Graphic Devices

GRAFccuu

Specifies terminals attached to the CPU

GRAF

Specifies a four-character constant to define a terminal

ccuu

Specifies a four-character device address that defines a terminal

Logical Devices

Although we support rule writing for logical devices, it is difficult to protect them. This is because logical device numbers are assigned at random and can become unpredictable. In future releases, we will use CA-VTERM to pass the physical device name (GRAFxxxx) to CA ACF2 for VM. This enhancement lets CA ACF2 for VM users protect by physical input sources.

LDEVldevname

Specifies the logical device name

LDEV

Specifies a four-character constant to define a logical device

ldevname

Specifies a four-character logical device name

VTAM Devices

Luname

Specifies a VTAM logical unit name

luname

Specifies an eight-character logical unit name

Processing Entry Records

Users with the CA ACF2 for VM SECURITY privilege can issue ACF subcommands to create, change, and display entry records.

Setting Entry Record Mode

Entry records are processed under the ENTRY setting of the ACF command. The required type code for processing input source records is SRC. The required type code for source group records is SGP. The syntax of the SET subcommand to establish this setting is:

```
SET { ENTRY(SRC)  }
    { ENTRY(SGP)  }
```

ENTRY(SRC)

Indicates you want to create input source entry records

ENTRY(SGP)

Indicates you want to create source group entry records

After you have established the ENTRY setting, you can use other ACF subcommands to process your source and source group records. The last SET ENTRY setting determines the type of entry record that is processed with the other ACF subcommands.

Creating Entry Records with the INSERT Subcommand

Use the INSERT subcommand under the ENTRY setting to create a record that translates a physical source name into a logical name. Never use the same record name for an SGP ENTRY record and SRC ENTRY record. The syntax for this INSERT subcommand is:

```
INsert recordname
       [USING(modelname) [TYPE(type)]]
       [NEWDATA(newdata)]
       [DSN(dsn)]
       [CLEAR]
```

recordname

Specifies the one- to eight-character entry record name

USING(modelname)

Obtains the model record name and uses the pseudo data set name and data items rather than starting from a new entry

TYPE(type)

Obtains the model record name from an entry type other than the one you specified in the last SET ENTRY subcommand

NEWDATA(newdata)

Specifies the entry record is a new data item to add

DSN(dsn)

Specifies a filename for authorization control. If a user has access to the dsn, the user can change the entry record. Use this operand with access rules to determine the authority level a user has with this record.

```
DSN(TLCAMS.DATA)
$KEY(TLCAMS)
 DATA UID(TLCPJM) r(a)
 DATA UID(TLCGLB) r(a) w(a)
```

In this example, TLCPJM can list DSN(TLCAMS.DATA) and TLCGLB can list and change DSN(TLCAMS.DATA).

CLEAR

CLEAR has meaning only when you specify a model record name. It eliminates all data items before adding the NEWDATA operand data (except for DSN).

Source Records

An example of how to use the INSERT subcommand to create an SRC record under the ENTRY setting is shown below.

```
SET ENTRY(SRC)

INSERT GRAF0490 NEWDATA(RM433)
```

In this example, CA ACF2 for VM translates the physical name of an input terminal (GRAF0490) into the logical name (RM433) of its room location. To authorize this terminal source as defined, you must add RM433 to the source field of the user's logonid record. If more than one terminal is located in the room, you must add a record for each physical device, or you could add one source group record as described above.

Source Group Records

An example of how to use the INSERT subcommand to create an SGP record under the ENTRY setting is shown below.

```
SET ENTRY(SGP)

INSERT RM433 NEWDATA(GRAF0490)
```

In this example, CA ACF2 for VM lets anyone with a source of RM433 in his logonid record to access the system from the real source, GRAF0490. To authorize using this terminal source as defined, you must add RM433 to the source field of the user's logonid record.

You use the INSERT subcommand only when adding the first source to a source group. You must use the CHANGE subcommand to add all subsequent sources to a source group record. The syntax for using the CHANGE subcommand to modify entry records is explained in Changing Entry Records.

Activating and Changing Entry Records

To activate a newly created or modified entry record, issue the ACFSERVE RELOAD XREF command. This command dynamically updates the source entry records cross-reference table. If you do not use this command, any changes to this cross-reference table are not effective until you IPL the service machine. For details, see the “Defining Structured Infostorage Records” chapter.

Changing Entry Records

Use the CHANGE subcommand under the ENTRY setting to add sources to a source group or update existing source groups. The syntax for this CHANGE subcommand is:

```
       {      *              }
CHange { recordname          }
       { LIKE(recmask)       }

       { [ OLDDATA(olddata) ]}
       { [ NEWDATA(newdata) ]}
       { [ VERDATA(verdata) ]}

         [ DSN(dsn)          ]
         [ CLEAR             ]
```

*****

Indicates you want to change the last record of this type processed since establishing the ENTRY setting with a specific type-code.

recordname

Specifies the one- to eight-character entry record name record to be changed.

LIKE(recmask)

Selects only entry records that match the specified mask. (Never use the same record name for an SGP entry record and SRC entry record.)

OLDDATA(olddata)

Specifies old data information.

VERDATA(verdata)

Specifies verification data.

NEWDATA(newdata)

Specifies new data.

DSN(dsn)

Specifies a filename for authorization control. Any user that can access this file can modify the entry record.

CLEAR

Clears an existing entry record of all data items. It eliminates all data items (except for DSN) before adding the NEWDATA operand data.

Examples

You must specify at least one of the following operands: CLEAR, DSN, NEWDATA, OLDDATA, VERDATA. You can specify more than one.

- NEWDATA adds an item to the list:

  ```
  CHANGE RM433 NEWDATA(GRAF0491)
  ```

  Under the ENTRY(SGP) setting, CA ACF2 for VM adds the input source GRAF0491 to the source group RM433.

- OLDDATA removes an item from the entry record:

  ```
  CHANGE RM433 OLDDATA(GRAF0491)
  ```

  CA ACF2 for VM removes the source GRAF0491 from the input source group RM433.

- Specify NEWDATA and VERDATA with the LIKE parameter and a mask. The NEWDATA item is added to all entry records containing the VERDATA item whose group name matches the mask supplied in the LIKE parameter:

  ```
  CHANGE LIKE(RM***) VERDATA(GRAF0491) NEWDATA(GRAF0492)
  ```

  CA ACF2 for VM adds the input source GRAF0492 to each group whose name begins with RM and extends up to seven characters and that already contains the input source GRAF0491.

- When you specify OLDDATA and VERDATA, and a match occurs, the entry is deleted.

- NEWDATA and OLDDATA replaces data:

  ```
  CHANGE RM433 OLDDATA(GRAF0492) NEWDATA(GRAF0490)
  ```

  CA ACF2 for VM removes input source GRAF0492 and replaces it with GRAF0490 in the source group RM433.

For information on how to activate a changed entry record, see the Activating and Changing Entry Records section.

Displaying Entry Records

Use the LIST subcommand under the ENTRY setting to list a record name or all entries that match a record name mask.

```
      { *             }
List  { recordname    }
      { LIKE(recmask) }
```

*****

Specifies that you want to list the last record of this type processed since you established the ENTRY setting with a specific type code.

recordname

Specifies that you want to list the one- to eight-character name of an individual entry record.

LIKE(recmask)

Selects only entry records that match the mask you specified.

```
LIST RM433
TYP: SGP ENTRY RM433 3 DATA ITEMS
  GRAF0490
  GRAF0491
  GRAF0492
```

Deleting Entry Records

Use the DELETE subcommand under the ENTRY setting to remove a record name or all entries that match a record name mask. The syntax is:

```
       { *             }
DELete { recordname    }
       { LIKE(recmask) }
```

*****

Specifies that you want to delete the last record of this type processed since you established the ENTRY setting with a specific type code.

recordname

Specifies that you want to delete the one-to eight-character name of an individual entry record.

LIKE(recmask)

Selects only entry records that match the mask you specified.

Ending Entry Record Processing

When you have finished processing entry records, enter the following command:

```
END
```

This exits you from the ACF command setting.

Chapter 19: Maintaining Cross-Reference Records

The cross-reference (XREF) setting lets you define groups of sources or groups of resources for CA-validation processing. You need to define the sources or resources in a group only once. Then use the group's name whenever you need to specify the same group again.

For example, to let users access a system only through a particular group of terminals, you could define the individual terminals in the group and assign the group a source group name in an XREF source group (X-SGP) record. You then need to specify only the source group name, rather than each individual terminal, in the SOURCE field of the users' logonid records.

During system entry validation, CA ACF2 for VM matches the group of terminals defined in the X-SGP record with the source group named in the logonid records to discover which terminals the users are permitted to use to enter the system.

This cross-reference concept applies to resource grouping too. For example, to let users access only a particular group of transactions, you could define the individual transactions in the group and assign the group a resource group name in an XREF resource group (X-RGP) record. Then specify the resource group name in the resource rule record key and write only one resource rule for the entire group of transactions. You save yourself the work of writing a rule for each transaction.

During resource access validation, CA ACF2 for VM matches the group of transactions defined in the X-RGP record with the resource group named in the resource rule to discover which transactions the users are permitted to access.

This chapter describes:

- The function of the X-SGP record
- The function of the X-RGP record
- Using the ACF command
- X-SGP record samples

  X-RGP record samples

- Migration considerations
- Activating XREF records.

This section contains the following topics:

Cross-Reference Source Group (X-SGP) Records

You can use the X-SGP record or the E-SGP record for grouping sources. However, we recommend that you use the X-SGP record because it provides more processing options. For more information about E-SGP records, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guid*e. The following sections describe the function of the X-SGP record and how to define the record.

What Source Group Records Do

Depending on the type of access to be validated, the X-SGP record cross-references the SOURCE field in one of the following records:

- Logonid record
- Access rule record
- Resource rule record.

An X-SGP record can also cross-reference another X-SGP record.

Using the SOURCE Field of the Logonid Record

For system access validations, CA ACF2 for VM matches the source group defined in the X-SGP record with the source group named in the SOURCE field of the user's logonid record. To ensure that a user can log on to only a particular group of terminals, you can define the terminals belonging to the group in the X-SGP record, then specify the name of the source group record in the SOURCE field of the user's logonid record. For more information about logonid records, see the “About the Logonid Record” chapter.

Using the SOURCE Field of the Data Set Access Rule

For data set access validations, CA ACF2 for VM matches the source group defined in the X-SGP record with the source group named in the SOURCE field of the data set access rule. To ensure that users can access a data set only from a particular group of terminals, you can define the terminals belonging to the group in the X-SGP record, then specify the name of the source group record in the SOURCE field of the access rule written for the data set. For more information about data set access rules, see the "About Access Rules" chapter.

Using the SOURCE Field of the Resource Rule

For resource access validations, CA ACF2 for VM matches the source group defined in the X-SGP record with the source group named in the SOURCE field of the resource rule. To ensure that users can access a resource only from a specific group of terminals, you can define the terminals belonging to the group in the X-SGP record, then specify the name of the source group record in the SOURCE field of the resource rule written for the resource. For more information about resource rules, see the "About Resource Rules" chapter.

Cross-Referencing X-SGP Records

Besides cross-referencing the other CA ACF2 for VM records just described, X-SGP records can cross-reference each other. This cross-reference function lets you define groups in larger groups.

For example, suppose you have three groups of terminals, some users are permitted to log on to all three groups, and the other users can log on to only one of the three groups. In this case, you want to define each of the three individual groups of terminals, and one larger group that consists of all three individual groups.

To do this, you define a total of four X-SGP records, one for each individual group of terminals, and one for the larger group that consists of all three of the individual terminal groups. For each X-SGP record ID (recid), you use the name of the terminal group or the name of the set of terminal groups that the X-SGP record defines. You may have X-SGP records with record IDs such as TERMA, TERMB, TERMC, and GROUP1, where GROUP1 consists of TERMA, TERMB, and TERMC.

After you have defined all of the X-SGP records, specify one of the individual terminal groups (TERMA, TERMB, or TERMC) or the set of all three terminal groups, GROUP1 in the SOURCE field of the users' logonid records.

- Specify TERMA in the SOURCE field of the logonid records. for those users who are permitted to log on to only TERMA terminals
- Specify TERMB in the SOURCE field of the logonid records. for those users allowed to log on to only the terminals in TERMB
- Specify TERMC in the SOURCE field of the logonid records. for those users permitted to log on to only TERMC terminals
- Specify GROUP1 in the SOURCE field of the logonid records for those users permitted to log on to TERMA, TERMB, and TERMC terminals.

You can also have a set of X-SGP records cross-reference another set of X-SGP records. Extending the scenario just described, suppose that you want to permit some users to have access to a system from the set of terminal groups indicated in the X-SGP GROUP1 record and from another set of terminal groups defined in the record X-SGP GROUP2. In this case, you could define an X-SGP record called GROUPA. GROUPA consists of GROUP1 and GROUP2.

You can use this X-SGP to X-SGP cross-reference ability as an indexing process to define up to 25 levels.

X-SGP Record Fields

Only the size of the record limits the combined field lengths for X-SGP records. You can define the following fields in an X-SGP record:

Record ID	Fields
grpname	SOURCE|GROUP INCLUDE(entry1,...,entryn) EXCLUDE(entry1,...,entryn)

The fields of the source group record are described below:

grpname

Specifies a one- to eight-character name of the source group.

SOURCE|GROUP

Specifies the function of the record. SOURCE indicates that the record defines a group of sources by logical or physical source names. The default value is SOURCE.

GROUP indicates that the record defines a group of source groups. Specify GROUP only when you want to define an X-SGP record that cross-references other X-SGP records. See the Cross-Referencing X-SGP Records section for an example.

INCLUDE(entry1,...,entryn)

Specifies the terminals, group of terminals, or group of terminal groups that you want to include in the record.

Each entry (whether it is a terminal ID or a record ID) can consist of a maximum of eight characters. Separate the entries using commas or blanks. You can mask entries using the asterisk or dash characters. For details on how to use masking characters, see the Masking Logonids section in the "About the Logonid Record" chapter. Only the size of the record limits this field, that is, approximately 4K.

- If you specify SOURCE, INCLUDE specifies the individual terminals, logical or physical, or a combination of both, that belong to this source group.
- If you specify the GROUP keyword, INCLUDE specifies the record IDs of the X-SGP records that this X-SGP record cross-references.

You can specify terminal IDs or record IDs; you cannot specify both in the same X-SGP record. That is, to use both the SOURCE and the GROUP functions, you need to define a separate X-SGP record for each and define a unique record ID for each record.

EXCLUDE(entry1,...,entryn)

Specifies the terminals, group of terminals, or group of terminal groups that you do not want to include in the record. Specify EXCLUDE entries only when you mask the INCLUDE field. For example, suppose you have 20 terminals whose IDs all begin with GRAF01 and you want to include all but two of these terminals in your source group. Specify GRAF01- in the INCLUDE field and specify the full terminal IDs of the two terminals you want to exclude, GRAF0105 and GRAF0106, in the EXCLUDE field.

Each entry (whether it is a terminal ID or a record ID) can consist of a maximum of eight characters. Separate the entries using commas or blanks. You can mask entries. For details on how to use masking characters, see the Masking Logonids section in the "About the Logonid Record" chapter. Only the size of the record limits this field, that is, approximately 4K.

If you specify SOURCE, EXCLUDE specifies the individual terminals that are excluded from the group defined by the INCLUDE field.

- If you specify GROUP, EXCLUDE specifies the record IDs to be excluded from the set of X-SGP records that this X-SGP record cross-references.

You can specify either terminal IDs or record IDs; you cannot specify both in the same X-SGP record. That is, to use both the SOURCE and the GROUP functions, you need to define a separate X-SGP record for each and define a unique record ID for each record.

How CA ACF2 for VM Sorts X-SGP Records

When you specify the EXCLUDE field, remember that CA ACF2 for VM performs sort processing from the most specific to the most general. The most specific entry is always chosen as the valid entry, because it is matched first.

Therefore, if you specify GRAF01*5 in the INCLUDE field, and GRAF0105 in the EXCLUDE field, GRAF0105 is excluded, even though it matches the masked field, because GRAF0105 is more specific. More importantly, remember the EXCLUDE entry can **override** the INCLUDE entry. For example, if you specified GRAF01- in the INCLUDE field and GRAF01 in the EXCLUDE field, the GRAF01 entry overrides the GRAF01- entry because it is more specific. Also, by default, if the INCLUDE field and the EXCLUDE field have the same entry, the entry is excluded from the group.

What Resource Group Records Do

The X-RGP record cross-references the resource name and the $TYPE control statement in a resource rule and can cross-reference another X-RGP record. For resource access validations, CA ACF2 for VM matches the X-RGP record ID with the resource name specified in the $KEY field of the resource rule record.
CA ACF2 for VM matches the value defined in the TYPE field of the X-RGP record with the $TYPE control statement in the resource rule.

To let users access a group of transactions, you need to do the following:

- Ensure that the type codes of the resources are defined in the VMO RESTYPE record.
- Specify the individual transactions that belong in the group.
- Assign the group a resource group name (which is the record ID of the X-RGP record).
- Specify the type code of the resource rule in an X-RGP record.
- Use the X-RGP record ID to write one resource rule for all of the transactions that belong to the resource group. Specify the resource group name (the X-RGP record ID) in the $KEY field of the resource rule record and ensure that you use the same type code for both the resource rule and the X-RGP record.
- Use the ACFSERVE RELOAD RESOURCE RGP command to rebuild the resident typelist for the resource type.
- Use the ACFSERVE RELOAD XREF command to rebuild the cross-reference table.

For more information about resource rules, see the “About Resource Rules” chapter. For information about the CA ACF2 for VM console operator commands, see the *Systems Programmer Guid*e.

Cross-Referencing X-RGP Records

Besides cross-referencing resource rules, X-RGP records can cross-reference each other. This cross-reference function lets you define groups in larger groups. For example, you have three groups of transactions. Some users can access all three groups while others can access only one of the three groups. You need to define each of the three individual groups of transactions and one larger group that consists of all three individual groups to CA ACF2 for VM.

To do this, define a total of four X-RGP records, one for each individual group of transactions, and one for the larger group that consists of all three of the individual transaction groups. Use the names of the transaction groups for the record IDs of the X-RGP records. You will have X-RGP records with record IDs such as TRANA, TRANB, TRANC, and GROUP1, where GROUP1 consists of TRANA, TRANB, and TRANC.

After you define all of the X-RGP records, write one resource rule for each X-RGP record, and use the record IDs (TRANA, TRANB, TRANC, GROUP1) for the $KEY resource names. You also must ensure that the type codes you specify in the X-RGP records match the type codes in the corresponding resource rule records.

You can also have a set of X-RGP records cross-reference another set of X-RGP records. Extending the scenario just described, suppose that you want some users to have access to the set of transaction groups indicated in the X-RGP GROUP1 record and another set of transaction groups defined in the X-RGP GROUP2 record. In this case, you define an X-RGP record called GROUPA. GROUPA consists of GROUP1 and GROUP2. You can use this X-RGP to X-RGP cross-reference ability as an indexing process to define as many as 25 levels.

How CA ACF2 for VM Processes Requests for Resource Access

When you use X-RGP records, CA ACF2 for VM searches the resident typelist first. If you have a $KEY(********) rule that allows access to all transactions, CA ACF2 for VM uses this rule to allow access before it finds the more specific rule. X-RGP processing is never performed. If there is no resident typelist for this resource, X-RGP processing is never performed.

If CA ACF2 for VM finds a matching entry in the resident typelist, it locates the resource rule and performs access validation. CA ACF2 for VM uses the validation result even if the access is denied. If CA ACF2 for VM cannot find a matching entry in the typelist, it searches all the X-RGP records. When CA ACF2 for VM finds a matching entry in the INCLUDE or EXCLUDE lists of an X-RGP record, it performs access validation. If the access is allowed, X-RGP processing ends. If access is denied, CA ACF2 for VM continues to search more X-RGP records until it finds another matching entry in a INCLUDE or EXCLUDE list, looking for a record that allows access. If CA ACF2 for VM cannot find an X-RGP record that allows access, access is denied.

When you use the resource grouping facility to implement resource access controls, you must carefully consider the way you want the accesses authorized and logged.

- X-RGP records let you use masking.
- X-RGP records let you exclude transactions from a resource group.

X-RGP Record Fields

Only the size of the X-RGP record limits the combined field lengths. You can define the following fields in an X-RGP record:

Record ID	Fields
grpname	RESOURCE|GROUP INCLUDE(entry1,...,entryn) EXCLUDE(entry1,...,entryn) TYPE(code)

The fields for resource group records are described below:

grpname

Specifies a 1- to 24-character name of the resource group.

RESOURCE|GROUP

Specifies the function of the record. RESOURCE indicates that the record defines a group of resources. The default value is RESOURCE.

GROUP indicates that the record defines a group of resource groups. Specify GROUP only to define an X-RGP record that cross-references other X-RGP records. See the Cross-Referencing X-SGP Records section for an example.

INCLUDE(entry1,...,entryn)

Specifies the resources or group of resource groups that you want to include in the record. This field accepts a maximum of 40 characters per entry. Use a command or a blank to separate each entry.

If you specify RESOURCE, the INCLUDE field specifies the resource names that belong in this resource group. For resource names, this field accepts a maximum of 40 characters per entry. Separate entries using a comma or a blank.

If you specify GROUP, the INCLUDE field specifies the record IDs of the X-RGP records that this X-RGP record cross-references. For record ID entries, this field accepts a maximum of 24 characters per entry. Separate entries using a comma or a blank.

You can specify resource names or record IDs; you cannot specify both in the same X-RGP record. That is, to use both the RESOURCE and the GROUP functions, you need to define a separate X-RGP record for each and define a unique record ID for each record.

You can mask resource or record ID entries. Only the size of the record limits the field; that is, the amount of storage that is dedicated to the record determines the maximum size of the INCLUDE field.

EXCLUDE(entry1,...,entryn)

Specifies the resources or group of resource groups that you want to exclude from the record.

If you specify RESOURCE, the EXCLUDE field specifies the individual resources that are excluded from the group the INCLUDE field defines. For resource entries, this field accepts a maximum of 40 characters per entry. Separate entries using a comma or a blank.

If you specify GROUP, the EXCLUDE field specifies the record IDs to be excluded from the set of X-RGP records that this X-RGP record cross-references. For record ID entries, this field accepts a maximum of 24 characters per entry. Separate entries using a comma or a blank.

You can specify resources or record IDs; you cannot specify both in the same X-RGP record. That is, to use both the RESOURCE and the GROUP functions, you need to define a separate X-RGP record for each and define a unique record ID for each record. You can mask resource or record ID entries.

You should use the EXCLUDE field only when you mask the INCLUDE field. For example, you have 20 transactions whose names all begin with TR1 and you want to include all but two of these transactions in our resource group. Specify TR1- in the INCLUDE field, and specify the full transaction names of the two transactions you want to exclude TR13 and TR17 in the EXCLUDE field.

When you define this field, remember that CA ACF2 for VM performs sort processing from the most specific to the most general. Some examples will clarify this concept. If you specify GRAF***5 in the INCLUDE field and GRAF0105 in the EXCLUDE field, the terminal identified as GRAF0105 is excluded because it more specifically matches the item in the EXCLUDE list.

If you specify GRAF01- in the INCLUDE field and GRAF0101 in the EXCLUDE field, the terminal identified as GRAF0101 is excluded because it more specifically matches the item in the EXCLUDE list.

If you specify GRAF0103 in the INCLUDE field and GRAF010* in the EXCLUDE field, the terminal identified as GRAF0103 is included because it most specifically matches the item in the INCLUDE list.

If the INCLUDE field and the EXCLUDE field have an identical entry, the entry is excluded from the group. For example, if the security administrator includes GRAF0120 and GRAF0121 and excludes GRAF0120, GRAF0120 is excluded.

TYPE

Specifies the type code for the resource rule that the X-RGP record cross-references. You can only use the TYPE parameter when you specify RESOURCE. You cannot use it if you specify GROUP. You can use the standard type codes that are described in the chapter "About Resource Rules." These type codes must be stored in the resident typelist before you can create the resource group record.

How CA ACF2 for VM Sorts X-RGP Records

CA ACF2 for VM performs sort processing from the most specific to the most general. For example, if you specify GRAF***5 in the INCLUDE field and GRAF0105 in the EXCLUDE field, the terminal identified as GRAF0105 is excluded because it more specifically matches the item in the EXCLUDE list.

If you specify GRAF01- in the INCLUDE field and GRAF0101 in the EXCLUDE field, the terminal identified as GRAF0101 is excluded because it more specifically matches the item in the EXCLUDE list.

If you specify GRAF0103 in the INCLUDE field and GRAF010* in the EXCLUDE field, the terminal identified as GRAF0103 is included because it most specifically matches the item in the INCLUDE list.

If the INCLUDE field and the EXCLUDE field have an identical entry, the entry is excluded from the group. For example, if the security officer includes GRAF0120 and GRAF0121 and excludes GRAF0120, GRAF0120 is excluded.

Processing XREF Records

Users with the CA ACF2 for VM Security privilege can issue ACF subcommands to create, change, and display XREF records.

Setting XREF Record Mode

To maintain XREF infostorage records, establish the appropriate ACF command setting and type code. The syntax is:

```
Set|T    [Xref(Sgp|Rgp)]
         [SYSid(sysid)|DIVision(div)      ]
         [MSYSid(sysidmask)|MDIV(divmask)]
```

The parameters for the SET subcommand are described below:

XREF(Sgp|Rgp)

Specifies the type of record you want to process: SGP (source group records) or RGP (resource group records).

SYSid(sysid)|DIVision(div)

Specifies the system ID record to apply to. Enter the one- to eight-character SYSID name. Masking does not apply. If you specify an asterisk (*), CA ACF2 for VM considers it part of the SYSID name. To process records for several system IDs, use the MSYSID parameter.

You can also specify the SYSID that you want an XREF record to apply to using the INSERT, CHANGE, LIST, and DELETE subcommands. DIVISION is a synonym for SYSID.

MSYSid(sysidmask)|MDIV(divmask)

Specifies a group of SYSIDs that you want this record to apply to. MDIV is a synonym for MSYSID.

INSERT Subcommand

The INSERT subcommand lets you create an XREF record for X-SGP or X-RGP records, depending on the type you specified in the SET command.

The INSERT subcommand has the following syntax:

```
        {*                    }
Insert  {recid Type(typelist)          }
         USING(modelrecid) newrecid
         Type(typelist)
        [SYSid(?|sysid)|DIVision(?|div)]
        [USYSid(?|sysid)|UDIVision(?|div)]
        [Source|Group ]
        [Resource|Group]
        [Include(entry1,...,entryn)]
        [Exclude(entry1,...,entryn)]
        [ADD|REP|DEL]
```

The INSERT subcommand accepts the following parameters:

*** (asterisk)**

Specifies the last record you processed while in this ACF command setting. The asterisk specifies only one record. You cannot use the asterisk to represent a record mask.

recid

Specifies the name of the XREF record you want to create. You cannot use masking characters to specify a group of records. Record IDs for source group records can be from one to eight characters. Record IDs for resource group records can be from 1 to 24 characters.

Type(typelist)

Specifies the type codes of the resources in the INCLUDE and EXCLUDE lists. You can mask this field. This field applies to X-RGP resource records only. It does not apply to X-SGP group records. The ADD, REP, and DEL parameters apply to the types you specify.

The type codes you specify must be defined in a resident typelist before you process the resource group record.

USING(oldrecid) newrecid

Identifies a model XREF record that you want to use to create a new XREF record. All values from the model record are inserted into the new record. Any other fields and values you specify add to or replace the fields and values in the new record.

If you specify the USYSID parameter, the USING parameter is not required. CA ACF2 for VM assumes that you want to copy the record for the SYSID specified in the USYSID parameter, but with the same record identification as the record you are inserting.

SYSid(?|sysid)|DIVision(?|div)

Specifies the one- to eight-character SYSID that you want this to indicate that you want the record to apply to the current active SYSID.

If you do not specify a SYSID (or USYSID), CA ACF2 for VM uses the SYSID value you specified when you establish the ACF command setting. If you did not supply a SYSID when you established the ACF command setting,

CA ACF2 for VM uses the default value defined at CA ACF2 for VM startup time. For further information, see the “Processing Shift and Zone Records” chapter. DIVISION is a synonym for SYSID.

USYSid(?|sysid)|UDIV(?|div)

Specifies the SYSID of a model XREF record you want to use to create a new XREF record. Specify this record ID with the a SYSID, the default SYSID is used. UDIV is a synonym for USYSID.

This time-saving feature lets you insert a record that is similar to a record previously defined for another SYSID. To change any fields and values that differ between the model and the new record, use the ADD, REP, or DEL parameters (described below).

Source|Group

Specify SOURCE to create a source group record. Specify GROUP to create a group of source group records. These parameters apply to the X-SGP setting only.

Resource|Group

Specify RESOURCE to create a resource group record. Specify GROUP to create a group of resource group records. These parameters apply to the X-RGP setting only.

Include(entry1,...,entryn)

Specifies a group of sources, source groups, resources, or resource groups that you want this record to apply to. The ADD, REP, and DEL parameters apply to the names you specify in this field.

Exclude(entry1,...,entryn)

Specifies a group of sources, source groups, resources, or resource groups that you do not want this record to apply to. Specify names here only if you specify a mask in the INCLUDE parameter. The ADD, REP, and DEL parameters apply to the names you specify in this field.

ADD

Adds the specified sources or resources to those copied from the model record. If a value already exists in a record field, the new value replaces the copied value. This parameter applies to multivalued fields only. The new value is added to the existing values. ADD is the default.

REP

Replaces the specified sources or resources in the model record with those you specify. If any source or resource is not part of the model record, CA ACF2 for VM adds that source or resource to the record. This parameter applies to multivalued fields only.

DEL

Deletes the specified source or resource from the newly inserted record. If a field can contain multiple values, then CA ACF2 for VM deletes just the values you specify from the field. This parameter applies to multivalued fields only.

CHANGE Subcommand

The CHANGE subcommand lets you change an existing XREF record. The CHANGE subcommand has the following syntax:

```
        {*                      }
CHange  {recid                  }
        {LIKE(recidmask)        }
        [Type(typelist)]
        [SYSid(?|sysid)|DIVision(?|div)     ]
        [MSYSid(?|sysidmask)|MDIV(?|divmask)]
        [Source|Group ]
        [Resource|Group]
        [Include(entry1,...,entryn)]
        [Exclude(entry1,...,entryn)]
        [ADD|REP|DEL]
```

The CHANGE subcommand takes the following parameters:

*** (asterisk)**

Specifies that you want to change the last record you processed since you established this ACF command setting. (The asterisk does not work in the case of multiple records or masking.)

recid

Specifies the name of the XREF record that you want to change. Record IDs for source group records can be from one to eight characters. Record IDs for resource group records can be from 1 to 24 characters.

Type(typelist)

Specifies a list of type codes of the resources in the INCLUDE and EXCLUDE lists. You can mask this field. This field applies to X-RGP resource records only. It does not apply to X-RGP group records. The ADD, REP, and DEL parameters apply to the type codes you specify.

You must define the type codes you specify in a resident typelist before you process the resource group record.

LIKE(recidmask)

Specifies a mask for the XREF records that you want to change. You cannot abbreviate the LIKE parameter. This masking follows the same conventions that apply to logonids, as described in the chapter “About the Logonid Record” in this guide.

SYSid(?|sysid)|DIVision(?|div)

Specifies the one- to eight-character SYSID that you want this to indicate that you want the record to apply to the current active SYSID.

If you do not specify a SYSID (or MSYSID), CA ACF2 for VM uses the SYSID value you specified when you established the ACF command setting. If you did not supply a SYSID when you established the ACF command setting,

CA ACF2 for VM uses the default value defined at CA ACF2 for VM startup time. For further information, see the section on the concept of SYSID. DIVISION is a synonym for SYSID.

MSYSid(?|sysidmask)|MDIV?|divmask)

Specifies a mask to indicate the SYSIDs that you want this changed record to apply to. This SYSID mask must contain at least one asterisk or a trailing dash. Structured infostorage records that you change using this parameter have the same record ID, but different SYSIDs. MDIV is a synonym for MSYSID.

Resource|Group

Specify RESOURCE to create a resource record. Specify GROUP to create a resource group record. These parameters apply to the X-RGP setting only.

Source|Group

Specify SOURCE to change a source record. Specify GROUP to change a source group record. These parameters apply to the X-SGP setting only.

Include(entry1,...,entryn)

Specifies a group of sources, source groups, resources, or resource groups that you want this record to apply to. The ADD, REP, and DEL parameters apply to the names you specify in this field.

Exclude(entry1,...,entryn)

Specifies a group of sources, source groups, resources, or resource groups that you do not want this record to apply to. Specify names here only if you specify a mask in the INCLUDE parameter. The ADD, REP, and DEL parameters apply to the names you specify in this field.

ADD

Adds the specified sources or resources to those in the existing record. If a value already exists in a record field, the new value replaces the copied value. This parameter applies to multivalued fields only. The new value is added to the existing values. ADD is the default.

REP

Replaces the sources and resources in an existing record with the fields and values you specify. If any field you specify is not part of the existing record, CA ACF2 for VM adds that field and its value to the record. This parameter applies to multivalued fields only.

DEL

Deletes the specified sources and resources from the existing record. This parameter applies to multivalued fields only. CA ACF2 for VM deletes the values you specify from the field.

LIST Subcommand

The LIST subcommand displays the contents of an XREF infostorage record. The LIST subcommand has the following syntax:

```
     {*       }
List {recid     }
     {LIKE(recidmask)}
     [SYSid(?|sysid)|DIVision(?|div)   ]
     [MSYSid(?|sysidmask)|MDIV(?|divmask)]
     [Source|Group ]
     [Resource|Group]
```

The LIST subcommand takes the following parameters:

*** (asterisk)**

Displays the last record you processed since you established this ACF command setting. You cannot use the asterisk to display more than one record. Use the LIKE parameter.

recid

Specifies a name XREF record that you want to display. Record IDs for source group records can be from one to eight characters. Record IDs for resource group records can be from 1 to 24 characters.

LIKE(recidmask)

Displays a group of structured infostorage records. You cannot abbreviate the LIKE parameter. This masking follows the same conventions that apply to logonids, as described in the chapter “About the Logonid Record” in this guide.

SYSid(?|sysid)|DIVision(?|div)

Specifies the one- to eight-character SYSID of the record that you to indicate that you want the record to apply to the current active SYSID.

If you do not specify a SYSID (or MSYSID), CA ACF2 for VM uses the SYSID value you specified when you established the ACF command setting. If you did not supply a SYSID when you established the ACF command setting,

CA ACF2 for VM uses the default value defined at CA ACF2 for VM startup time. For further information, see the chapter “Defining Structured Infostorage Records” chapter. DIVISION is a synonym for SYSID.

MSYSid(?|sysidmask)|MDIV(?|divmask)

Specifies a mask to indicate the SYSIDs of the record that you want to display. This SYSID mask must contain at least one asterisk or a trailing dash. Structured infostorage records that you display using this parameter have the same record ID, but different SYSIDs. MDIV is a synonym for MSYSID.

Source|Group

Specify SOURCE to view a source record. Specify GROUP to view a source group record. These parameters apply to the X-SGP setting only.

Resource|Group

Specify RESOURCE to view a resource record. Specify GROUP to view a resource group record. These parameters apply to the X-RGP setting only.

DELETE Subcommand

The DELETE subcommand lets you delete a structured infostorage record. The DELETE subcommand has the following syntax:

```
        {*              }
DELete  {recid          }
        {LIKE(recidmask)}
        [SYSid(?|sysid)|DIVision(?|div)     ]
        [MSYSid(?|sysidmask)|MDIV(?|divmask)]
        [Source|Group ]
        [Resource|Group]
```

The DELETE subcommand takes the following parameters:

*** (asterisk)**

Deletes the last record you processed since you established this ACF command setting. You cannot use the asterisk to delete multiple records. Use the LIKE parameter.

recid

Specifies the name of the XREF record that you want to delete. Record IDs for source group records are from one to eight characters. Record IDs for resource group records are from 1 to 24 characters.

LIKE(recidmask)

Specifies a group of XREF records that you want to delete. This masking follows the same conventions that apply to logonids, as described in the chapter “About the Logonid Record” in this guide. Use extreme care when deleting multiple records with this parameter.

When you issue an online command that deletes multiple records from the CA ACF2 for VM databases, CA ACF2 for VM prompts you to confirm the DELETE LIKE request. If you respond Y, the command executes. If you respond N or anything else, CA ACF2 for VM ignores the command and prompts for the next command.

SYSID(?|sysid)|DIVision(?|div)

Specifies the one- to eight-character SYSID of the record that you to delete the record from the currently active SYSID.

If you do not specify a SYSID (or MSYSID), CA ACF2 for VM uses the SYSID value you specified when you established the ACF command setting. If you did not supply a SYSID when you established the ACF command setting,

CA ACF2 for VM uses the default value defined at CA ACF2 for VM startup time. For further information, see the section on the concept of SYSID. DIVISION is a synonym for SYSID.

MSYSid(?|sysidmask)|MDIV(?|divmask)

Specifies a mask to indicate the SYSIDs of the record that you want to delete. This SYSID mask must contain at least one asterisk or a trailing dash. The XREF records that you delete using this parameter have the same record ID, but different SYSIDs. MDIV is a synonym for MSYSID.

Source|Group

Specify SOURCE to delete a source record. Specify GROUP to view a source group record. These parameters apply to the X-SGP setting only.

Resource|Group

Specify RESOURCE to delete a resource record. Specify GROUP to delete a resource group record. These parameters apply to the X-RGP setting only.

X-SGP Record Samples

Suppose a site has terminals that are defined as LVxx where xx is a number from 00 to 99. The security administrator wants to ensure that users access a data set only from terminals LV20 through LV30 with the exception of terminal LV25. No one should use LV25 to access the data set. The security administrator needs to insert an X-SGP record as follows:

```
acf
 ACF
set x(sgp)
 XREF
insert lterms source include(lv2*,lv30) exclude(lv2,lv25)
```

LTERMS

Is the record name. The record ID is the name that the security administrator assigns to this group of terminals.

LV2*

Is the masked ID that matches any existing terminal ID that begins with the characters LV2.

If this site also has terminals defined as LV2A, LV2B, LV2X and so forth, and the security administrator wants to restrict access, the situation is more complicated because LV2A, LV2B, and LV2X match the criteria specified in the above insert example. In this case, the only way to accomplish inclusion of the proper terminals is to specify the following:

```
acf
 ACF
set x(sgp)
 XREF
insert lterms source include(lv20,lv21,lv22,lv23,lv24,lv26,lv27,lv28,lv29,lv30)
```

Now suppose that the security administrator creates an X-SGP record for a group of terminals and gives it the record ID LTERMS. And he creates another X-SGP record for a different group of terminals called DTERMS. To let some users have access to both LTERMS and DTERMS terminals, he can insert an X-SGP record as follows:

```
acf
 ACF
set x(sgp)
 XREF
insert groupa group include(lterms,dterms)
```

GROUPA

Is the record name of this X-SGP record, which is the name of the set of X-SGP records to which this X-SGP record is cross-referenced

LTERMS

Is the record name of the individual X-SGP records that belong to this set of X-SGP records identified as GROUPA

DTERMS

Is the record name of the individual X-SGP records that belong to this set of X-SGP records identified as GROUPA.

The security administrator must specify LTERMS in the SOURCE field of the data set access rule entry to refer to users who access the data set from the LTERMS group of terminals and DTERMS in the SOURCE field of the data set access rule entry to refer to users who can access the data set from the DTERMS group of terminals. Also, the security administrator must specify GROUPA in the SOURCE field of the data set access rule entry to refer to users who are permitted access to the data set from both terminal groups (LTERMS and DTERMS).

X-RGP Record Samples

Suppose a security administrator wants to grant users DIAL access to user IDs USER20, USER21, USER22, and USER30. The security administrator then inserts an X-RGP record as follows:

```
acf
 ACF
set x(rgp)
 XREF
insert trana resource include(user2-,user30) exclude(user23) type(dia)
```

TRANA

Is the record name that the security administrator selects to assign to this group of user IDs

USER2-

Is the masked ID to which any user ID that begins with the characters USER2 matches

DIA

Is the resource rule type code for DIAL accesses.

The resource rule for this group of transactions might look like the one below:

```
$KEY(TRANA) TYPE(DIA)
 UID(PERCLK) ALLOW
```

Now suppose the security administrator wants to permit some users to access the TRANA group of transactions and another group of transactions that he has identified as TRANB in another X-RGP record. The security administrator inserts an X-RGP record as follows:

```
acf
 ACF
set x(rgp)
 XREF
insert group1 group include(trana,tranb)
```

GROUP1

Is the record name of this X-RGP record that is the name of the set of X-RGP records to which this X-RGP record refers

TRANA

Is the record ID of the individual X-RGP records that belong to this set of X-RGP records identified as GROUP1

TRANB

Is the record ID of the individual X-RGP records that belong to this set of X-RGP records identified as GROUP1.

To follow through with his security plan, the security administrator must be sure to specify TRANA in the $KEY field of the resource rule for the TRANA group of users, TRANB in the $KEY field of the resource rule for the TRANB group of users, and GROUP1 in the $KEY field of the resource rule for the set of user groups (TRANA and TRANB). **DIA must the type code** specified **for all these rules**.

In the previous example, TRANA contains certain user IDs, TRANB contains certain user IDs and GROUP1 includes user IDs from TRANA and TRANB. If someone attempts access to a user ID that is in TRANA (TR20), it is included in GROUP1 because all transactions in TRANA are included in GROUP1.

If a TRANA rule exists that permits a user access, and a GROUP1 rule exists that prevents that user access, validation processing eventually grants access because **some** rule permits access. Access is also granted if the TRANA rule prevents access and the GROUP1 rule grants access.

As an example, assume payroll personnel need access to all the user IDs in both TRANA and TRANB. It is necessary to write only a GROUP1 rule. It is possible that the TRANA and the TRANB rule prevent access for the payroll manager.

```
$KEY(TRANA) TYPE(DIA)
 UID(PERCLK) ALLOW

$KEY(TRANB) TYPE(DIA)
 UID(ACTCLK) ALLOW

$KEY(GROUP1) TYPE(DIA)
 UID(PAYMGR) ALLOW
```

The TRANA rule permits access to personnel clerks. The TRANB rule permits access for accounting clerks. Both rules prevent the payroll manager from access because they do not include his UID. The GROUP1 rule permits his access. Because GROUP1 includes TRANA and TRANB access is granted to the payroll manager even though the TRANA and TRANB rules prohibit his access.

Migration Considerations

This section describes migrating E-SGP records to X-SGP records.

E-SGP Records to X-SGP Records

The ACFESGP conversion utility converts E-SGP records to X-SGP records. Refer to the description of ACFESGP in the *Reports and Utilities Guid*e.

CA ACF2 for VM provides the following support routines for source grouping:

ACF00SGP (also called ACFSGRP)

This routine provides the name of the first-level source group to which a source belongs

ACF00SSL

This routine provides the names of sources that belong to a particular group

ACF00SST

This routine provides all the names of the source groups to which a source belongs.

For more information about these support routines, see the *Systems Programmer Guid*e.

You can use the ACFRGP utility to obtain a list of all the resource groups to which a particular resource belongs. This utility lists the groups in the order that validation occurs. For more information about this utility, see the *Reports and Utilies Guid*e.

Activating XREF Records

To activate a newly created or changed XREF record, issue the ACFSERVE RELOAD XREF command. ACFSERVE RELOAD XREF SGP dynamically updates the source entry cross-reference table. ACFSERVE RELOAD XREF RGP updates the resource cross-reference table. For details, see the “Using the ACFSERVE Commands” chapter.

Chapter 20: Processing Scope Records

Scope records are stored on the Infostorage database with the storage class of “S.” Only one type code is allowed for scope records in storage class SCP.

Scope list records limit the authority of a privileged CA ACF2 for VM user who has one or more of the following CA ACF2 for VM privileges in his logonid record:

- ACCOUNT
- AUDIT
- CONSULT
- LEADER
- SECURITY

A privileged user whose logonid record is defined with a SCPLIST attribute can access only the rules, files, and records defined in the SCPLIST field of that attribute. In earlier releases of CA ACF2 for VM, you learned about four different kinds of scope list logonid attributes: DSNSCOPE, LIDSCOPE, UIDSCOPE, and SCPLIST. We recommend that you use SCPLIST because we will not support DSNSCOPE, LIDSCOPE, and UIDSCOPE in future releases.

This chapter provides information on:

- Creating scope lists, limiting privileged users' authority
- Creating data set, logonid, and UID scopes that limit who a privileged user can write rules for
- Setting the mode for processing scope records
- Creating, changing, displaying, deleting, and processing scope records.

This section contains the following topics:

SCPLIST: The Logonid Record Scope List Field

To apply a scope record to a privileged user, include the name of the scope list in the user's logonid.

```
CHANGE TLCAMS SCPLIST(PAYROLL)
```

Under the LID or ACF setting, this CHANGE subcommand applies the scope record PAYROLL to the logonid record TLCAMS. PAYROLL is specified in the user's SCPLIST logonid attribute.

If you specify SCPLIST in the logonid record but the name entered in parentheses does not point to a valid scope record on the Infostorage database, the user has no scope of authority (the user cannot change or delete any record type).

Setting Scope Entry Record Mode

Use the ACF subcommands to create, change, and display scope lists much like other records on the Infostorage database. Process all scope records under the SCOPE setting of the ACF command. The required type code for scope lists is SCP. The syntax for the SET subcommand is:

```
SEt SCope(SCP)
```

SCOPE(SCP)

Specifies that you want to process SCPLIST records.

After you establish the SCOPE setting, you can use other ACF subcommands to process your scope records.

Creating Scope Records

Use the INSERT subcommand under the SCP setting to create scope records. The syntax for this INSERT subcommand is:

```
INsert  scpname
        [USing(scpname)]
        {ADD|REPlace|DELete}

   { [  Dsn(entry1,entry2,...,entryn)  ] }
   { [  Inf(entry1,entry2,...,entryn)  ] }
   { [  Lid(entry1,entry2,...,entryn)  ] }
   { [  Uid(entry1,entry2,...,entryn)  ] }
```

USing(scpname)

Specifies that you want to use an existing scope record as a model for the new scope record being inserted.

scpname

This is the one- to eight-character name defined in the SCPLIST field of a user's logonid record.

ADD|REPlace|DELete

These subfunctions modify the new record from the model record. ADD is the default.

There are four different types of fields. You must include at least one in a scope record through the INSERT subcommand:

Dsn(entry1,entry2,...,entryn)

These are one- to eight-character access rule scope fields specifying the VM user ID ($KEY value) of a CMS file or minidisk. It can also be a high-level index of an OS data set or DOS file. You can mask this field.

Inf(entry1,entry2,...,entryn)

These 1- to 44-character Infostorage database scope fields place the matching Infostorage database records in the scope of the user.

Lid(entry1,entry2,...,entryn))

These are one- to eight-character logonid record scope fields to put the logonid or logonid mask in the scope of the user. You must also specify a UID field for authorization to occur. Access to logonid records is not authorized unless you specify both the LID and UID parameters through the INSERT subcommand. When creating scope records, you must understand CA ACF2 for VM masking conventions (the differences between using the asterisk and using the dash).

Uid(entry1,entry2,...,entryn))

These entries represent the UIDs or UID masks extending from 1-to 24- characters that are put in the scope of the user. Access to logonid records is not authorized unless you specify both the LID and UID parameters.

You can specify multiple entries in DSN, INF, and LID fields. Separate multiple entries by commas or blanks. Separate UID field entries with commas only.

Masking for the SCPLIST UID field works in the same way as for the UID field of rules. Specifically, the UID field is automatically padded out to the right with masking characters (up to 24 characters), although this is transparent to the user.

Scope records grant no special privileges. They limit the records where user's privileges apply to. restrict other special privileges granted to the user (SECURITY, ACCOUNT, AUDIT). Because only security administrators can create, change, or delete Infostorage records, a user without the SECURITY privilege could not create a new resource rule record, regardless of any matching INF (Infostorage) scope field entries.

An example of using the INSERT subcommand under the SCOPE(SCP) setting shows how you can define scope record PAYSCOPE to govern the scope of a security administrator in a payroll department:

```
INSERT PAYSCOPE DSN(PAYWORK,PAYTEST) -
               LID(PAY-) -
               UID(FINMGR-)
```

When you assign the scope list PAYSCOPE to the security administrator's logonid SCPLIST field, the security administrator is limited to:

- Writing access rules that govern the two groups of Payroll work files (VM user IDs PAYWORK and PAYTEST).
- Creating and updating logonid records that match the logonid mask PAY (all Payroll employees) and the UID mask FINMGR. You must specify the LID and UID fields of this scope record and the logonid records must match an entry in both fields to let the user create and update those records. If you do not specify either field, by default, no records match.

Changing Scope Records

Use the CHANGE subcommand under the SCP setting to modify specific entries in particular fields of a single scope record or a group of scope records. The syntax for this CHANGE subcommand is:

```
        { *             }
CHange  { recname       }
        { LIKE(recmask) }   {ADD|DELete|REPlace}

      { [ Dsn(entry1,entry2,...,entryn)   ] }
      { [ Inf(entry1,entry2,...,entryn)   ] }
      { [ Lid(entry1,entry2,...,entryn)   ] }
      { [ Uid(entry1,entry2,...,entryn)   ] }
```

*****

Specifies that you want to change the last scope record processed since you established the current SCOPE setting.

recordname

Specifies that you want to change the one- to eight-character name of an individual scope record.

LIKE(recmask)

Specifies that you want to change a one- to eight-character mask for the names of scope records. Define this scope record name in the SCPLIST field of the user's logonid record.

ADD|DELete|REPlace

These subfunctions modify the record.

ADD

Adds the specified entries to the indicated fields of the record. This is the default.

DEL

Deletes the specified entries from the indicated fields.

REP

Replaces all entries in the indicated field with the specified entries.

There are four different types of fields that you can modify in a scope record with the CHANGE subcommand.

Displaying Scope Records

Use the LIST subcommand under the SCP setting to list a scope record or a group of scope records that match a record mask. The syntax for this LIST subcommand is:

```
      { *               }
List  { scopename       }
      { LIKE(scopemask) } {ALL,DSN,INF,LID,UID\}
```

*****

Displays the last scope record processed since you established the current SCOPE setting.

scopename

Specifies the one- to eight-character name of an individual scope record.

LIKE(scopemask)

Specifies a one- to eight-character mask for the names of scope records. The LIKE operand with a mask lists multiple records that match the mask.

ALL,DSN,INF,LID,UID

Display the fields in the scope record.

ALL

Displays all fields of a scope record. ALL is the default.

DSN

Displays the DSN field of a scope record

INF

Displays the INF field of a scope record

LID

Displays the LID field of a scope record

UID

Displays the UID field of a scope record.

Deleting Scope Records

Use the DELETE subcommand under the SCP setting to delete a record name or all entries that match a record mask. The syntax for this DELETE subcommand is:

```
       { *             }
DELete { recordname    }
       { LIKE(recmask) }
```

*****

Deletes the last scope record processed since you established the current SCOPE setting.

recordname

Specifies the one- to eight-character name of an individual scope record.

LIKE(recmask)

Specifies a one- to eight-character mask for the names of scope records.

Ending Scope Processing

When you have finished processing scope records, enter the following command:

```
END
```

This exits you from the ACF command setting.

Chapter 21: Processing Shift and Zone Records

Shift and zone records let you restrict system access even further. Shift records define the time-of-day and date when a user can access the system. Zone records define the local time zone where the user accesses the system. CA ACF2 for VM stores shift and zone records on the Infostorage database with the storage class value of “T.”

This chapter provides information on:

- Shift and zone records
- How to set the mode to maintain shift records
- How to create, display, change, and delete shift and zone records
- Special shift records, such as days of the week.

This section contains the following topics:

Types of Shift Records

There are two different type codes in the storage class “T,” SFT for Shift records, and ZON for Zone records. These records control access according to time-of-day and date.

Shift records (SFT)

The shift record has a type code of SFT. In a user's logonid record, the value of the SHIFT field indicates the shift record that applies to that user. You can also reference this record in access rules and resource rules to control data access by time-of-day and day-of-week.

A shift record name is one-to eight-characters long. The fields in this record can specify:

- Days of the week (two characters, such as MO and TU)
- Dates (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending on the DATE operand of the OPTS VMO record
- Time (hhmm, hours and minutes, in five minute intervals using a 24-hour clock)

You can specify the LOGSHIFT field in a user's logonid record, letting the user access the system outside of his specified shift period. All such off-shift system accesses are logged. These loggings appear in the Invalid Password/Authority Log (ACFRPTPW).

Zone Records (ZON)

The zone record has a type code of ZON. It contains one field that specifies that particular time zone's offset (in hours and minutes) from CPU time. In a user's logonid record, the value of the ZONE field indicates the zone record that applies to that user. zone record names are one to three characters.

Setting Shift Entry Record Mode

You can create, change, and display a shift or zone record through the ACF subcommands, similar to other records on the Infostorage database. All processing of shift and zone records is done under the SHIFT setting of the ACF command. The required type code for processing shift records is SFT. The required type code for zone records is ZON. The syntax for the SET SHIFT|ZONE subcommand is:

```
SEt   { SHIFT(SFT)  }
      { SHIFT(ZON)  }
```

SHIFT(SFT)

Processes shift records.

SHIFT(ZON)

Processes zone records.

The last SET SHIFT setting determines the type of shift record that is processed with the other ACF subcommands. After you establish the SHIFT setting, you can use other ACF subcommands to process shift and zone records.

Creating Shift Records

The CHANGE, DELETE, INSERT, and LIST subcommands let you maintain shift records on the system. Whenever you insert a new shift record or change an existing record, you must issue the ACFSERVE RELOAD SHIFT command to activate the record. The INSERT subcommand under the SFT setting creates new shift records. The syntax for the INSERT subcommand is:

```
INsert  { *          }
        { recordname }

        { [DAYS(MO,TU,WE,TH,FR,SA,SU) -      }
        { (mm/dd/yy,...,mm/dd/yy))]          }
        { [NDAYS(MO,TU,WE,TH,FR,SA,SU) -     }
        { (mm/dd/yy,...,mm/dd/yy))]          }

        [ TIME(hhmm-hhmm,...,hhmm-hhmm)      ]
        [ NTIME(hhmm-hhmm,...,hhmm-hhmm)     ]
        [ INCLUDE(recordname,...,recordname) ]
```

*****

Inserts the last shift record processed since you established the setting.

recordname

Specifies a one- to eight-character name of the shift record.

DAYS

Specified as a two-character abbreviation (such as MO for Monday) or in the actual date format (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending on the DATE operand of the OPTS VMO record. These are the days you want to allow system or resource access. You must specify DAYS if you use NDAYS.

NDAYS

Specified in the same way as DAYS (two-character day or actual dates). Specifies the days and dates to exclude from the DAYS parameter. If you specify NDAYS, but no value for DAYS, you create a shift record that does not permit any access on any day.

TIME

Specified in hours and minutes, based on a 24-hour clock (such as 0900-1700). This is the time period when accesses are allowed. You can specify multiple entries. You should specify the TIME parameter because it defaults to no allowable time period.

NTIME

Specified the same way as TIME (24-hour time in hours and minutes). NTIME specifies the time period to exclude from the TIME parameter. To use NTIME, you must specify a TIME parameter. CA ACF2 for VM denies access if you do not specify a TIME parameter.

INCLUDE

Specifies the record name of any existing shift record to include as part of the record you are defining.

All shift records must contain a DAYS or NDAYS parameter.
CA ACF2 for VM does not automatically grant access for days, date, and times excluded by the NDAYS and NTIME parameters. You must specify the DAYS or TIME you want to allow.

Creating Zone Records

The INSERT subcommand under the ZON setting creates zone records. Whenever you insert a new zone record or change an existing record, you must issue the ACFSERVE RELOAD SHIFT command to activate the record. The syntax for this INSERT subcommand is:

```
INsert { *    }
       { zonename } ADJUST(+hhmm|-hhmm)
```

*****

Inserts the last shift record processed since you established the SHIFT(ZON) setting.

zonename

Inserts a one- to three-character name.

ADJUST(+hhmm|-hhmm)

A negative or positive value (in hours and minutes) to be added or subtracted from the processing CPU time. For example, a processing CPU in Chicago might specify the following zones for Eastern time (one hour ahead of Chicago local time) and Pacific time (two hours behind), in this way:

```
INSERT EST ADJUST(+0100)
INSERT PST ADJUST(-0200)
```

Changing Shift Records

The CHANGE subcommand under the SFT setting modifies shift records or groups of shift records. The syntax for this CHANGE subcommand is:

```
        { *             }
CHange  { recordname    }
        { LIKE(recmask) }

        { [DAYS(MO,TU,WE,TH,FR,SA,SU)        }
        { (mm/dd/yy,...,mm/dd/yy)]           }
        { [NDAYS(MO,TU,WE,TH,FR,SA,SU)       }
        { (mm/dd/yy,...,mm/dd/yy)]}          }

        [ ADD|DEL|REP                        ]
        [ TIME(hhmm-hhmm,...,hhmm-hhmm)      ]
        [ NTIME(hhmm-hhmm,...,hhmm-hhmm)     ]
        [ INCLUDE(recordname,...,recordname) ]
```

*****

Changes the last shift record processed since you established the current setting.

recordname

Specifies a one- to eight-character field in the shift record to change.

LIKE(recmask)

Changes a record in the names of shift records that match this mask.

ADD|DEL|REP

Specifies the new information you should add, delete, or replace in existing shift record fields. ADD is the default.

DAYS

Specified as a two-character abbreviation (such as MO for Monday) or in the actual date format (mm/dd/yy, dd/mm/yy, or yy/mm/dd), depending on the DATE operand of the OPTS VMO record. These are the days you want to allow system or resource access. You must specify DAYS if you use NDAYS.

NDAYS

Specified in the same way as DAYS (two-character day or actual dates). Specifies the days and dates to exclude from the DAYS parameter. If you specify NDAYS, but no value for DAYS, you create a shift record that does not permit any access on any days.

TIME

Specified in hours and minutes, based on a 24-hour clock (such as 0900-1700). This is the time period when accesses are allowed. You can specify multiple entries. You should specify the TIME parameter because it defaults to no allowable time period.

NTIME

Specified in the same way as TIME (24-hour time in hours and minutes). Specifies the time period to exclude from the TIME parameter. To use NTIME, you must specify a TIME parameter. Access is denied if you do not specify the TIME parameter.

INCLUDE

Specifies the record name of any existing shift record to include as part of the record you are defining.

If you do not specify REPLACE or DELETE, CA ACF2 for VM creates a separate new shift record.

Changing Zone Records

The CHANGE subcommand under the ZON setting modifies zone records or groups of zone records. The syntax for this CHANGE subcommand is:

```
        { *              }
CHange  { recordname     } ADJUST(+hhmm|-hhmm)
        { LIKE(recmask)} }
```

*****

Changes a field in the last zone record processed since you established the setting.

recordname

Changes a one- to three- character name of the zone record.

LIKE(recmask)

Changes the zone records that match this mask.

ADJUST(+hhmm|-hhmm)

Adds or subtracts the negative or positive value in hours and minutes from the processing CPU time.

Displaying Shift Records

The LIST subcommand under the SFT setting displays shift records or groups of shift records. The syntax for this LIST subcommand is:

```
       [ *              ]
List   [ recordname     ]
       [ LIKE(recmask)  ]
```

*****

Lists the last shift record processed since you established the setting.

recordname

Lists the one- to eight-character name of the shift record.

LIKE(recmask)

Lists the names of shift records that match the mask.

Displaying Zone Records

The LIST subcommand under the ZON setting displays zone records or groups of zone records. The syntax for this LIST subcommand is:

```
       { *             }
List   { recordname    }
       { LIKE(recmask) }
```

*****

Displays the last zone record processed since you established the setting.

recordname

Displays the one- to three-character name of the zone record.

LIKE(recmask)

Displays the names of zone records that match the mask.

Deleting Shift Records

The DELETE subcommand under the SFT setting removes shift records or groups of shift records. The syntax for this DELETE subcommand is:

```
       { *             }
DELete { recordname    }
       { LIKE(recmask) }
```

*****

Deletes the last shift record processed since you established the setting.

recordname

Specifies the one- to three-character name of the shift record to delete.

LIKE(recmask)

Specifies the names of shift records that match a mask.

Deleting Zone Records

The DELETE subcommand under the ZON setting removes zone records or groups of zone records. The syntax for this DELETE subcommand is:

```
        { *              }
DELete  { recordname     }
        { LIKE(recmask)  }
```

*****

Deletes the last zone record processed since you established the setting.

recordname

Specifies the one- to three-character name of the zone record to delete.

LIKE(recmask)

Specifies the names of zone records that match a mask.

Special Shift Record Options

You have a great deal of flexibility with the INSERT subcommand when you create shift records. For example, you can specify DAYS(MO,TU,WE,TH,FR) to allow access on Monday through Friday, but specify NDAYS(12/25/96) to prevent access on holidays, such as Christmas. Even if Christmas falls on a day specified in DAYS, the date specified in NDAYS prevents access.

You can specify multiple time entries with the TIME operand. For example:

```
TIME(0900-1200,1300-1700).
```

These values allow access from 9:00 a.m. through 11:59 a.m. and again from 1:00 p.m. through 4:59 p.m. Because you did not specify 1200-1300, access from 12:00 p.m. through 12:59 p.m. is prevented.

The NTIME operand prevents access during a time specified in the TIME operand. For example, if you entered the following, you could specify NTIME(1200-1300) to prevent access during the noon period:

```
TIME(0900-1700)
```

The INCLUDE parameter specifies a special shift requirement. For example, you can define a normal shift through the INSERT subcommand as follows:

```
INSERT NORMAL DAYS(MO,TU,WE,TH,FR),
       NDAYS(01/01/99,07/04/99,12/25/99),
       TIME(0900-1700),NTIME(1200-1300)
```

This SHIFT record allows access during weekdays from 9:00 a.m. through 4:59 p.m., excludes access during the lunch hour (12:00 p.m. through 12:59 p.m.), and prevents access on three holidays. You can specify a special shift to allow programmers access on Saturday mornings and during the regular work week this way:

```
INSERT SPECL INCLUDE(NORMAL),DAYS(SA),
             TIME(0900-1200)
```

In the above example, the SHIFT (SPECL) includes all specifications of the shift specified as NORMAL, and Saturdays from 9:00 a.m. through 11:59 a.m. The INCLUDE operand prevents access on all holidays for a year. For example, you can insert a record citing all the holidays for the upcoming year:

```
INSERT HOLIDAYS NDAYS(01/01/99,05/25/99,07/04/99,
                      09/07/99,11/26/99,12/25/99)
```

You can include the HOLIDAYS record in other records using the INCLUDE operand of INSERT or you can insert a record that prevents access on all Christmas dates for the next five years:

```
INSERT XMAS NDAYS(12/25/98,12/25/99, 12/25/00,
                  12/25/01,12/25/02)
```

You can include the XMAS record in other records with the INCLUDE operand of INSERT. As you can see, the INCLUDE feature is very powerful and saves you from changing all shift records when you just need to maintain a few days.

Ending Shift and Zone Processing

When you finish processing shift and zone records, enter the following command:

```
END
```

This exits you from the ACF command environment.

Chapter 22: Defining Structured Infostorage Records

This chapter explains structured infostorage records for VM. Structured infostorage records are composed of fields that you can individually list, update, and validate in a specific record on the Infostorage database. (Unstructured infostorage records are composed of lists and you must validate and retrieve the entire record to update individual lists.)

CA ACF2 for VM lets you define certain VM system options through VM Option records (VMO records) and specify who can issue ACFSERVE commands through ACFSERVE privilege records.

This chapter is in two parts, Part I: Defining VM System Options and Part II: Defining User Privileges for ACFSERVE Commands.

Because most of the ACF subcommands under the CONTROL(ACFSERVE) setting function like the ACF subcommands under the CONTROL(VMO) setting, we often refer you back to the relevant section or definition in Part I: Defining VM System Options to avoid repeating information.

This section contains the following topics:

Part I: Defining VM System Options

This section describes VMO records. These records are similar to OS/390 GSO records. You can issue the SET CONTROL(VMO) subcommand of ACF to define VM system options with these VMO records. You can abbreviate this command to a minimum of SET C(V). VMO records define such system options as operating and performance parameters and site exits. If you do not create VMO records for your system, CA ACF2 for VM creates VMO records using default values.

Besides the options VMO records define, the CA ACF2 for VM Field Definition Record (ACFFDR) also defines some system options. These options are rarely modified after you install CA ACF2 for VM. If you make any changes to the options in the ACFFDR you must reassemble the ACFFDR and IPL the system to make them active. You can make changes to the VMO records active dynamically with an ACFSERVE command. You can find complete information on all the ACFFDR macros in the Appendix of the *Installation Guide*.

This section provides:

- An example of a VMO record
- An overview of the VMO records
- An overview of how to put VMO records into effect
- A description of each of the VMO records, including syntax, function, fields, and SHOW command examples
- A description of how to specify SYSID and SYSID.

Example of a VMO Record

The following is an example of a VMO record:

```
ABC1 / PSWD LAST CHANGED BY ADMISO ON 07/01/92-14:52
       MAXTRY(2) MINPSWD(5) PASSLMT(3) PSWDALT
       PSWDFRC WRNDAYS(3)
```

Each VMO record consists of three components, the SYSID (described later in this chapter), the record name, and record fields. In the above example:

ABC1

Represents the SYSID associated with the VMO record.

PSWD

Represents the record name. This record defines system options relating to password controls.

MAXTRY

The data appearing on the second and third lines illustrates the actual record fields and their contents. For example, the content of the MAXTRY field is two.

We explain each field fully in the section that describes the specific record ID. The VMO PSWD section in PSWD Record - Password Maintenance Support section describes the PSWD record fields.

VMO Record Names

CA selects record IDs for VMO records. You can define or modify these record IDs. These predefined record IDs are listed below, with their basic functions.

APPLDEF

Defines the format of structured infostorage application records.

BACKUP

Defines the authority necessary to do a backup, sets the time for automatic backups, and defines who is notified when CA ACF2 for VM takes a backup.

CLASMAP

The CLASMAP record translates eight-character resource classes into three-byte CA ACF2 for VM resource type codes.

CMDLIM

Defines which CP commands are protected and the CA ACF2 for VM mode used during validation.

DBSYNC

Specifies the options that apply to the Database Synchronization Component.

DIAGLIM

Defines which diagnose instructions are protected and the CA ACF2 for VM mode used during validation.

EXITS

Specifies the exits the service machine loads.

LINUX

Defines Linux machines to CA ACF2 for VM.

MAINT

Specifies the logonids used for system maintenance functions.

OPTS

Defines the CA ACF2 for VM system control options available to the system.

PSWD

Defines the user password controls.

RESCLASS

Defines the CA ACF2 for VM resource record types that CA ACF2 for VM uses to validate access to resources.

RESTYPE

Defines the resource record types that have resident resource type lists (directories) CA ACF2 for VM built.

RESWORD

Specifies the words or word prefixes that cannot be used in passwords.

RULEOPTS

Specifies the options that apply to access rule and resource rule maintenance.

SAFDEF

Defines the SAF environment and instructs CA ACF2 for VM how to process a SAF call.

SSFTYPE

Provides a relation of predefined CA Standard Security Facility classes to CA ACF2 for VM resource record types (except RESOURCE).

TAPEVOLS

Defines the tape volumes that CA ACF2 for VM secures through volume-level protection.

WARN

Specifies a warning message for the user when the system is in WARN mode and CA ACF2 for VM detects a violation.

Each VMO record has a unique set of fields. The fields associated with each VMO record are described following the description of the structure of each record.

Applying VMO Records to Your System

The chart below provides a brief overview of how to apply specific VMO record changes to your system.

VMO Record	Do this to activate
APPLDEF	ACFSERVE RELOAD CONTROL VMO
BACKUP	ACFSERVE RELOAD CONTROL VMO
CLASMAP	ACFSERVE RELOAD CONTROL VMO
CMDLIM	IPL the system
DBSYNC	ACFSERVE RELOAD CONTROL VMO
DIAGLIM	IPL the system
EXITS	Restart the service machine
LINUX	ACFSERVE RELOAD CONTROL VMO
MAINT	ACFSERVE RELOAD CONTROL VMO
OPTS	ACFSERVE RELOAD CONTROL VMO For IDLEMN field: IPL the system For IDLEOP field: IPL the system For MODE field: IPL the system For MAXLID field: Restart service machine
PSWD	ACFSERVE RELOAD CONTROL VMO
RESCLASS	ACFSERVE RELOAD CONTROL VMO
RESTYPE	Restart the service machine
RESWORD	ACFSERVE RELOAD CONTROL VMO
RULEOPTS	ACFSERVE RELOAD CONTROL VMO
SAFDEF	ACFSERVE RELOAD CONTROL VMO
SSFTYPE	ACFSERVE RELOAD CONTROL VMO
TAPEVOLS	ACFSERVE RELOAD CONTROL VMO
WARN	ACFSERVE RELOAD CONTROL VMO

For more information on the ACFSERVE RELOAD command, see “Using the ACFSERVE Commands”.

APPLDEF-Infostorage Record Definitions

Record ID	Fields
APPLDEFqual	CLASS(classcode|classname) TYPE(typecode|typename) APPLDIV(divmask) APPLDLEN(maxidlength) DFTDRTN(defaultdivrout) RECID(RSB(mod1|recidmask1),...,RSB(modn|recidmaskn)) RECIDLEN(maxrecidlen) SELAUTH(privilege list)

This record defines structured infostorage records when your site has unique needs that standard infostorage records do not meet, such as VMO records. CA ACF2 for VM uses the APPLDEF record to validate access to your infostorage records.

Structured infostorage records contain fields that you can list, alter, and validate individually. In contrast, unstructured infostorage records (such as SHIFT records) are composed of lists you can only retrieve and validate, as opposed to accessing individual values in a record.

All infostorage records have keys in the class-type-division-recid form. The APPLDEF record defines the key structure for infostorage records and identifies the Record Structure Block (RSB) that defines the fields and where that RSB resides. A record structure block is a module that tells CA ACF2 for VM how to format and validate an infostorage record. It defines field lengths, validation routines, and other field characteristics.

Fields

CLASS(classcode/classname)

Specifies the class name and code associated with the storage class of this infostorage application. The classcode is the actual numeric character used in the class field of the record key (CA ACF2 for VM uses alphabetic characters). The classname is the name you indicate in the SET subcommand to access these records.

TYPE(typecode/typename)

Specifies the type name and code associated with the storage class of this infostorage application. The typecode is the actual three characters CA ACF2 for VM uses as the type field of the record key. The typename is the name you indicate in the SET subcommand to access these records.

APPLDIV(divmask)

Specifies the division that groups a set of related infostorage records. You can mask this field to refer to several different records in one infostorage application.

The division mask is the third field of the record key. You can access these records when you enter the SET subcommand. VMO records use this field for the SYSID.

APPLDLEN(maxidlength)

Specifies the maximum length of an application division value. The application division is specified with the APPLDIV field.

DFTDRTN(defaultdivrout)

Indicates the default division routine to invoke if you omit division when you use the SET subcommand to access the records. DFTDRTN can be the name of a CA ACF2 for VM-supplied routine or the name of a site-supplied routine.

CA ACF2 for VM supplies two default division routines, ACF00DFT and ACF00SID. Both of these default division routines are shipped as part of the service machine module (ACF2VM). ACF00SID returns the current SYSID (this routine is used for all VMO records). ACF00DFT returns the APPLDIV value if the class and type are unique. There is only one division defined for this APPLID and APPLIDIV is not masked.

RECID(RSBmodule/recidmask,...,RSBmodule/recidmask)

Indicates the record IDs to use with this infostorage application and the record structure block (RSB) associated with those record IDs. The recidmask is the name of the record ID in the record key of these records. You can mask the recidmask so several or all record IDs can use the same RSB. The RSB defines the fields of the records. RSB(modn) is the name of the RSB that defines the fields for the associated record IDs of this infostorage application.

RECIDLEN(maxrecidlen)

Indicates the maximum length of the record ID. Specify the record ID with the RECID field. The default is zero.

SELAUTH(privilege list)

Specifies the privilege field you must have in your logonid to use the SET and LIST subcommands for the infostorage application this APPLDEF record defines. The SET subcommand accesses the infostorage application. The LIST subcommand displays the field values of the infostorage record. Any of the privileges that you can define in the logonid record (SECURITY, ACCOUNT, AUDIT, CONSULT, LEADER, and USER) you can also specify as the SELAUTH value. You can specify more than one privilege. If you specify ALL, a user with any of the standard CA ACF2 for VM logonid privileges can use the SET and LIST subcommands for the defined application. Because the SECURITY privilege allows full access authority, SECURITY in this field lets anyone with the SECURITY privilege issue other ACF subcommands besides SET and LIST. They can also issue INSERT, CHANGE, and DELETE subcommands. Any of the other access privileges, such as USER, let you use the SET and LIST subcommands. If you do not specify a value, the default is SECURITY, AUDIT, ACCOUNT, CONSULT, LEADER.

When you tell CA ACF2 for VM users to bypass INFOLIST validation processing for the infostorage record record defines, it checks the value you specify in the SELAUTH field only to determine the authority required to access the record. **Specify this field with caution.**

If you need more than one APPLDEF record, you can append a qualifier to the record name in the format APPLDEF(qualifier) to generate a unique record ID (for example, APPLDEF001 or APPLDEF.RSBA). This optional qualifier, together with the record ID name, cannot total more than 16 characters. If you use a period (.) as part of the qualifier string for the record name,
CA ACF2 for VM counts it as a character. For example, because APPLDEF is seven characters, a qualifier can be up to nine characters long. APPLDEF.RSBABCDE is a valid record ID and qualifier. You must create one APPLDEF record for each site-defined infostorage application.

SHOW Subcommand

In this example, SHOW APPLDEF displays the applications defined to the system.

```
-- INSTALLATION DEFINED STRUCTURED INFOSTORAGE APPLICATIONS --

VMO RECORD (SYSID / RECID): 2ND LEVEL / APPLDEF.1
CLASS(SHORT / LONG):    1     / OUROWN
TYPE (SHORT / LONG):    VAA    / VAA
SELECTION AUTHORIZATION:  SECURITY, ACCOUNT, AUDIT, CONSULT, LEADER
DEFAULT DIVISION ROUTINE:  ACFOOSID
ACTIVE DIVISION:      *********
ASSOCIATED RSB / RECORD ID: TSTRSBMS / MESSAGE
              TSTRSB01 / TEST01
              TSTRSBMS / TEST02

VMO RECORD (SYSID / RECID): 2NDLEVEL / APPLDEF
CLASS (SHORT / LONG):    2    / TSTAPPL
TYPE (SHORT / LONG):    TST   / TESTING
SELECTION AUTHORIZATION:  SECURITY, ACCOUNT, AUDIT, CONSULT, LEADER
DEFAULT DIVISION ROUTINE:  ACF00SID
ACTIVE DIVISION:      ********
ASSOCIATED RSB / RECORD ID: TSTRSBMS / TEST02
```

If you have not defined any records, the SHOW APPLDEF command displays this title:

```
--NO APPLICATION DEFINITIONS EXIST ON THIS SYSTEM--
```

BACKUP Record-Automatic Time Backup

Record ID	Fields
BACKUP	AUTOLOG(null|machine id) DDSNID(BACKUP|ddsngrp) MDISK(195|ccuu) NOTIFY(OPERATOR|user1,...,user16) TIME(00:01|time1,...,time16)

The BACKUP VMO record sets the time of day when CA ACF2 for VM automatically creates backup copies of its three databases. This record also defines the filenames of the backup copies and the address of the minidisk that contains the backup copies. See the “Backup and Recovery” chapter for complete information about CA ACF2 for VM database recovery and the automatic backup facility.

Fields

AUTOLOG(null|machine id)

Specifies the user ID of the postbackup service machine. This machine is autologged after any successfully completed backup, whether invoked automatically or by the ACFSERVE command.

DDSNID(BACKUP|ddsngrp)

Specifies the name of the @DDSN macro where the filenames of the backup databases are obtained. The default is BACKUP.

MDISK(195|ccuu)

Specifies the address of the minidisk where the backup copies reside. You must access this minidisk in read and write mode when the automatic backup process begins. The default is the 195 minidisk.

NOTIFY(OPERATOR|user1,...,user16)

Sends messages about backup processing to a list of logonids. These messages are sent when backup starts, ends, or terminates with an error. You can specify multiple logonids using commas as separators. The default is OPERATOR.

TIME(00:01|time1,...,time16)

Indicates the time of day when the backup is started. You can specify up to 16 time-of-day values in a 24-hour format, separated by commas (for example, TIME(00:01,12:01)). The default is (00:01). Use TIME(00:00) to prevent automatic backups. You can also start backup processing with the ACFSERVE BACKUP command. When you specify NOBACKUP to the ACF2 startup prompt, automatic backups are suppressed.

SHOW Subcommand

The SHOW BACKUP subcommand displays the backup options currently in effect.

CLASMAP Record-SAF Resource Classes

Record ID	Fields
CLASMAPqual	RESOURCE(class) RSRCTYPE(typecode) MUSID(musassid|********) ENTITYLN(0|entitylength)

The CLASMAP record translates eight-character resource classes into three-byte CA ACF2 for VM resource type codes. The three-character resource type code allows you to write resource rules to validate security calls for the specified classes. CA ACF2 for VM checks the CLASMAP record for this type code for all SAF calls.

Fields

RESOURCE(class)

Specifies the explicit eight-character resource class from the CLASS keyword on the RACROUTE macro. Normal CA ACF2 for VM resource name masking conventions apply.

RSRCTYPE(typecode)

Specifies the explicit three-character resource type code associated with the class. If you define a RESOURCE but do not define a RSRCTYPE, CA ACF2 for VM uses the first three characters of the RESOURCE as the RSRCTYPE. Use this type code to write resource rules to perform validation. This value cannot be a mask. If you want to mask the name of the resource in your resource rule key, add this type code to the VMO RESTYPE record and perform a rebuild. For more details, see the “About Resource Rules” chapter.

MUSID(musassid|******)**

Identifies the MUSASS to which the CLASMAP record applies. This lets several MUSASSes that share the same resource class use a different type code. Normal CA ACF2 for VM resource name masking conventions apply.

ENTITYLN(0|entitylength)

Specifies the entity length of the specified SAF class. The default is 0. Zero causes CA ACF2 for VM to search its internal CLASMAP definitions; non-zero causes the VMO CLASMAP to be used. The resultant CLASMAP record, VMO or internal, is used for both RSRCTYPE and ENTITYLN. If the resultant ENTITYLN is zero, CA ACF2 for VM assigns a length of 39, the IBM default.

Creating Multiple VMO CLASMAP Records

If you need more than one CLASMAP record, append a qualifier to the record name in the format CLASMAPqual to generate a unique record ID (for example, CLASMAPVMAN or CLASMAP.DATASET). This optional qualifier can be up to nine characters, and must immediately follow the characters CLASMAP. If you use a period (.) as part of the qualifier string for the record name, CA ACF2 for VM counts it as one of the nine characters.

Using CLASMAP Records to Validate SAF RACROUTE Calls

You must specify a CLASMAP record for the following type of SAF RACROUTE call that you want to validate:

```
REQUEST=AUTH, CLASS=DATASET|others
```

AUTH calls with a CLASS specification of DATASET result in a data set validation. AUTH calls with any other CLASS specified result in a resource validation.

For more information, see the Defining Structured Infostorage Records section.

SHOW Subcommand

The SHOW CLASMAP subcommand of the ACF command displays the internal definitions (CA ACF2 for VM-defined) and external definitions (site-defined) of SAF calls that are being translated.

CMDLIM Record-Command Limiting

Record ID	Fields
CMDLIM	COMMANDS(null|cmdname1,...,cmdname256) LIMIT(ALL|NONE|INCLUDE|EXCLUDE) MDLTYPE(E21|mdltype) MODE(ABORT|WARN|LOG|QUIET|RULE, no-rule,no$mode) NOSPOOL(PREVENT|PREVENT-LOG|LOG|ALLOW) SYNERR(PREVENT|PREVENT-LOG|LOG|ALLOW)

The CMDLIM VMO record provides CA ACF2 for VM validation for CP commands. Use the INCLUDE or EXCLUDE values to list the commands. You cannot reload this record. IPL the system to make any changes effective.

Fields

COMMANDS(null|cmdname1,...,cmdname256)

Specifies all protected commands. You can specify up to 256 command names.

LIMIT(ALL|NONE|INCLUDE|EXCLUDE)

Specifies how to limit the commands. The values for the LIMIT operand are defined below.

ALL

This is the default. It specifies that CA ACF2 for VM validates all CP commands.

NONE

Specifies that CA ACF2 for VM will not validate any CP commands.

INCLUDE

Indicates that CA ACF2 for VM validates the commands listed.

EXCLUDE

Indicates that CA ACF2 for VM does not validate the commands listed.

MDLTYPE(E21|mdltype)

Specifies the three-character name that identifies the appropriate command limiting model for this VM system. Each command limiting model defines a valid syntax for a CP command to CA ACF2 for VM. E21 is the default for command limiting models.

Note: The value of MDLTYPE can be any three-characters that normally identifies the set of models as belonging to a particular CP release or to a particular VM system.

MODE(ABORT|WARN|LOG|QUIET|RULE,no-rule,no$mode)

Specifies the mode that CA ACF2 for VM command limiting is performed in. With this operand, you can implement the command limiting feature gradually. However, the mode for this operand applies only to command limiting and is independent of the CA ACF2 for VM system mode. The command limiting modes are defined below.

ABORT

Indicates the command is denied.

WARN

Indicates execution is allowed, but the violation is logged and a warning message is issued to the user.

LOG

Indicates execution is allowed and the violation is logged.

QUIET

Indicates CA ACF2 for VM command limiting is disabled. However, you can still issue commands.

RULE,no-rule,no$mode

Indicates CA ACF2 for VM checks the $MODE control statement in the command limiting rule.

NOSPOOL(PREVENT|PREVENT-LOG|LOG|ALLOW)

Sets the global SPOOL FILE NOT FOUND option for the system. NOSPOOL defines the actions that CA ACF2 for VM takes whenever you issue a CP spool command for a spool file that does not exist. (There are two CA ACF2 for VM mechanisms for overriding this operand. They are explained following this text.) You can specify one of four keyword options to enable CA ACF2 for VM to take these actions when spool file errors are found:

PREVENT

This is the default. It causes CA ACF2 for VM to display the following message:

```
ACFpgm277I No spool files found
```

CA ACF2 for VM automatically denies the command so that it never reaches the IBM CP command processor. This avoids the overhead of going through rule validation, and the overhead of passing the command to CP where it would be rejected. CA ACF2 for VM does not issue a violation and the SEC-VIO count is not incremented because of the syntax error. CA ACF2 for VM does not create an SMF record for the error.

PREVENT-LOG

This setting works the same as PREVENT, but the error generates an SMF record that appears in the Command Limiting Journal. This option does not cause a violation.

LOG

CA ACF2 for VM passes the command to CP for normal syntax checking. CP issues you the standard error messages. CA ACF2 for VM creates an SMF record for the error, and it appears in the Command Limiting Journal.

ALLOW

CA ACF2 for VM passes the command to CP for normal syntax checking. CP sends you the standard error message. CA ACF2 for VM does not create an SMF logging record for the error.

The NOSPOOL field of the CMDLIM VMO record is one mechanism for changing the way CA ACF2 for VM handles a SPOOL FILE NOT FOUND condition. You can also override this operand two other ways:

Command model options

Specifies how you want CA ACF2 for VM to react to each SPOOL FILE NOT FOUND condition (PREVENT, PREVENT-LOG, LOG, or ALLOW). This is the second level of override.

Logonid record field

Specifies how you want CA ACF2 for VM to react when a user spool command results in a SPOOL FILE NOT FOUND condition (PREVENT, PREVENT-LOG, LOG, or ALLOW). This is the highest level of override.

When the default is in effect for both mechanisms, CA ACF2 for VM performs the same way as the PREVENT option for the NOSPOOL field.

For more information on these override features, consult the *Command and Diagnose Limiting Guide.*

SYNERR(PREVENT|PREVENT-LOG|LOG|ALLOW)

Sets the global command syntax error option for the system. SYNERR defines the actions CA ACF2 for VM takes when it finds a command syntax error. You can specify one of four keywords to let CA ACF2 for VM take these actions when it finds command syntax errors:

PREVENT

This is the default for the SYNERR operand. CA ACF2 for VM denies the command so it never reaches the IBM CP command processor. This avoids the overhead of going through rule validation, and the overhead of passing the command to CP where it would be rejected. CA ACF2 for VM does not send a violation and the
SEC-VIO count is not incremented.
CA ACF2 for VM does not create SMF records for the error. It rejects the command and sends the following message:

```
ACFpgm274E ACF2 syntax error, Operand number 'nn'-command 'cmd' rejected
```

PREVENT-LOG

This setting works the same as PREVENT, but
CA ACF2 for VM generates an SMF record that appears in the Command Limiting Journal. This does not cause a violation.

LOG

CA ACF2 for VM passes the command to CP for normal syntax checking. CP issues you standard error messages. CA ACF2 for VM writes an SMF record and it appears in the Command Limiting Journal (ACFRPTCL).

ALLOW

CA ACF2 for VM passes the command to CP for normal syntax checking. CP issues standard error messages and CA ACF2 for VM does not create an SMF record for the error.

SHOW Subcommand

The SHOW CMDLIM subcommand displays the command limiting options in effect. For more information about this record, see the Command *and Diagnose Limiting Guide.*

DBSYNC Record-Database Synchronization Component

Record ID	Fields
DBSYNC	CCI_ID(CCIVM|null|machineid) INFO|NOINFO LASTACC|NOLASTACC LID|NOLID ONCEADAY|NOONCEADAY RULES|NORULES SYNC_ID(ACFSYNC|null|machineid) SYNCQLMT(2048|limit)

The DBSYNC VMO record specifies various options for the Database Synchronization Component. These options were implemented as a replacement for shared CA ACF2 for VM databases between VM and OS/390.

Fields

CCI_ID(CCIVM|null|machineid)

Specifies the user ID of the Common Communications Interface (CCI) service machine. You must have installed CCI from the CA-CIS Services VM tape.

INFO|NOINFO

Specifies if updates to the Infostorage database are synchronized.

LASTACC|NOLASTACC

Specifies whether updates to logonid records due to logons are synchronized. By specifying NOLASTACC, no logonid updates are synchronized unless the password is changed during logon. If the password is changed, the update is synchronized according to the LID|NOLID setting.

LID|NOLID

Specifies whether updates to logonid records are synchronized. This does not include updates to logonid records due to logon unless the password was changed during the logon. See the LASTACC setting.

ONCEADAY|NOONCEADAY

Specifies whether to limit updates to logonid records under the LASTACC setting to just once a day. If NOLASTACC is specified, the ONCEADAY setting is ignored.

RULES|NORULES

Specifies whether updates to the Rules database are synchronized.

SYNC_ID(ACFSYNC|null|machineid)

Specifies the user ID of the Database Synchronization Component service machine.

SYNCQLMT(2048|limit)

Specifies the number of synchronization requests that CA ACF2 for VM can queue if the Database Synchronization Component service machine is not responding.

ACFSERVE Subcommand

The ACFSERVE QUERY SYNC subcommand displays the options currently set for the Database Synchronization Component.

DIAGLIM Record-Diagnose Limiting

Record ID	Fields
DIAGLIM	DIAGS(null|diag1,...,diag256) LIMIT(ALL|NONE|INCLUDE|EXCLUDE) MDLTYPE(E21|mdltype) MODE(ABORT|WARN|LOG|QUIET|RULE, no-rule,no$mode)

The DIAGLIM VMO record specifies the diagnose requests for CA ACF2 for VM validation. You cannot reload this record. IPL the system to make any changes effective.

Fields

DIAGS(null|diag1,...,diag256)

Specifies the diagnoses CA ACF2 for VM protects when you specify the LIMIT (INCLUDE or EXCLUDE) operand. You can specify up to 256 diagnose codes.

LIMIT(ALL|NONE|INCLUDE|EXCLUDE)

Specifies how to limit the diagnoses. The values for this option are defined below.

ALL

Specifies that CA ACF2 for VM validates all diagnoses. This is the default.

NONE

Specifies that all diagnoses listed are allowed and CA ACF2 for VM does not validate any diagnoses.

INCLUDE

Indicates that CA ACF2 for VM validates the diagnoses listed.

EXCLUDE

Indicates that CA ACF2 for VM does not validate the diagnoses listed.

Here is how to use the LIMIT operand:

- LIMIT(INCLUDE) DIAGS(4,1C,80)

  Indicates that CA ACF2 for VM validates only the 4, 1C, and 80 diagnoses. It does not validate any other diagnoses.

- LIMIT(EXCLUDE) DIAGS(0,4C,6C)

  Indicates that CA ACF2 for VM does not validate the 0, 4C, and 6C diagnoses. It validates all other diagnoses.

- LIMIT(ALL) DIAGS( )

  Tells CA ACF2 for VM to validate all diagnoses. It ignores any diagnoses specified in the DIAGS( ) parameter list.

- LIMIT(NONE) DIAGS( )

  Tells CA ACF2 for VM not to limit any diagnoses. It ignores any diagnoses specified in the DIAGS( ) parameter list.

MDLTYPE(E21|mdltype)

Specifies the three-character name that identifies the appropriate diagnose limiting rule for this VM system. E21 is the default for diagnose limiting.

MODE(ABORT|WARN|LOG|QUIET|RULE,no-rule,no$mode)

Specifies the mode that CA ACF2 for VM diagnose limiting is performed in. With this operand, you can implement diagnose limiting gradually. However, the mode for this operand applies only to diagnose limiting and is totally independent of the CA ACF2 for VM system mode. The values for this operand are defined below.

ABORT

Indicates the diagnose is denied.

WARN

Indicates execution is allowed, but the violation is logged and a warning message is issued.

LOG

Indicates execution is allowed and the violation is logged.

QUIET

Indicates CA ACF2 for VM diagnose limiting is disabled. However, you can still issue diagnoses.

RULE,no-rule,no$mode

Indicates CA ACF2 for VM checks the $MODE control statement in the diagnose instruction.

SHOW Subcommand

The SHOW DIAGLIM subcommand displays the current diagnose limiting options in effect for this VMO record.

EXITS Record-User Password Exits

Record ID	Fields
EXITS	EXPPXIT(null|module name) PENCRYP(null|module name) NEWPXIT(null|module name)

The EXITS VMO record defines self-contained CMS modules that are user password exits. The exits are loaded during service machine initialization.

Fields

EXPPXIT(null|module name)

Specifies the name of a user-written expired password routine that validates expired user passwords. The routine gets control during system entry validation when the password associated with the logonid is expired. If you do not use this exit module, you do not have to change the EXPPXIT operand.

PENCRYP(null|module name)

Specifies the name of a user-written password encryption exit module. The named module gets control during password validation at log on, during a password modification (such as through the CHANGE or INSERT subcommands), or during password revalidation (DirMaint). The PENCRYP exit module lets you implement your own password encryption algorithm. If you do not use this exit module, you do not have to change the PENCRYP operand.

NEWPXIT(null|module name)

Specifies the name of a user-written password exit module. The named module gets control when you supply a new password when you access the system, or when you specify the PASSWORD field through the CHANGE or INSERT subcommands. The NEWPXIT exit module performs password checking and CA ACF2 for VM validation. If you do not use this exit module, you do not have to change the NEWPXIT operand.

Requirements for using user password exits are:

- Code Testing. We recommend that you test all user exit code on a second level VM system because an error in a user exit routine can cause service machine abends.
- Code Implementation. Perform the following steps to implement a password exit:
  - Write the exit (a self-contained CMS module), assemble it, and put the text deck on a disk accessed during service machine initialization.
  - Define the operands of the EXITS field: NEWPXIT(modulename) or PENCRYP(modulename).
- IPL the CA ACF2 for VM service machine.

SHOW Subcommand

The SHOW ACTIVE subcommand displays the values currently in effect.

Linux Machine Definitions (LINUX)

Record ID	Fields
Linux*qual*	ACTIVE|NOACTIVE IPADDR(*IP Address*) MACHNAME(*Linux machine name*)

The Linux record defines LINUX machines to CA ACF2 for VM.

Fields

ACTIVE|NOACTIVE

Specifies whether the Linux machine is active or not. ACTIVE is the default.

IPADDR(*IP Address*)

The IP address of the Linux machine. This is a required field.

MACHNAME(*Linux machine name*)

Specifies the LINUX machine name. This filed can be from 1-246 characters long. Valid characters are A-Z, z-a, 0-9, - (dash) and . (period). This is a required field.

MAINT Record-System Maintenance Options

Record ID	Fields
MAINTqual	LID (null|logonid1,logonid2,...,logonid256)

The MAINT VMO record defines logonids that bypass access validation. When CA ACF2 for VM recognizes these logonids, it bypasses rule validation and does not create SMF records. You must define logonids in the MAINT VMO record as noncancelable (NON-CNCL). If a logonid does not have the NON-CNCL privilege, CA ACF2 for VM does not examine the maintenance program list. The MAINT VMO record defines virtual machines that perform maintenance, such as disk backup machines.

Fields

LID(null|logonid1,logonid2,...,logonid3)

Indicates the logonids of authorized maintenance users.

If you need more than one MAINT record, add a qualifier to the record name to generate a unique record ID (for example, MAINTABC or MAINT.ABC). This optional qualifier, when combined with the record ID of MAINT, can be up to 16 characters, and must immediately follow the characters MAINT. If you use a period (.) as part of the qualifier string for the record name, CA ACF2 for VM counts it as one of the characters. For example, in the record ID of MAINTABC, because the record ID (MAINT) is only five characters, the qualifier (ABC) could be up to 11 characters. An example of how to use qualifiers follows.

Assume that you have two CPU-assigned CA ACF2 for VM SYSIDs of VMCPUPRD and VMCPUTST, then:

- Create a MAINT.ALL record with a SYSID of VMCPU***, specifying the MAINT user IDs that both CPUs will use
- Create a MAINT.PRD record with a SYSID of VMCPUPRD, specifying the MAINT user IDs that only VMCPUPRD will use
- Create a MAINT.TST record with a SYSID of VMCPUTST, specifying the MAINT user IDs that only VMCPUTST will use.

On each CPU, issue an ACFSERVE RELOAD CONTROL VMO MAINT command:

- VMCPUPRD processes the MAINT.ALL and MAINT.PRD VMO records
- VMCPUTST processes the MAINT.ALL and MAINT.TST VMO records.

SHOW Subcommand

The SHOW MAINT subcommand displays the MAINT values currently in effect.

OPTS Record-CA ACF2 for VM Options Specification

Record ID	Fields
OPTS	ACCTVLD(NO|FULL|LID) ATTKEY(SYSTEM,key value) CPUTIME(LOCAL|GMT) DATE(MDY|DMY|YMD) DFTLNXG(*default_linux_group*) DFTLNXU(*default_linux_user*) DIAG84|NODIAG84 DIRIPL|NODIRIPL DSPVLD|NODSPVLD IDLEMN(60|nnn) IDLEOP(OFF|ALLOFF|DISC|LOGOFF|REPROMPT|NOLOGOFF) INFOLIST(SECURITY,AUDIT|privilege list) IUCVVLD|NOIUCVVLD MAXLID(512|nnnn) MAXPGRPS(32|nnn|125) MAXVIO(10|nnn) MODE(ABORT|WARN|LOG|QUIET|RULE,no-rule,no-$mode) NOTIFY|NONOTIFY POSIXDB|NOPOSIXDB TAPEDSN|NOTAPEDSN VMCFVLD|NOVMCFVLD VMCHK(NULL|bitfld)

The OPTS VMO record defines many CA ACF2 for VM system-wide options.

Fields

ACCTVLD(NO|FULL|LID)

Specifies account validation for VM operating systems. Account validation is optional. The values for this operand are defined below.

NO

Indicates CA ACF2 for VM account support is disabled. Directory account validations are performed the same as VM native.

FULL

Indicates full account support is in effect. We recommend this value for CA ACF2 for VM account validation. CA ACF2 for VM does not use the CP directory for accounting purposes except at system initialization time. This is the default.

- The default account number that is automatically selected for a virtual machine during system entry is kept in the new VMACCT logonid field. Virtual machines whose VMACCT field contains blanks must have an account number specified for them during system entry. Use the ACCOUNT operand of the LOGON and AUTOLOG commands for this purpose.
- CA ACF2 for VM validates account resource rules for all virtual machines.
- If you use a virtual machine, issue the CP SET ACCOUNT command to change your account number during a session.

LID

This is the alternative to full account support. With this value, CA ACF2 for VM account support is in effect only for machines specified with the VLDVMACT attribute. CA ACF2 for VM does not use the CP directory for accounting purposes except at system initialization time.

- The default account number that is automatically selected for a virtual machine during system entry is kept in the new VMACCT logonid field. You must specify an account for virtual machines with the VLDVMACT attribute and whose VMACCT field contains blanks during system entry. Use the ACCOUNT operand of the LOGON and AUTOLOG commands for this purpose. Virtual machines without the VLDVMACT attribute whose VMACCT field contains blanks cannot access the system.
- CA ACF2 for VM validates account resources only for virtual machines with the VLDVMACT logonid attribute. Machines without VLDVMACT do not undergo account resource rule validation. They are automatically assigned the account numbers kept in their VMACCT logonid fields unless these fields contain blanks. In that case, the machines cannot access the system.
- If you use a VLDVMACT-defined virtual machine, you can issue the CP SET ACCOUNT command to change your account number. If you use a machine that does not have VLDVMACT, you cannot change your account number.

ATTKEY(SYSTEM,key value)

Defines the $KEY value for your dedicated and attached DASD volumes for access validation. The values for this operand are defined below.

SYSTEM

Specifies the default ATTKEY value.

key-value

Specifies an ATTKEY value that you can define.

CPUTIME(LOCAL|GMT)

Specifies the time mode of the CPU. This specification determines how CA ACF2 for VM calculates a user's access time when processing zone records. If you specify GMT (Greenwich Mean Time), CA ACF2 for VM bases all time zone calculations on the time-of-day (TOD) clock. In LOCAL mode, CA ACF2 for VM first adjusts the TOD clock by the value stored in the CVTTZ field of the Communications Vector Table (CVT) and then bases all time zone calculations on the adjusted TOD clock. The default is LOCAL.

DATE(MDY|DMY|YMD)

Specifies whether the dates used are in month-day-year (mm/dd/yy), day-month-year (dd/mm/yy), or year-month-day (yy/mm/dd) format. You must specify these formats in the DATE field as MDY, DMY, or YMD, respectively. Because dates are stored internally in Julian (YYDDD) format, there is no incompatibility in changing this option after some rule sets already exist. The default is month-day-year (mm/dd/yy).

DFTLNXG(*default_linux_group*)

Specifies the name of the default Linux group profile record. It is used when a user accesses Linux and does not have a valid group defined in the Linux user profile record.

DFTLNXU(*default_linux_user*)

Specifies the name of the default Linux user profile record. It is used when a user accesses Linux and does not have a valid Linux user profile record.

DIAG84|NODIAG84

Indicates whether a user with the CP class B privilege (as IBM defines it) can dynamically update fields in a VM directory entry. This is the online VM directory and the duration of the update is until the next system IPL. You must provide the VM directory password of the entry. This operand lets you use the VM directory password, or your CA ACF2 for VM password. The values for this field are defined below.

DIAG84

Means that your CA ACF2 for VM password is required unless you have the DG84DIR attribute in your logonid record. If you have this attribute, you must use the directory password when you issue the DIAG84 diagnose instruction. This is the default.

You must supply your CA ACF2 for VM password for all diagnose x'84' operations except LOGPASS and MDISK. These operations require the directory password.

NODIAG84

Specifies that you must use the VM directory to update-in-place a VM directory entry.

DIRIPL|NODIRIPL

Controls command limiting of the IPL statement in the VM directory. DIRIPL causes command limiting to validate the directory IPL statement. DIRIPL prevents users from changing the IPL statement and bypassing command limiting. You should define the CMDLIM VMO record so that the IPL command is limited. If the IPL command is not limited, this option has no effect.

NODIRIPL bypasses command limiting for the directory IPL statement. If you limit the IPL command, only IPL commands are limited and the directory IPL statement is always allowed.

DSPVLD|NODSPVLD

Activates or deactivates dataspace protection. CA ACF2 for VM activates dataspace protection at installation. CA ACF2 for VM uses ESA dataspaces in CP on all releases of VM running on ESA-capable hardware.

IDLEMN(60|nnn)

Specifies the number of minutes (from 1 to 240) that the terminal a user is logged onto can be idle before idle terminal processing begins. You can override this value at the individual logonid level using the VMIDLEMN logonid restriction. The default is 60

IDLEOP(OFF|ALLOFF|DISC|LOGOFF|NOLOGOFF|REPROMPT)

Specifies the type of idle terminal processing performed when a user exceeds the idle time limit. You can override this value at the individual logonid level using the VMIDLEOP logonid restriction, except when IDLEOP is set to ALLOFF. The values for this parameter are:

OFF

Disables system-wide idle terminal processing. However, you can enable idle terminal processing at the individual logonid level using the VMIDLEOP logonid restriction. This is the default.

ALLOFF

Disables system-wide idle terminal processing, regardless of the VMIDLEOP restrictions defined at the individual logonid level.

DISC

Forces disconnection from the system when the user exceeds the idle terminal limit.

LOGOFF

Forces the user off the system when the user exceeds the idle terminal limit.

NOLOGOFF

Prompts a user to enter logon password when idle terminal limit is exceeded. Incorrect passwords are counted as a password violation so the user can be suspended after entering too many incorrect passwords. The user can also disconnect from the system at this prompt. Similar to REPROMPT, but the option to logoff is not allowed.

REPROMPT

Prompts a user to enter his logon password when he exceeds the idle terminal limit. Incorrect passwords are counted as a password violation so the user can get suspended after he enters too many incorrect passwords. The user can also logoff or disconnect from the system at this prompt.

INFOLIST(SECURITY,AUDIT|privilege list)

Specifies which logonid attributes are necessary to list Infostorage records (such as entry lists, scope, and shift and zone records). The user does not have the authority to change these records. It does not convey the authority to compile and store resource rules or access rules. The default is SECURITY,AUDIT, indicating that users with either attribute are authorized. Scopes do not affect logonids with the privileges listed in the INFOLIST record. You should also refer to the SELAUTH field of the APPLDEF VMO record described earlier in this chapter.

IUCVVLD|NOIUCVVLD

Implements CA ACF2 for VM validation of a one way IUCV and APPC/VM path connection. This validation takes place through resource rules. The values for this field are defined below.

IUCVVLD

Specifies standard CA ACF2 for VM validation through resource rules. This is the default.

NOIUCVVLD

Specifies that CA ACF2 for VM does not validate IUCVVLD.

MAXPGRPS(32|nnn|125)

For POSIX environments only. Defines the maximum number of POSIX groups allowed. Be conservative in your estimate. CP allocates storage based on this number for every user that logs onto VM. The minumum value is 32; maximum is 125.

MAXLID(512|nnnn)

Specifies the maximum number of active CA ACF2 for VM logonids. Active logonids include virtual machines logged onto VM, users logged onto SRF applications, users logged onto CA ACF2 for VM VSE, and users executing batch jobs. You can use the same CA ACF2 for VM logonid in two or more of the preceding situations and it will still be counted as one active logonid. MAXLID calculates the amount of virtual storage that the CA ACF2 for VM service machine needs to reserve for active logonids during startup. It is not intended to restrict the number of VM user IDs logged onto a system. The minimum value is eight and the maximum value is 524,288. The default is 512. You must adjust the storage size of the service machine VM directory entry to accommodate this value. The default MAXLID value of 512 is normally inadequate for large VM systems. For large systems, you must change the default value to reflect the maximum number of users that are logged onto the system simultaneously.

MAXVIO(10|nnn)

Specifies the maximum number of security violations in a single job (or CMS session) before CA ACF2 for VM terminates the job or CMS session rather than just the task. The default value is ten. The maximum value is 32,767.

MODE(ABORT|WARN|LOG|QUIET|RULE,no-rule,no$mode)

Defines the mode of CA ACF2 for VM as it relates to data access. Depending on the MODE value, CA ACF2 for VM validates:

- CP LINK commands
- CMS file access requests
- OS/390 data set access requests
- VSE file access requests
- CP ATTACH requests for DASD
- LINK and DEDICATE commands defined in a user's VM directory entry
- Tape volume access requests for volumes defined in the TAPEVOLS VMO record.

Set MODE to one of the following:

ABORT

Logs attempted violations, issues violation messages, and denies accesses. This is the default value.

WARN

Logs access violations and issues warning messages, but lets accesses continue.

LOG

Logs access violations, but lets accesses continue.

QUIET

Disables CA ACF2 for VM data access rule validations.

CA ACF2 for VM logonid record validations and similar user and system access validations still occur.

RULE

Specifies that CA ACF2 for VM checks the $MODE control statement in the appropriate access rule set to determine what action to take if the access request violates security. The value of the $MODE statement can be QUIET, LOG, WARN, or ABORT, as defined above. CA ACF2 for VM only uses the $MODE control statement when the (RULE,no-rule,no-$mode) option is in effect and it determines that a data access request violates security. The two positional parameters, no-rule and no-$mode, are defined as follows:

no-rule

Specifies the action CA ACF2 for VM takes if no access rule is found when RULE mode is in effect. The value for this parameter can be QUIET, LOG, WARN, or ABORT, as defined above.

no-$mode

Specifies the action CA ACF2 for VM takes if no $MODE control statement is found in the applicable access rule set when RULE mode is in effect. The value for this parameter may be QUIET, LOG, WARN, or ABORT, as defined above.

For example, if TLCAMS tries to link to TLCPJM's 0191 minidisk for write access but the rule does not grant TLCAMS access, CA ACF2 for VM checks the $MODE control statement in the access rule and bases the access on the $MODE value. If $MODE(LOG) is specified in the access rule set, TLCAMS can link to TLCPJM's 0191 minidisk and CA ACF2 for VM creates a logging record. If $MODE(ABORT) is specified, TLCAMS cannot access and CA ACF2 for VM creates a logging record detailing the access violation attempt.

Note: If an I/O error occurs reading a rule, CA ACF2 for VM treats it as a RULE NOT FOUND condition. If you are in RULE mode, CA ACF2 for VM ignores any $mode statement in that rule. If the no-rule setting of RULE mode is set to QUIET, access is allowed and CA ACF2 for VM does not create any SMF loggings. Because of this potential security exposure, we recommend that you do not use QUIET for the no-rule or no-$mode settings of RULE mode if you are protecting sensitive data.

NOTAPEDSN|TAPEDSN

Specifies whether the VMTAPE interface performs validation at the data set name (DSN) level. With NOTAPEDSN (the default), the interface does not perform DSN validation. DSN validation occurs if TAPEDSN is specified. There is no ACF subcommand for displaying the OPTIONS field of the VMO record TAPEDSN setting.

NOTIFY|NONOTIFY

Specifies whether CA ACF2 for VM sends a message to the user at logon time (giving the date, time, and source of the user's last successful system access). This message warns you if someone used your logonid and password to access CA ACF2 for VM. Teach your users about this message and show them how to report possible misuse of their logonids. Specify NONOTIFY to suppress the message. The default is NOTIFY, which displays the message.

POSIXDB|NOPOSIXDB

For POSIX environments only. Defines who owns the POSIX database.

POSIXDB

CA ACF2 for VM informs VM that it is maintaining the POSIX databases. CA ACF2 for VM responds to the DIAGA0 subcode X'08' and takes over POSIX Database management.

NOPOSIXDB

CA ACF2 for VM does not maintain POSIX databases. CA ACF2 for VM responds negatively to the DIAGA0 subcode X'08'. This is the default.

VMCFVLD|NOVMCFVLD

Implements CA ACF2 for VM validation of a one-way VMCF path connection. This validation takes place through resource rules. The values for this operand are defined below.

VMCFVLD

Specifies standard CA ACF2 for VM validation through resource rules. This is the default.

NOVMCFVLD

Specifies that CA ACF2 for VM does not validate through resource rules.

VMCHK(null|VM|VMXA|bit-fld)

Defines the logonid record privilege you need to access the system. If you do not have the appropriate privilege turned on in your logonid record, the logon is denied. Before you modify this operand to reflect VM or a locally defined bit-field, ensure that users have the appropriate value turned on in their logonid record. The values for this operand are listed below.

null

If you do not specify a value for VMCHK, standard
CA ACF2 for VM logon validation is done but does not require a specific privilege to logon. We recommend that you only use VMCHK in environments with shared CA ACF2 for VM databases. To remove a privilege from VMCHK if one exists, use the null value instead of a field name. For example, specify change opts vmchk(). The null setting is the default.

bit-field

You can specify CA ACF2 for VM-defined VM and VMXA logonid privileges to be checked at logon. If you specify a locally-defined field, add an @CFDE macro defining your bit-field and add the field to your USERID or USERXLID COPY files.

SHOW Subcommand

The SHOW STATE subcommand displays the values specified in the OPTS VMO record for the ATTKEY, CPUTIME, DATE, MAXLID, MAXVIO, MODE, and VMCHK fields. The SHOW ACTIVE subcommand displays the ACCTVLD, DIAG84, IUCVVLD, and VMCFVLD fields.

POSIX Support

You can maintain POSIX Database information in the VM directory or with an External Security Manager (ESM) such as CA ACF2 for VM. See the “OpenExtensions VM Support” chapter for additional information.

Password Phrase Record (PWPHRASE)

Password phrases may be used for user authentication with applications that support password phrases. You may have a password and a password phrase defined to your Logonid. Password phrases are not required to be specified. You can authenticate passwords for applications that support only passwords. However, passwords and password phrases are mutually exclusive for authentication. You may authenticate using only one, a password or password phrase, but not both, during a single authentication process for applications that support both passwords and password phrases.

The VMO PWPHRASE record allows you to set the following global password phrase options and controls:

Record ID	Fields
PWPHRASE	ALLOW|NOALLOW ALPHA(0|*nnn*) CMD-CHG|NOCMD-CHG HISTORY(0|*nn*) LID|NOLID MAXDAYS(100|*nnn*) MAXLEN(100|*nnn*) MINDAYS(0|*nnn*) MINLEN(9|*nnn*) MINWORD(1|*nnn*) NUMERIC(0|*nnn*) REPCHAR(null|0|*nn*) SPECIAL(0|*nnn*) SPECLIST() TEMP-AGE|NOTEMP-AGE WARNDAYS(1|*nnn*)

Fields

ALLOW|NOALLOW

Specifies whether all users on the system are allowed to authenticate using a password phrase. The default is NOALLOW, which indicates that authentication with a password phrase is not allowed for all users.

Note: The NOALLOW option may be overridden by specifying the PWPALLOW option on the Logonid. For more information, see the PWPALLOW option in the Logonid Record Field Descriptions section of the chapter "About the Logonid Record."

ALPHA(0|*nnn*)

Specifies the minimum number of alphabetic characters (a-z or A-Z) required in a new password phrase. Valid values are 0 - 100. The default is 0, which indicates that CA ACF2 for VM will not validate the password phrase for alphabetic characters.

Note: Changes to this parameter take effect at the next password phrase change of the user.

CMD-CHG|NOCMD-CHG

Specifies if password phrase changes are allowed with the ACF CHANGE command. The default is CMD-CHG, which permits password phrase changes through the ACF CHANGE command.

Note: This option does not affect administrators who are changing the password phrases of other users. It does, however, affect administrators changing their own password phrases. The purpose of the CMD-CHG option is to require users to change their password phrases only at system entry.

HISTORY(0|nn)

Specifies the number of previous password phrases to be checked to prevent reuse of a password phrase. Valid values are 0- 32. A value of either 0 or 1 indicates that no previous password phrases are checked; only the current password phrase is checked (the default is 0).

For example, specifying HISTORY(2) indicates that the current password phrase and the previous password phrase are checked. HISTORY(32) indicates that the current password phrase and the last 31 previous password phrases are checked.

LID|NOLID

Specifies that a logonid cannot be contained in any part of a new password phrase. The default is NOLID, which indicates that CA ACF2 for VM will not check for a logonid in a new password phrase.

Note: Before the password phrase is compared to the logonid, it is temporarily upper-cased. Changes to this parameter take effect at the next password phrase change of the user.

MAXDAYS(0|*nnn*)

Specifies the global value for the maximum number of days permitted between password phrase changes before the password phrase expires. This is based on the date specified in the PWP-TOD field in the User PWPHRASE Profile record. Valid values are 0-255. The default is 0, indicating that there is no global value set

Note: Any non-zero value in the PWP-MAXD field of the User PWPHRASE Profile record will override this value for validations.

MAXLEN(100|*nnn*)

Specifies the global maximum number of characters allowed in a new password phrase. Valid values are 9-200. The default is 100.

Note: If you use the Database Synchronization Component to logically share databases with another system, each system must have the same value for the MAXLEN field.

MINDAYS(0|*nnn*)

Specifies the global value for the minimum number of days that must elapse before a password phrase can be changed. Valid values are 0-254. The default is 0, indicating that there is no value set.

MINLEN(9|*nnn*)

Specifies the global minimum number of characters required in a new password phrase. Valid values are 9-200. The default is 9.

Note: If you use the Database Synchronization Component to logically share databases with another system, each system must have the same value for the MINLEN field.

MINWORD(1|*nnn*)

Specifies the global minimum number of words required in a new password phrase. Words are delimited by one or more spaces (x'40'). Valid values are 0-50. The default is 1. Zero indicates that this option is not active and CA ACF2 for VM will not perform validation of new password phrases for this option.

Note: Changes to this parameter take effect at the next password phrase change.

NUMERIC(0|*nnn*)

Specifies the minimum number of numeric characters (0-9) required in a new password phrase. Valid values are 0-100. The default is 0, which indicates that CA ACF2 for VM will not validate the new password phrase for numeric characters.

Note: Changes to this parameter take effect at the next password phrase change of the user.

REPCHAR(null|0|*nn*)

Specifies the number of consecutively repeating pairs of characters allowed in a new password phrase. Valid values are 0-99. The default is null-specified as REPCHAR(), which indicates that CA ACF2 for VM will not validate the new password phrase for consecutively repeating pairs of characters. A value of 0 indicates that the new password phrase cannot contain any consecutively repeating pairs of characters, for example, RABIT. A value of 1 indicates that a new password phrase can contain up to one consecutively repeating pair of characters, for example RABIT, RABBIT, but not RABBBIT). A valid new password phrase could be "The rabbit jumped" or "I need your help". However, CA ACF2 for VM will not allow "The rabbbit jumped" since "bbb" is considered two consecutively repeating characters.

Note: Changes to this parameter take effect at the next password phrase change of the user.

SPECIAL(0|*nnn*)

Specifies the minimum number of special characters required in a new password phrase. Special characters include: characters listed in the SPECLIST() field of this record, national characters (@ # $), and blanks (spaces). Valid values are 0-100. The default is 0, which indicates that no special characters are required. For example, when SPECIAL(3) is specified, a valid password phrase must contain at least three special characters, such as: "reading and writing are great skills" and "jane doe@companyx is my email."

Note: Changes to this parameter take effect at the next password phrase change of the user.

SPECLIST()

Specifies the list of valid, non-alphanumeric characters that may be contained in a new password phrase in addition to default alphanumeric (a z, A-Z, 0-9), national (@ # $) characters and blanks (spaces). If this field is not specified, the default is national characters and blanks. The following character values that may be specified in this field are:

Character Name	Character
Ampersand	&
Asterisk	*
Carat	^
Colon	:
Equal sign	=
Exclamation point	!
Hyphen	-
Percent sign	%

Character Name	Character
Period	.
Question mark	?
Underscore	_
Vertical line	|

Example: When SPECLIST(& * -) is specified, a valid password phrase can contain ampersand(&), asterisk(*), and hyphen(-) characters. The following are examples of valid password phrases: 'NEW#PHRA', 'NEW*PH&A', '123#PHRA', 'NEWPHRA@' or '#NEW-PHR'.

Note: Single and double quote marks are not permitted within new password phrases. Changes to this parameter take effect at the next password phrase change of the user.

TEMP-AGE|NOTEMP-AGE

Specifies whether temporary password phrases will be included in the password phrase history. A "temporary password phrase" is a new password phrase that is immediately expired at the time it is set. The default is TEMP-AGE, temporary password phrases will be aged.

WARNDAYS(1|*nnn*)

Specifies the number of days a warning message is issued before the password phrase expires. On those days, a warning message is displayed each time a user tries to access the system.

Valid values are 0-255. The default is 1. If zero is specified, no warning message is issued.

Implementing PWPHRASE

To implement the use of password phrases, the initial password phrase must be set for the user by inserting the PWPHRASE segment of the USER profile record.

Activating the VMO PWPHRASE Options

You must issue the following command for the insert or change to the VMO PWPHRASE record to take effect. CA ACF2 for VM does not recognize the change until the VMO records are built at the next IPL of the system.

```
ACFSERVE RELOAD CONTROL VMO PWPHRASE
```

Display the VMO PWPHRASE Options

Display the VMO PWPHRASE Password Phrase options defined to the system with the SHOW STATE command.

PSWD Record-Password Maintenance and Support

Record ID	Fields
PSWD	MAXTRY(1|nnn) MINPSWD(1|n) PASSLMT(2|nnn) PSWDALPH|NOPSWDALPH PSWDALT|NOPSWDALT PSWDFRC|NOPSWDFRC PSWDHST|NOPSWDHST PSWDLC|NOPSWDLC PSWDLID|NOPSWDLID PSWDMAX(*days*) PSWDMIN(*days*) PSWDMIXD|NOPSWDMIXD PSWDNCH|NOPSWDNCH PSWDNMIC|NOPSWDNMIC PSWDNUM|NOPSWDNUM PSWDPAIR(0|*n*) PSWDPLST() PSWDSPLT|NOPSWDSLPT PSWDRSV|NOPSWDRSV PSWDUC|NOPSWDUC PSWDVOWL|NOPSWDVOWL PSWNAGE|NOPSWNAGE PSWXHIST|NOPSWXHIST PSWXHST(0|*nn*) WRNDAYS(1|*nnn*)

The PSWD VMO record defines logonid password options and controls.

Fields

MAXTRY(1|nnn)

Specifies the maximum number of attempts, including the initial password entry, allowed before the terminal session is canceled. The default is one. The maximum value is 255.

MINPSWD(1|n)

Specifies the minimum number of characters required in a new password. When CA ACF2 for VM is first installed, set MINPSWD to one to allow conversion of the passwords currently in the VM directory. You can raise the minimum at a later time. The old passwords remain valid until you change them or they expire. The default is one. The maximum value is eight.

PASSLMT(2|nnn)

Specifies the maximum number of invalid password attempts allowed in a single day before CA ACF2 for VM denies all accesses by this logonid. A security administrator can issue the ACFSERVE RESET logonid command to reduce the user's invalid password violation county by one. The default setting is 2. The maximum value is 32,767.

If you do not change the PASSLMT default of 2, a user can enter two invalid passwords and, on his third invalid password, CA ACF2 for VM suspends his logonid. The user can still enter a correct password on his third attempt and gain access to the system.

Important! CA ACF2 for VM does not reset the PSWD-VIO count after a 24-hour period. When a user's PSWD-VIO count is greater than the PASSLMT, CA ACF2 for VM denies access to the system. However, a security administrator can reset this count manually or use the ACFSERVE RESET command to reset the PSWD-VIO count.

After a 24-hour period, if a user's PSWD-VIO count is less than PASSLMT, the PSWD-VIO and PSWD-DAT fields remain unchanged until a security administrator modifies them or the first password violation occurs on a new day. In the case of the first password violation, CA ACF2 for VM automatically sets the PSWD-VIO count to one and sets PSWD-DAT to the current date.

PSWDALPH|NOPSWDALPH

Specifies whether CA ACF2 for VM requires at least one alphabetic (a-z or A-Z) character to be present in a new password. The default is NOPSWDALPH, which specifies that CA ACF2 for VM will not validate the new password for at least one alphabetic character.

PSWDALT|NOPSWDALT

Specifies whether you can enter a new password at logon. The default is PSWDALT-allow password alteration. You can also change the PASSWORD field of your logonid record with the ACF CHANGE command to change your password. To prevent such changes, redefine the PASSWORD field in the @CFDE macro of the CA ACF2 for VM Field Definition Record (ACFFDR). See the *Installation Guide* for further details on defining the logonid record.

PSWDFRC|NOPSWDFRC

Specifies whether you are forced to change the password at the next logon whenever someone other than yourself (such as a security administrator or account manager) changes the password. You should not use the NOPSWDALT and PSWDFRC operands together. If you set PSWDFRC, CA ACF2 for VM uses the PSWDALT option. The default is PSWDFRC.

PSWDHST|NOPSWDHST

Specifies whether users can enter new passwords that match previous passwords. CA ACF2 for VM remembers the user's last four passwords (the current password and the previous three). The default is NOPSWDHST, which specifies that password history is not checked.

Note: PSWDHST implements the support formerly found in the sample NEWPXIT. If PSWDHST is used, remove password history checking from NEWPXIT or remove NEWPXIT. If both are present, CA ACF2 for VM remembers only two passwords instead of four.

Note: See the PSWXHIST option (extended password history) if more than four remembered passwords are desired.

PSWDLC|NOPSWDLC

Specifies whether CA ACF2 for VM requires at least one lowercase (a-z) character in a new password. The default is, NOPSWDLC, CA ACF2 for VM does not require at least one lowercase (a-z) character in a new password.

Note: Changes to this parameter take effect at the next password change of the user.

PSWDLID|NOPSWDLID

Specifies whether CA ACF2 for VM will check if a new password matches the logonid. PSWDLID specifies that new passwords will be checked and rejected if they match the logonid. The default is NOPSWDLID, which specifies that passwords are not checked for logonid match.

PSWDMAX(*days*)

Specifies the global value for the maximum number of days (based on the date specified in the PSWD-TOD field) permitted between password changes before the password expires. Any non-zero value in the LIDREC MAXDAYS field will override this value for validations. A zero in the LIDREC MAXDAYS field will also override this value if the LIDZMAX flag is also set in the LIDREC. PSWDMAX(0) specifies that there is no global value set; only the value in the LIDREC MAXDAYS field will be used for validations.

PSWDMIN(*days*)

Specifies the global value for the minimum number of days that must elapse before a user can change his password. Any non-zero value in the LIDREC MINDAYS field will override this value for validations. A zero in the LIDREC MINDAYS field will also override this value if the LIDZMIN flag is also set in the LIDREC. PSWDMIN(0) specifies that there is no global value set, only the value in the LIDREC MINDAYS field will be used for validations.

Note: If there are currently non-zero values in the MAXDAYS and MINDAYS fields for a LIDREC and you would now like the global value to apply, you must do the following:

change lidrec maxdays(0) mindays(0)

If there is currently a zero value for either field and you want the zero to apply, you must set the LIDZMAX flag for the MAXDAYS zero value to apply and the LIDZMIN flag for the MINDAYS zero value to apply.

PSWDMIXD|NOPSWDMIXD

Specifies passwords are case sensitive. PSWDMIXD is a global setting for all logonids and goes into effect as passwords are changed. The default is NOPSWDMIXD.

Once PSWDMIXD is on, existing (current) passwords are not affected. That is, they can be entered in any combination of upper and lower case characters and they will always be uppercased before password validation is performed. Once a user has change their password while PSWDMIXD is on, their password becomes case sensitive. If PSWDMIXD is turned off, their password remains case-sensitive until they set a new password while NOPSWDMIXD is set. The PSWD-MIX field in the logonid record indicates that the current password is case-sensitive.

Note: Before setting PSWDMIXD on, read "Considerations for Mixed-Case Passwords" carefully.

PSWDNCH|NOPSWDNCH

Specifies whether users can change their passwords with the ACF CHANGE command. The default is NOPSWDNCH, which permits password changes through the ACF CHANGE command. This option does not affect administrators who are changing the passwords of other users. (It does affect administrators changing their own passwords.)

The purpose of the PSWDNCH option is to require users to change their passwords only at system entry. PSWDNCH can also be used with the NOPSWDALT option to require all password changes to be done by security administrators.

PSWDNMIC|NOPSWDNMIC

Specifies that CA ACF2 for VM requires at least one numeric (0-9) character in a new password. The default is NOPSWDNMIC, which specifices that CA ACF2 for VM will not validate the new password for at least one numeric character.

PSWDNUM|NOPSWDNUM

Specifies whether CA ACF2 for VM will check if a new password is all numeric. PSWDNUM specifies that new passwords will be checked and rejected if they contain only numerics (digits 0-9). The default is NOPSWDNUM, which specifies that passwords are not checked for all numeric characters.

PSWDPAIR(0/*n*)

Specifies that only *n* (0 to 4) number of consecutively repeated pairs of characters in a new password be allowed. The default is 0, which specifies that CA ACF2 for VM will not validate the new password for consecutively repeating pairs of characters.

For example, when PSWDPAIR(1) is specified, a new password can specify up to one consecutively repeating pair of characters. So a valid new password can be 'RABBIT, 'NEEDED', or 'NEWPSWD'. However, CA ACF2 for VM will disallow 'RABBBIT' for 'BBB' is considered as two consecutively repeating pairs of characters.

Note: PSWDPAIR(4) allows eight duplicate characters, for example 'AAAAAAAA'. This specification is the same as PSWDPAIR(0).

PSWDPLST( )

Specifies that CA ACF2 for VM will allow new password to contain non-alphanumeric characters in addition to default password characters, which are alphanumeric (a-z, A-Z, 0-9) and national (@ # $). There are 12 non-alphanumeric characters it can specify. By default it specifies none.

The following are the supported non-alphanumeric characters:

Character name	Character	Notes
Asterisk	*	
Ampersand	&	
Carat	^	
Colon	:	
Equal sign	=	
Hyphen	-	
Exclamation Mark	!	
Period	.	
Percentage	%	
Question	?	
Underscore	_	
Vertical Line	|	

Note: The PSWDPLST cannot override the default characters, which are the alphanumeric and national characters.

Example: When PSWDPLST(& * -) is specified, a valid password can contain ampersand(&), asterisk(*), and hyphen(-) characters. The following are examples of valid passwords:

```
'NEW$PWD', 'NEW*PS&D', '123$PSWD', 'NEWPSWD@' or '$NEW-PSWD'
```

Note: While CA ACF2 for VM for VM and CA ACF2 for VM for z/OS and OS/390 support the special characters @, #, and $, these characters are special characters to VM. The # character is the default LINEND character. When this character is encountered in an input line, the line splits at that point. The @ is the default CHARDEL character. When this character is encountered in an input line, the previous character is deleted as if the @ was a destructive backspace. Because of this, do not use these two characters in a password that VM uses. The only way that you can use them is to place the ESCAPE character (by default, ") before the @ or #. The ESCAPE character causes the @ and # to be treated as standard characters instead of special characters.

PSWDSPLT|NOPSWDSPLT

Specifies whether a password contains a national or a user-defined character. The default is NOPSWDSPLT.

Note: If a user-defined character list, PSWDPLST, is not specified, then the new password can only contain characters from the national character set.

Example: When PSWDSPLT and PSWDPLST(& %) are specified, then a valid password must contain either a national or user-defined (& %) character. The following are examples of valid passwords:

```
'BIG&RED', 'BIG%RED', 'BIG$RED', or '456%RED'
```

Example: When PSWDSPLT and PSWDPLST() are specified, then a valid password must contain a national character. The following are examples of valid passwords:

```
'N$EWPASS', or 'NEWPAS$S'
```

PSWDRSV|NOPSWDRSV

Specifies whether users can enter new passwords that begin with a reserved word prefix. The default is NOPSWDRSV, which specifies that passwords are not checked for reserved word prefixes. The list of reserved prefixes is specified in the VMO RESWORD infostorage record.

PSWDUC|NOPSWDUC

Specifies whether CA ACF2 for VM requires at least one uppercase (A-Z) character in a new password. The default is NOPSWDUC, CA ACF2 for VM does not require at least one uppercase (A-Z) character in a new password.

Note: Changes to this parameter take effect at the next password change of the user.

PSWDVOWL|NOPSWDVOWL

Specifies whether CA ACF2 for VM will validate if a new password can specify upper or lower cased vowel (A, E, I, O, U, a, e, i, o, u) characters. If NOPSWDVOWL is specified, vowels are not allowed. CA ACF2 for VM stores the password as uppercase. The default is PSWDVOWL.

PSWNAGE|NOPSWNAGE

Specifies that if an administrator is changing a password on behalf of another user and that password will be immediately expired, the password history will not be updated to include the temporary password that the administrator assigns.

PSWXHIST|NOPSWXHIST

Specifies that Extended Password History is to be used. This is an extension of password history, which is specified using the PSWDHST field. With PSWDHST, four previous passwords are checked against the new password. You can extend this support to check up to 64 previous passwords. The number of previous passwords is determined by the value in PSWXHST(*nn*). Note that PSWDHST still works the same when NOPSWXHIST is specified. The default is NOPSWXHIST, which specifies that extended password history is not active. When PSWXHIST is on, previous passwords are stored in a PROFILE(USER) DIV(PASSWORD) record in the INFOSTG database where the record name is the same as the logonid.

NOTE: If you currently use NEWPXIT to maintain password history, be aware of the following:

- If you use NEWPXIT unmodified, then PSWDHST and PSWXHIST will work concurrently with NEWPXIT. However, there is no reason to continue using NEWPXIT because you can set PSWDHST to replace NEWPXIT immediately.
- If you have modified NEWPXIT in order to add additional user defined fields, to save additional password history, then PSWDHST and PSWXHIST will work concurrently with NEWPXIT. You can use PSWXHIST to eventually replace NEWPXIT but you need to decide when NEWPXIT can safely be removed.

  To help make this decision, let's look at an example: Suppose you have three user defined old password and TOD stamp fields in addition to the four provided by the original NEWPXIT code. That means you maintain seven old passwords altogether. If you set the VMO PSWD record to PSWXHIST PSWXHST(7) then seven old passwords will be maintained by CA ACF2 for VM. Note that PSWXHST(*nn*) could be set higher than 7 if desired. When a password is changed, NEWPXIT pushes down all the old passwords, updating the first four CA ACF2 for VM defined fields and the three user defined fields. PSWXHIST processing will leave the first four fields alone since they have already been updated by NEWPXIT, and it will save the rest of the passwords in the PROFILE(USER) DIV(PASSWORD) record for the logonid. After the user's password has been changed three times, the NEWPXIT user defined fields and the PROFILE(USER) DIV(PASSWORD) record fields will be in sync. Once this has been accomplished for all logonids with passwords, NEWPXIT can safely be removed.
- If you use NEWPXIT to do additional things in addition to maintaining password history, then instead of removing NEWPXIT once you are using PSWXHIST, simply remove the code from NEWPXIT that maintains the password history.

PSWXHST(0|*nn*)

The number of previous passwords to be retained when PSWXHIST is specified. A value of 0 or a value between 5 and 64 is valid. Values 1, 2, 3 and 4 are not valid because the password history that is maintained using PSWDHST keeps the first four previous passwords. If less than five previous passwords need to be retained, then specify PSWDHST NOPSWXHIST PSWXHST(0). The default is 0.

WRNDAYS(1|nnn)

- Specifies the number of days you are sent a warning before your password expires. CA ACF2 for VM displays this warning message on every CA ACF2 for VM access. The default is one.

SHOW Subcommand

You can display the password options in effect with the SHOW STATE subcommand.

Considerations for Mixed-Case Passwords

There are several important things to consider before setting PSWDMIXD. All applications that perform password validation must support mixed-case passwords. If an application does not support mixed-case password validation and PSWDMIXD is on, then the users of the application must type their passwords in upper-case.

The following types of password validations are supported.

- Standard VM logon, but not from the logo screen. The password field on the logo screen can be disabled and a password prompt will be issued.
- CA PAM for Linux for System z

The following is a list of the types of password validation that are not support

- FTP
- Any process that converts the password to upper-case before being passed to CA ACF2 for VM

Shared Logonid Database

If you are sharing the Logonid database with other z/VM systems, then all the systems must use either PSWDMIXD or NOPSWDMIXD. If PSWDMIXD is used, all systems must be running CA ACF2 for VM r8 or later.

If you are using Database Synchronization to share Logonids with a z/OS system protected by CA ACF2 for VM for z/OS, then all the systems must use PSWDMIXD or NOPSWDMIXD. If PSWDMIXD is used, all systems must be running CA ACF2 for VM r8 or later.

RESCLASS Record-Resource Class List

Record ID	Fields
RESCLASS	ACCOUNT(ACT) AUTOLOG(ALG) DIAL(DIA) DSPACE(DSP) GRPLOGON(GRP) IUCV(IUC) POSIXGRP(PGR) VMCF(VMC)

This record defines the type codes required for resource rule validation. If you redefine any or all of these type codes, you must store resource rules under the redefined type code. If you define new type codes, you may need to update the RESTYPE VMO record to reflect the new type codes. The RESTYPE VMO record is explained later in this chapter.

Fields

ACCOUNT(ACT)

Defines the type code for ACCOUNT resource rule validation. ACT is the default type code.

AUTOLOG(ALG)

Defines the type code for AUTOLOG resource rule validation. ALG is the default.

DIAL(DIA)

Defines the type code for DIAL resource rule validation. DIA is the default. CA ACF2 for VM performs DIAL validation on all target machines except those with the DIALBYP logonid privilege.

DSPACE(DSP)

Defines the type code for VM dataspace resource rule validation. DSP is the default type code.

GRPLOGON(GRP)

Defines the type code for GROUP LOGON resource rule validation. GRP is the default.

IUCV(IUC)

Defines the type code for IUCV and APPC/VM resource rule validation. IUC is the default.

POSIXGRP(PGR)

Defines the type code for Posix group resource rule validation. PGR is the default.

VMCF(VMC)

Defines the type code for VMCF resource rule validation. VMC is the default.

SHOW Subcommand

The SHOW STATE subcommand displays the current values in effect for the RESCLASS VMO record.

RESTYPE Record-Resident Type List Support

Record ID	Fields
RESTYPE	TYPES(ACT,ALG,DIA,DSP,GRP,IUC,PGR,VMC|$type1, $type2,...,$type64) SIZE(64k|nnnnK|nnnnM)

This record automatically builds a resident type list for the specified resource types when you initialize the service machine. You can specify up to 64 separate resource types. Resident type lists accommodate masked rule names and locate resource rules faster.

You must build a resident type list for resource types if you want to mask the $KEY parameter in resource rules. There are two ways to do this:

- You can use the RESTYPE record to automatically build a permanent list for up to 64 resource types
- You can use the ACFSERVE RELOAD RESOURCE command to manually build a temporary type list. If you change a resource rule or write a new rule, issue the ACFSERVE RELOAD RESOURCE command to activate the changed or new rule.

To build a resident type list for each of the default type codes, enter:

```
insert restype
 MYSYSID / RESTYPE LAST CHANGED BY MAINT ON 02/01/91-14:48
          SIZE(64K) TYPES(ACT ALG DIA GRP IUC PGR VMC)
```

You can also build a resident type list for any resource types you have defined locally. For example, to include the resource type LCL with the default resource types, enter:

```
change restype types(lcl) add
 MYSYSID / RESTYPE LAST CHANGED BY MAINT ON 02/01/91-14:50
          SIZE(64K) TYPES(ACT ALG DIA GRP IUC LCL PGR VMC)
```

Fields

TYPES=(ACT,ALG,DIA,DSP,GRP,IUC,PGR,VMC|$type1, $type2,...,$type64)

Specifies the three-character resource type code of each resident type automatically built during initialization of the CA ACF2 for VM service machine. Each type code corresponds to the three-character $TYPE parameter specified in a set of related resource rules. You can specify up to 64 type codes.

SIZE(64K|nnnnK|nnnnM)

Specifies the amount of storage reserved for the resource rule cache. This parameter requires a suffix of K or M. Use K to specify K bytes, and M to specify megabytes. The resource rule cache is in the CA ACF2 for VM service machine. It also improves overall system performance when the CA ACF2 for VM resource support protects resources CA ACF2 for VM or your site defined. The default is SIZE=64K. The minimum value is 16K. The maximum value is 14336K (14Mb), but we recommend that you use 4096K (4Mb) as the maximum value. Do not specify a cache size less than 16K. An extremely large cache might require a larger storage area for the CA ACF2 for VM service machine. You can also specify a null size (SIZE=()) to prevent use of a resource rule cache. You cannot change this value with an ACFSERVE RELOAD command. Instead, use ACFSERVE RESTART to modify it.

You can use ACFSERVE commands to delete a resident type list, create a new resident type list, and to reload a resident type list. See the “Using the ACFSERVE Commands” chapter for more information and command syntax.

SHOW Subcommand

The SHOW STATE subcommand displays the names of the resident resource types that currently reside on the service machine. For more detailed information on the types of rule caches and the RESTYPE VMO record, see the *Systems Programmer Guid*e.

RESWORD - Reserved Word Prefix List

Record ID	Fields
RESWORD	PREFIXES(APPL, APR, AUG, ASDF, BASIC, CADAM, DEC, DEMO, FEB, FOCUS, GAME, IBM, JAN, JUL, JUN, LOG, MAR, MAY, NET, NEW, NOV, OCT, PASS, ROS, SEP, SIGN, SYS, TEST, TSO, VALID, VTAM, XXX, 1234|*prefix1,…,prefix255*)

The RESWORD record defines the words or prefixes that are not allowed in the specification of a password. In the case of mixed-case passwords, the password is temporarily upper-cased before being checked against the RESWORD prefixes.

Field Descriptions

PREFIXES

Specifies up to 256 password prefixes. Specify from one to eight characters. CA ACF2 for VM provides a default RESWORD record. You can change the values or supply others up to a total of 256. The list is sorted in collating sequence for ease of display.

Valid prefixes for the RESWORD list can include acronyms for the company or industry, names of software systems, terminal ID prefixes, and so forth.

Following is the default reserved word prefix list:

APPL	IBM	PASS
APR	JAN	ROS
AUG	JUL	SEP
ASDF	JUN	SIGN
BASIC	LOG	SYS
CADAM	MAR	TEST
DEC	MAY	TSO
DEMO	NET	VALID
FEB	NEW	VTAM
FOCUS	NOV	XXX
GAME	OCT	1234

SHOW Subcommand

The SHOW RESWORD subcommand displays the current reserved word prefixes in effect.

RULEOPTS Record-CA ACF2 for VM Rule Option Specifications

Record ID	Fields
RULEOPTS	CENTRAL|NOCENTRAL CHANGE|NOCHANGE COMPDYN|NOCOMPDYN DECOMP(SECURITY,AUDIT|privilege list) RULELONG|NORULELONG VOLRULE|NOVOLRULE $NOSORT\|NO$NOSORT

This record defines the options that determine how CA ACF2 for VM uses and maintains command limiting rules, resource rules, and access rules.

Fields

CENTRAL|NOCENTRAL

Specifies whether the data owner has authority to store access rules. If you specify CENTRAL, only security administrators and users with the %CHANGE or %RCHANGE feature have this privilege. You can combine the NOSTORE attribute of the logonid record with NOCENTRAL to let only selected users update their own rules. The default is NOCENTRAL (all users can update the access rule sets for their own disks and files).

CHANGE|NOCHANGE

Specifies whether %CHANGE and %RCHANGE are recognized. If you specify NOCHANGE, CA ACF2 for VM ignores any %CHANGE or %RCHANGE statements in a rule set when it determines if a user can replace any access rule. The default is CHANGE, activating %CHANGE and %RCHANGE authorization.

COMPDYN |NOCOMPDYN

Specifies whether to compile a rule set with the 32K (RULELONG) compiler. With this option, CA ACF2 for VM defaults to use the 4K rule format compiler and will dynamically switch to use the 32K compiler if the 4K compiler fails due to an out-of-buffer condition or because a rule option was used that requires the 32k compiler. For example, the rule set is too large to fit in a 4K buffer. The default is NOCOMPDYN.

Note: If COMPDYN is specified, RULELONG also needs to be specified. At startup or when the RULEOPTS records is reloaded, COMPDYN will be disabled if RULELONG is not active.

DECOMP(SECURITY,AUDIT|privilege list)

Specifies the logonid attributes necessary to decompile an access rule or resource rule that you do not have the authority to change. The default is SECURITY, AUDIT (users with either attribute are authorized). This authority applies to all scoped or unscoped users with the privileges listed in this field. You can only specify CA ACF2 for VM-defined privileges in this field. Scopes do not affect the privileged attributes listed in the DECOMP field.

RULELONG|NORULELONG

Specifies whether you want to use rules greater than 4K long. NORULELONG, the default, indicates that rules are compiled and stored in the current format with a limit of 4K. RULELONG indicates a formatted access and resource rule record capable of expanding greater than 4K, to a maximum of 32K. The Rules and Infostorage databases must be redefined to accommodate this greater length. RULELONG also requires the use of VSAM databases. If you do not increase the size of these databases, this option is disabled. Message ACFpgm47EW displays during CA ACF2 for VM startup or during a refresh of the RULEOPTS record if this option is disabled. Once you set this option, the Rules and Infostorage databases are no longer compatible with previous releases of CA ACF2 for VM.

Note the following:

- While RULELONG allows for more rule lines and new options, the rules compiled by the RULELONG compiler are also larger, often twice as large. To prevent rules that do not need to use the RULELONG compiler from being expanded for no reason, enable the Dynamic Compile (COMPDYN) option.
- If the Dynamic Compile option (COMPDYN) is also enabled, RULELONG will be allowed even if CMS databases are used, even though the maximum rule size will still be 4K. The only reason to enable RULELONG and COMPDYN when using CMS databases would be to use a rule option that requires RULELONG. Currently, ACTIVE is the only option that requires RULELONG. However, if an option that requires RULELONG is added to a rule that is near or over 2K in length, it will most likely not fit within the 4K CMS database limit. If this happens, NEXTKEY can be used to split the rule.

VOLRULE|NOVOLRULE

Specifies that CA ACF2 for VM validates access to tape volumes listed in the TAPEVOLS list through tape volume access rules. This operand defines the correct syntax for these rules. The values for this operand are defined below.

VOLRULE

Creates rules in the VOLUME.@volser format. You must specify the $KEY value as VOLUME. You must specify the filename as the tape volume identifier (volser) with the prefix @. You only need to create one rule set with this format. For example, if you specify 767305 and 123*** as the secured volume list in the TAPEVOLS record, the following rule set gives any user with the SVM UID mask write access to these volumes:

```
$KEY(VOLUME)
 @767305 UID(*****svm) WRITE (A)
 @123aaa UID(*****svm) WRITE (A)
```

NOVOLRULE

Creates rules in the @volser.VOLUME format. It is the default. You must specify the $KEY value as the tape volume identifier (volser) with the prefix @. You must specify the filename as VOLUME. You must create a separate rule set for each tape volume with this format. For example, if you specify 767305 and 123*** as the secured volume list in @SECVOLS, the rule sets below give any user with the SVM UID mask write access to these volumes:

```
$KEY(@767305)
 VOLUME UID(****svm) WRITE (A)

$KEY(@123aaa)
 VOLUME UID(****svm) WRITE (A)
```

$NOSORT|NO$NOSORT

Specifies whether the $NOSORT rule set control statement is active. If you specify $NOSORT, CA ACF2 for VM recognizes the $NOSORT control statement during rule compilation and suppresses normal CA ACF2 for VM rule sorting (from most specific to most general). If you specify NO$NOSORT (the default), CA ACF2 for VM ignores any $NOSORT control statements and automatically sorts rule sets during rule compilation.

SHOW Subcommand

The SHOW STATE subcommand displays the values currently in effect for the RULEOPTS record.

SAFDEF Record-SAF Call Environments

Record ID	Fields
SAFDEFqual	ID(name) FUNCRET(4|retcode) FUNCRSN(0|rsncode) JOBNAME(mask|********) MODE(IGNORE|GLOBAL|LOG|QUIET) PROGRAM(mask|********) RACROUTE(keyword=value,...,keyword=value) RB(mask|********) RETCODE(0|4|8) USERID(mask|********)

The SAFDEF record defines the SAF environment and instructs CA ACF2 for VM how to process a SAF call. CA ACF2 for VM provides internal SAFDEFs for SAF default protection. Both internal and external SAFDEFs display when you issue a SHOW SAFDEF command.

You can use the SAFDEF record to override how CA ACF2 for VM processes SAF calls. CA ACF2 for VM performs validation based on the environment you define in this record. You can create multiple SAFDEF records.

Fields

ID(name)

Specifies an ID name associated with the record. You can specify up to eight characters. This field is optional. We recommend you specify an ID because this name will appear as the first field displayed in the SHOW SAFDEF output.

Select a name that is unique and that conveys meaning about the type of SAF call you are defining. For example, VERSMS would be an appropriate ID for a SAFDEF record that defines the environment for a REQUEST=VERIFY call from DFSMS.

FUNCRET(4|retcode)

Specifies the SAF function-dependent return code to be returned to the caller making the RACROUTE request when MODE is specified as IGNORE. For detailed descriptions of these return codes, refer to the IBM document entitled External Security Interface (RACROUTE) Macro Reference Guide. The default is four.

FUNCRSN(0|rsncode)

Specifies the SAF function-dependent reason code to be returned to the caller making the RACROUTE request when MODE is specified as IGNORE. For detailed descriptions of these reason codes, refer to the IBM document entitled External Security Interface (RACROUTE) Macro Reference Guide. The default is zero.

JOBNAME(mask|******)**

Specifies the user ID of the service machines that apply to this SAFDEF record. You can specify an eight-character user ID or a mask. The default is all service machines.

MODE(IGNORE|GLOBAL|LOG|QUIET)

Specifies the mode you want CA ACF2 for VM to use to process this SAF request. This field defaults to GLOBAL; a value is required. You can specify any one of the following values:

IGNORE

Bypass processing this SAF request.

GLOBAL

Process this SAF request with the mode specified in the VMO OPTS record. For generalized resource validations, use the CA ACF2 for VM recommendation to allow or deny the SAF request.

LOG

Upon return of the validation call, allow access even if access is denied. LOG does not force logging if a logonid is allowed access.

QUIET

Process this REQUEST=AUTH call in QUIET mode.

PROGRAM(mask|******)**

This field is not active for VM sites. For more information for this field, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guid*e.

RACROUTE(keyword=value,...,keyword=value)

Identifies the SAF request being made. Use this field to specify any valid RACROUTE parameters and values. This is a multi-value field. The maximum length that you can specify for the parameter keyword, operator, and value is 64 characters. Separate the entries with commas or blanks.

Note: There are SAFDEF restrictions with FASTAUTH processing. FASTAUTH does not allow the use of ENTITY on the RACROUTE field.

You can specify the following relational operators (depending on your type of keyboard) to indicate the presence of a particular value (for example, ENVIR=CREATE) or the presence of a pointer address (ACEE=>).

```
=   Equal to
¬=  Not equal to
<>  Not equal to
!=  Not equal to
=>  Pointer value
¬=>  No pointer value
!=>  No pointer value
```

Pointer values are valid only if the keyword operand is specified as a pointer to a data area or data structure (for example, ACEE). When you specify a pointer value, do not also specify a value for the operand. For example, the following request defines a VERIFY request for all user IDs except JOHN, where an ACEE is supplied:

```
RACROUTE(REQUEST=VERIFY,ACEE=>,USERID¬=JOHN)
```

You can mask character data types using the standard CA ACF2 for VM masking characters (asterisk and dash). You can mask other types of data only if the mask is complete. A complete mask indicates that the parameter matches all values. For example, you can specify the following to indicate that this parameter matches all values of USERID:

```
USERID=-
```

Whereas, the following indicates that the USERID option does not apply to this RACROUTE request.

```
USERID¬=-
```

RB(mask|******)**

This field is not active for VM sites. For more information for this field, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guid*e.

RETCODE(0|4|8)

Specifies the SAF return code to be returned to the caller making the RACROUTE request when MODE is specified as IGNORE.

0

Allow the request.

4

Let the caller decide how to process the request.

8

Deny the request.

The default is four.

USERID(useridmask|******)**

Specifies the user ID that applies to this SAFDEF record. The default is all user IDs.

Creating Multiple VMO SAFDEF Records

To create multiple SAFDEF records, append a qualifier to the record name in the format SAFDEFqual so that you can define a unique record ID for that SAFDEF record for a particular SYSID. The RECID can be a maximum of 16 bytes. Therefore, you can specify a qualifier of up to ten characters. It must immediately follow the characters SAFDEF. If you use a period (.) as part of the qualifier string for the record name, CA ACF2 for VM counts it as one of the ten available characters.

For example, if you want to create a SAFDEF record for a VERIFY call from HSM and an AUTH call from HSM, you must use a qualifier to distinguish these two records. You could name the SAFDEF record for the VERIFY call SAFDEF.VERHSM and the SAFDEF record for the AUTH call SAFDEF.AUTHHSM. Naming records using qualifiers enables you to describe multiple unique environments and allows CA ACF2 for VM to add those records to the Infostorage database with unique identifiers.

CA ACF2 for VM processes all SAF calls by default. To override the default SAF processing for a specific security event, you can specify a SAFDEF record.

SHOW Subcommand

The SHOW SAFDEF and SHOW ALL subcommands display the values in the SAFDEF record.

SAFDEF Record Example

This section shows a simple SAFDEF record that you can create when you want CA ACF2 for VM to ignore validating the request.

Ignoring SAF Calls

A basic SAFDEF record that you may have to create is shown below:

```
set control(VMO)
 CONTROL(VMO)
list like(safdef-)
 PRD1 / SAFDEF.XYZ LAST CHANGED BY USER01 ON 07/03/00-12:13
           ID(AUTHXYZ) MODE(IGNORE) RETCODE(0)
           PROGRAM(********) JOBNAME(XYZ-)
           RACROUTE(REQUEST=AUTH,CLASS=DATASET)
```

In this example, the SAFDEF record is for the XYZ product. The products makes a SAF call that CA ACF2 for VM intercepts. In this case, suppose the SAF call is a RACROUTE REQUEST=AUTH,CLASS=DATASET. The site decided to instruct CA ACF2 for VM to bypass processing of this request because it did not want CA ACF2 for VM to validate these calls.

By specifying MODE(IGNORE) and RETCODE(0), CA ACF2 for VM allows the product XYZ to access the data set without creating a logging record. The site does not have to create a rule.

SSFTYPE Record-Standard Security Facility Protection

Record ID	Fields
SSFTYPE	ACTAP(ACA|$type) ACTDI(ACD\|$type) ACTLS(ACL|$type) ACTME(ACM\|$type) ACTPA(ACP|$type) ACTQU(ACQ\|$type) ACTRP(ACR|$type) CACMD(CAC\|$type) CADOC(CAD|$type) CALIBMEM(LBM\|$type) JOBNAME(JOB|$type) PANEL(PAN\|$type) RECIPID(REC|$type) RESOURCE(NULL\|class/type1,class/type2,...,class/tpe20) SUBMIT(SUB\|$type) SCHEDULE(SCH|$type) STATION(STA\|$type) UNVEDIT(UNE|$type) UNVRPRT(UNR\|$type) UNVPGM(UNP|$type) VMANAPPL(VMA\|$type) VTRMNODE(VTR|$type)

The SSFTYPE VMO record translates SSF resources to CA ACF2 for VM resource types. The CA ACF2 for VM resource type is the value that you need to put into the $TYPE control statement in any resource rule you write for the product. For example, resource SUBMIT has a default type of SUB. With CA ACF2 for VM, you can set a unique resource type for each SYSID. This allows you to have separate resource rules in a multi-CPU shared database environment.

Fields

ACTAP(ACA|$type)

ACTDI(ACD|$type)

ACTLS(ACL|$type)

ACTME(ACM|$type)

ACTPA(ACP|$type)

ACTQU(ACQ|$type)

ACTRP(ACR|$type)

CACMD(CAC|$type)

CADOC(CAD|$type)

CALIBMEM(LBM|$type)

JOBNAME(JOB|$type)

PANEL(PAN|$type)

RECIPID(REC|$type)

RESOURCE(NULL|class/type1,class/type2,...class/type20)

SUBMIT(SUB|$type)

SCHEDULE(SCH|$type)

STATION(STA|$type)

UNVEDIT(UNE|$type)

UNVRPRT(UNR|$type)

UNVPGM(UNP|$type)

VMANAPPL(VMA|$type)

VTRMNODE(VTR|$type)

SHOW Subcommand

The SHOW STATE subcommand displays the SSFTYPE listing.

TAPEVOLS Record-Tape Level Protection

Record ID	Fields
TAPEVOLS	VOLMASK(volmask1,...,volmask255)

This record defines the tape volumes where CA ACF2 for VM provides volume-level protection.

Fields

VOLMASK(volmask1,...,volmask255)

Specifies a maximum of 255 volume serial masks, up to six characters each. Standard CA ACF2 for VM masking conventions are supported. The default of this record is null (no volume level protection).

You must specify at least one volser if you want CA ACF2 for VM volume-level protection. You can specify up to 255 VOLMASKs. When you define a volume in the TAPEVOLS record, CA ACF2 for VM generates a pseudo data set name of @volser.VOLUME to validate an access request to that volume. However, if you specify the VOLRULE field in the RULEOPTS VMO record, the pseudo data set name is generated as VOLUME.@volser.

SHOW Subcommand

The SHOW STATE subcommand displays the volume serial number of all volumes specified in the TAPEVOLS VMO record.

WARN Record-System WARN Mode Message

Record ID	Fields
WARN	MSG(msgtext)

This record defines the text of the warning message (message ID 245W) displayed on the terminal when a violation takes place and CA ACF2 for VM is in WARN mode. CA ACF2 for VM message 245W contains a variable. The WARN VMO record text contains the substitution value for that variable. For more information on message 245W, see the *Messages Guid*e. For information on national language support for CA ACF2 for VM messages, see the *Systems Programmer Guid*e.

Fields

MSG(msgtext)

Specifies text of the message (up to 124 characters in single quotes). The default warning message is:

```
AFTER JULY 1, 1999 THIS ACCESS WILL NOT BE ALLOWED.
```

You can specify one string per system.

SHOW Subcommand

The SHOW subcommand does not display this value. You can list this record with the LIST subcommand under the CONTROL(VMO) setting.

SYSID Concepts

The SYSID is a string of one-to eight- characters that group VMO records. You can define the content of the SYSID. Use the SYSID to group VMO records by system in a shared database environment. When you start CA ACF2 for VM, the SYSID you select remains in effect until you change it with the ACFSERVE SET SYSID command. The initial SYSID selection logic follows:

1. Specify a SYSID in response to the CA ACF2 for VM startup prompt to set the startup SYSID during system startup. The format of this reply is SYSID(sysid). Any SYSID you enter is saved in the startup options file if RPROMPT or IPROMPT is YES.
2. If you did not set the SYSID, CA ACF2 for VM uses the value specified in the ACFFDR @SYSID macro. If this value is null, CA ACF2 for VM uses the value defined in the SYSID macro HCPSYS.
3. After system startup is complete, use the ACFSERVE SET SYSID command to change the current SYSID.

Under the CONTROL(VMO) setting of the ACF command, CA ACF2 for VM selects a SYSID for the session according to the following logic:

1. When a user establishes the CONTROL(VMO) setting, CA ACF2 for VM selects the current CA ACF2 for VM SYSID string (obtained originally as described above) as the default SYSID for the session.
2. Use the SYSID and MSYSID parameters of the ACF SET subcommand to change the session default. MSYSID is similar to SYSID, but indicates the use of SYSID masking. For example, you can start the system with a SYSID string of CPU1, but you can enter the following command to begin an ACF command session:

   ```
   SET CONTROL SYSID(cpu2)
   ```

As a result of this subcommand, the default SYSID becomes CPU2 for the duration of the CONTROL(VMO) setting.

You can specify the SYSID or MSYSID parameter with any of the ACF subcommands under the CONTROL(VMO) setting (INSERT, CHANGE, LIST, and DELETE). These parameters override the SYSID default for the execution of that command only. For example, assume the default SYSID is CPU1 when you enter the following:

```
CHANGE PSWD MINPSWD(5) SYSID(cpu2)
```

The MINPSWD field change applies to the PSWD VMO record defined for CPU2, not for the VMO record defined for the system CPU1.

3. The SHOW SYSTEMS subcommand displays the ACFFDR SYSID, STARTUP SYSID, and CURRENT SYSID.

Multiple System SYSID Use

Sites using multiple systems can use a different SYSID name for each system to secure each system. This lets you create a different set of VMO records for each SYSID, or you can share one or more VMO records between systems.

You define the naming conventions for the SYSID. For example, in the ACFFDR of each system, you can specify:

```
@SYSID CPU1
@SYSID CPU2
```

To share a VMO record between multiple CPUs, use masking characters in the SYSID field. For example, to share the VMO record between CPU1 and CPU2, set the SYSID to the CPU ID that the OPTS VMO record controls. You can create a new VMO record through one of the VMO records as an example. The new record uses a SYSID that contains masked characters. You can then delete the VMO records for CPU1 and CPU2 with the ACF command.

```
ACF
SET CONTROL(VMO) SYSID(cpu*)
INSERT OPTS USYSID(cpu1)
DELETE OPTS SYSID(cpu1)
DELETE OPTS SYSID(cpu2)
END
```

Using MSYSID

Sites using multiple SYSIDs to secure their systems can use the standard CA ACF2 for VM masking technique and a special parameter (MSYSID) with the CHANGE subcommand to change one VMO record for multiple SYSIDs.

```
ACF
SET CONTROL(VMO)
CHANGE OPTS MSYSID(cpu*) DATE(mdy)
END
```

The MSYSID parameter applies the change to the VMO record for all SYSIDs that begin with CPU and end with any alphanumeric character. You can also apply this parameter to the LIST subcommand. To list the OPTION records for all SYSIDs, enter the following commands:

```
ACF
SET CONTROL(VMO)
LIST MSYSID(-) OPTS
END
```

To list all MAINT records for all SYSIDs, use the following command:

```
ACF
SET CONTROL(VMO)
LIST MSYSID(-) LIKE(MAINT-)
END
```

Because MAINT records are qualified, enter the specific name with the qualifier or use the LIKE parameter with a mask.

Using the “?”

Use a question mark (?) as a SYSID value to indicate that you want the currently active SYSID value for this system.

For example, a security administrator enters the VMO environment, using a SYSID of TEST. Commands entered at this point reference records with a SYSID of TEST.

```
ACF
SET CONTROL(VMO) SYSID(test)
```

However, the security administrator remembers that he has to change a record used in the production, not test, environment. He uses a SYSID of **?** to reference the active system ID value, in this case PROD.

```
CHANGE SYSID(?) OPTS MAXVIO(5)
```

Maintaining VMO Options

After you establish the CONTROL(VMO) setting, you can insert, change, list, and delete VMO records. After you process VMO records, you must issue an ACFSERVE RELOAD command, IPL your system, or restart the service machine to apply the affected records to the system. Refer to the chart for a complete list of all VMO records and the action you must take to enable the option. Use CA ACF2 for VM SHOW subcommands to verify the values in effect.

In addition to using the ACF command to maintain VMO options as described below, the ACFFS Full Screen ACF command can be used to maintain VMO records using full screen panels. See the chapter "Using the Full-Screen Feature" for more information on the ACFSS command.

ACF Subcommands under the CONTROL(VMO) Setting

You can process VMO records after you establish the CONTROL(VMO) setting of the ACF command and specify the type code VM through the SET CONTROL (VMO) command.

After you establish the CONTROL(VMO) setting, you can issue any of the following subcommands:

- CHANGE
- EXEC
- SET
- CMS
- HELP
- SHOW
- CP
- INSERT
- XEDIT
- DELETE
- LIST

- END
- QUIT

The subcommands END, QUIT, and HELP operate under the CONTROL(VMO) setting as described earlier in this guide. The other subcommands are described in the following text. SET and SHOW are common subcommands. They are described in this chapter because they operate differently under the CONTROL(VMO) setting than under other ACF command settings.

Setting the CONTROL(VMO) Mode

Use the SET subcommand to establish the CONTROL(VMO) setting to process VMO records. The SET subcommand also establishes the active SYSID and other parameters that affect how certain ACF subcommands operate. The syntax for the SET subcommand under the CONTROL setting is:

```
SEt Control(Vmo) [ SYSid(sysid)      ]
                 [ MSYSid(sysidmask) ]
```

The SYSID and MSYSID parameters are described below. The other parameters operate as previously explained.

SYSID(sysid)

Specifies the SYSID of the VMO records processed under the CONTROL(VMO) setting. This SYSID remains active for the duration of the CONTROL(VMO) setting, but you can override it with the SYSID parameter of individual ACF subcommands. If you do not specify the SYSID parameter when you establish the CONTROL(VMO) setting, CA ACF2 for VM uses the system default. You can abbreviate the SYSID parameter name to a minimum of SYS.

MSYSID(sysidmask)

Specifies a mask to indicate the SYSIDs of the VMO records processed under the CONTROL(VMO) setting. These SYSIDs remain active for the duration of the CONTROL(VMO) setting, but you can override them with the SYSID parameter of individual ACF subcommands. If you do not specify the SYSID parameter when you establish the CONTROL(VMO) setting, CA ACF2 for VM uses the system default. You can abbreviate the MSYSID parameter name to a minimum of MSYS.

Creating VMO Records

The INSERT subcommand under the CONTROL(VMO) setting creates a VMO record to define system options. The syntax for this INSERT subcommand is:

```
       { *                          }
Insert { recordname                 }
       { USING(oldrecord newrecord) }

       [ SYSid(?|sysid)             ]
       [ USYSid(?|sysid)            ]
       [ field1,field2,...,fieldn   ]
       [ ADD|REP|DEL                ]
```

Issue the INSERT subcommand with the SYSID, record ID, and fields defined in the record:

```
ACF
SET CONTROL(VMO)
INSERT SYSID(cpu1) PSWD MAXTRY(2) MINPSWD(5)
```

This sample subcommand inserts the PSWD VMO record. It defines the password options MAXTRY and MINPSWD. When CA ACF2 for VM is running under the SYSID of CPU1, users have a maximum of two attempts to enter the correct password before the session is canceled. All passwords must contain at least five characters. If you do not specify the SYSID parameter, options are defined for the default SYSID.

The INSERT subcommand has the following parameters:

*****

Specifies that you want to use the same recordname as the last record processed since you established the CONTROL(VMO) setting. The asterisk does not work with multiple records or masking.

recordname

Specifies the name of the VMO record being inserted. You cannot mask this field. Acceptable values are:

- APPLDEFqual
- BACKUP
- CLASMAPqual
- CMDLIM
- DBSYNC
- DIAGLIM
- EXITS
- LINUXqual
- MAINTqual
- PSWD
- OPTS
- RESCLASS
- RESTYPE
- RESWORD
- RULEOPTS
- SAFDEFqual
- SSFTYPE
- TAPEVOLS
- WARN

USING(oldrecord newrecord)

Identifies a model record to create the new VMO record. All values from the model record are inserted into the new record. Any other fields and values specified in the INSERT subcommand add to or replace the fields and values in the new record. If you specify the USYSID parameter, the USING parameter is not required. (CA ACF2 for VM assumes that you want to copy the record for the SYSID specified in the USYSID parameter, but with the same record identification as the record you are inserting.)

USYSID(?|sysid)

Specifies the SYSID for a model record to create the new VMO record. You can specify the ID for this record ID with the USING or RECID parameter. If you specify a question mark (?) instead of a SYSID, the currently active SYSID is used. This time-saving feature inserts a record similar to a record previously defined for another SYSID. You can change any fields and values that differ with the INSERT or CHANGE subcommand. You cannot mask the SYSID in this parameter. USYS is an acceptable abbreviation for the name of the USYSID parameter.

SYSID(?|sysid)

Specifies the one- to eight-character SYSID where this record applies. A question mark (?) indicates that you want to use the currently active SYSID for this system. If you do not specify this operand, CA ACF2 for VM uses the default SYSID when you establish the CONTROL(VMO setting or the SET SYSID(sysid) subcommand for this session. SYS is an acceptable abbreviation for SYSID.

field1,field2,...,fieldn

Specifies the fields and values added to the new record (or to replace fields copied from a model record). For the specific field names in each record, refer to the previous description of the VMO records and their fields. These rules apply to field names:

- State the field name to turn on bit fields. Prefix the field name with NO to turn them off.
- Specify the value in parentheses for fields with variable or character values. For example, TIME(12:30).
- Specify the multiple values in parentheses for fields that are defined with the capacity to contain multiple values. Use a space or comma as a delimiter in the parentheses.

ADD

Indicates that the specified fields and values are added to those copied from the model record. If a value already exists in a VMO field, the new value replaces the copied value. If the field allows multiple values, the new value is added to the existing values. ADD is the default. If you do not specify the ADD, REP, or DEL parameters, ADD is assumed.

REP

Indicates that any value specified for a field completely replaces the value of that field as copied from the model record. If any specified field was not copied from the model record, that field and its value are added to the record.

DEL

Deletes the specified field and its value from the newly inserted record. If a field contains multiple values, only the specified values are deleted from the field.

Changing VMO Records

Use the CHANGE subcommand under the CONTROL(VMO) setting to change existing VMO records. The syntax for this CHANGE subcommand is:

```
       { *                }
CHange { recordname       }field,...,field
       { LIKE(recordmask) }[ADD|DEL|REP]

       [ SYSid(sysid)      ]
       [ MSYSid(sysidmask) ]
```

The following CHANGE subcommand alters the BACKUP VMO record so that the time of the automatic backup is scheduled for 02:00:

```
ACF
SET CONTROL(VMO)
CHANGE BACKUP TIME(02:00)
```

Because we did not specify a SYSID, the subcommand affects the BACKUP VMO record for the default SYSID. The CHANGE subcommand has the following parameters:

*****

Specifies that you want to use the same recordname as the last record you processed since you established the CONTROL(VMO) setting. The asterisk does not work with multiple records or masking.

SYSID(sysid)

Specifies the SYSID of the VMO record to be changed. If you specify a question mark (?) as the SYSID value, CA ACF2 for VM uses the default value for the system. Any value you specify with the SYSID parameter can contain asterisks or a trailing dash. These characters are treated as part of the SYSID itself and not as an indication of masking. If you specify the SYSID parameter, you cannot specify the MSYSID parameter. SYS is an acceptable abbreviation for the parameter name SYSID.

MSYSID(sysidmask)

Specifies a mask to indicate the SYSIDs where the specified VMO record is changed. This SYSID mask must contain at least one asterisk or a trailing dash. VMO records changed with this parameter have the same record ID, but different SYSIDs. MSYS is an acceptable abbreviation for the parameter name MSYSID.

recordname

Specifies the record ID to be changed.

LIKE(recordmask)

Specifies a mask for the VMO record IDs to be listed. You cannot abbreviate the LIKE parameter. Masking follows the same conventions that apply to logonids.

ADD

Indicates that CA ACF2 for VM adds any fields and values specified with the CHANGE subcommand to the existing fields and values in the specified VMO record. If a field already exists in a record, then it is not added. However, if a field takes multiple values, then the specified values are added to the existing values in the field. If you do not specify the ADD, REP, or DEL parameters, ADD is assumed.

REP

Indicates that any fields specified with this CHANGE subcommand completely replace the corresponding field in the specified VMO record. If a specified field does not currently exist in the record, that field and value is added to the record.

DEL

Indicates that any fields and values specified with this CHANGE subcommand are deleted from the specified VMO record. If any specified value does not exist in a field, the field remains unchanged.

field,...,field

Specifies the fields and any values to change as the ADD, REP, or DEL parameter directs. For the specific field names in each record, refer to the previous description of the VMO records and their fields.

Listing VMO Records

Use the LIST subcommand under the CONTROL(VMO) setting to list the contents of a specified VMO record. The syntax for this LIST subcommand is:

```
      { *                }
List  { recordname       }
      { LIKE(recordmask) }

      [ SYSid(sysid)      ]
      [ MSYSid(sysidmask) ]
```

The LIST subcommand has the following parameters:

*****

Specifies that you want to use the same recordname as the last record you processed since you established the CONTROL(VMO) setting. The asterisk does not work with multiple records or masking.

recordname

Specifies a one- to eight-character ID of the VMO record listed.

LIKE(recordmask)

Specifies a mask for the VMO record IDs listed. You cannot abbreviate the LIKE parameter. This masking follows the same conventions as logonids.

SYSID(sysid)

Specifies the SYSID of the VMO listed. If you specify a question mark (?) as the SYSID value, CA ACF2 for VM uses the default value for the system. Any value you specify with the SYSID parameter can contain asterisks or a trailing dash. These characters are treated as part of the SYSID itself and not as an indication of masking. If you specify the SYSID parameter, you cannot specify the MSYSID parameter. SYS is an acceptable abbreviation for the parameter name SYSID.

MSYSID(sysidmask)

Specifies a mask to indicate the SYSIDs where the specified VMO record is listed. This SYSID mask must contain at least one asterisk or a trailing dash. VMO records listed with this parameter have the record ID, but different SYSIDs. MSYS is an acceptable abbreviation for the parameter name MSYSID.

Deleting VMO Records

Use the DELETE subcommand under the CONTROL(VMO) setting to delete VMO records. The syntax for this DELETE subcommand is:

```
       { *                }
DELete { recordname       }
       { LIKE(recordmask) }

       [ SYSid(?|sysid)    ]
       [ MSYSid(sysidmask) ]
```

The DELETE subcommand has the following parameters:

*****

Specifies that you want to use the same recordname as the last record you processed since you established the CONTROL(VMO) setting. The asterisk does not work with multiple records or masking.

recordname

Specifies a one- to eight-character ID of the VMO record to be listed.

LIKE(recnamemask)

Specifies a mask for the VMO record IDs to be deleted. This masking follows the same conventions as logonids. Use caution when deleting multiple VMO records with this parameter.

SYSID(?|sysid)

Specifies the SYSID of the VMO record to be deleted. If you specify a question mark (?) as the SYSID value, CA ACF2 for VM uses the default value for the system. Any value you specify with the SYSID parameter can contain asterisks or a trailing dash. These characters are treated as part of the SYSID itself and not as an indication of masking. If you specify the SYSID parameter, you cannot specify the MSYSID parameter. You can abbreviate the SYSID parameter to a minimum of SYS.

MSYSID(sysidmask)

Specifies a mask to indicate the SYSIDs where the specified VMO record is deleted. This SYSID mask must contain at least one asterisk or a trailing dash. Use caution when you delete multiple VMO records with this parameter. You can abbreviate the MSYSID parameter to a minimum of MSYS.

Displaying VMO Records

Use the SHOW subcommand under the CONTROL(VMO) setting to display a currently-active SYSID and fields of a VMO record. As it does under other settings, the SHOW subcommand under the CONTROL(VMO) settings also displays many of the system options in effect (For more information on the ACF SHOW subcommand, see the “Using the ACF Command” chapter.) The syntax of the SHOW subcommand for VMO records is:

```
      { ACF2                    }
      { ACTive                  }
      { ALL                     }
      { APPLDEF                 }
      { BACKUP                  }
      { CLASMAP                 }
      { CMDLIM                  }
      { DIAGLIM                 }
SHow  { Fields[(recordname)]    }
      { LINUX[(linuxopts)]      }
      { MAINT                   }
      { Mode                    }
      { RESWORD                 }
      { SAFDEF                  }
      { STate                   }
      { SYstems                 }
```

The SHOW subcommand accepts only one parameter at a time. All parameters (except for MODE and FIELDS) operate under the CONTROL(VMO) setting as described in the description of the SHOW subcommand in Chapter 3. The MODE and FIELDS parameters are described below.

MODE

The SHOW MODE command displays the current setting of the ACF command, the VM type code, and the currently active SYSID:

```
ACF
SET CONTROL(VMO)
SHOW MODE

MODE: CONTROL  TYPE: VMO  SYSID: CPU1
```

FIELDS[(recordname)]

The SHOW FIELDS subcommand displays the names of the fields in the logonid record or the specified VMO record. The SHOW subcommand also indicates which fields you can change. An asterisk (*) shows single-valued fields that you can change and the plus sign (+) shows multivalued fields you can change.

VM-Related Commands

Several ACFSERVE commands process VMO records and related VM facilities. For more information on the ACFSERVE commands, see the “Using the ACFSERVE Commands” chapter.

Part II: Defining User Privileges for ACFSERVE Commands

Structured infostorage records for ACFSERVE privileges now controls the ACFSERVE command. These records are similar to the VMO records. This section describes ACFSERVE privilege records and how to define them.

To define ACFSERVE privilege records, issue the SET CONTROL(ACFSERVE) subcommand of ACF. You can abbreviate this subcommand to a minimum of SET C(ACFS). ACFSERVE privilege records can override the default ACFSERVE authorizations and loggings CA ACF2 for VM supplies.

ACFSERVE privilege records are optional and you do not have to define them. You need these records only if you override the CA ACF2 for VM default authorizations. You can make changes to the ACFSERVE privilege records active dynamically with an ACFSERVE command.

Example of an ACFSERVE Privilege Record

The following is an example of an ACFSERVE privilege record:

```
ABC1 / RESET LAST CHANGED BY ADMISO ON 07/01/92-14:52
       AUTH(ACCOUNT SECURITY) AUTLOGIC(OR) LOG SCOPED
```

Each ACFSERVE privilege record consists of three components: the SYSID, the record name, and the record fields. In the above example,

ABC1

Represents the SYSID associated with the ACFSERVE privilege record.

RESET

Represents the record name. This record defines the privileges for the ACFSERVE RESET command.

The data on the second line represents the actual record fields and their contents.

AUTH

Specifies the authorization (logonid privilege) necessary to issue the specified command. Our example specifies ACCOUNT SECURITY.

AUTLOGIC

Specifies that a logonid with the ACCOUNT or SECURITY privilege can issue the ACFSERVE RESET command.

SCOPED

Indicates that a scoped security administrator can also issue the command.

LOG

Indicates that CA ACF2 for VM writes an SMF record to log the command.

All ACFSERVE privilege records contain the same record fields. You must define separate records for each ACFSERVE command you need to override.

ACFSERVE Privilege Record Fields

All ACFSERVE privilege records have the same field names. These fields are defined below.

AUTH(privilege list)

Specifies the CA ACF2 for VM or site-defined privilege field that must be in the logonid of users who can issue the specified ACFSERVE command. For example, you can specify a privilege list of AUTH(SECURITY,ACCOUNT) in the RESET record. This means that the user must have the SECURITY or ACCOUNT privilege in his logonid record to issue the ACFSERVE RESET command (assuming AUTLOGIC is defined as OR). If you specify AUTH without any values, any user can issue the indicated ACFSERVE command.

AUTLOGIC(AND|OR)

Specifies the type of logical test applied between the AUTH field's privilege list and the user's logonid. AND means that all of the AUTH field's privilege list entries must be in the logonid record of users who can issue the specified ACFSERVE command. OR means that any one of the AUTH field's privilege list entries must be in the logonid record of users who can issue the specified ACFSERVE command.

SCOPED|NOSCOPED

Specifies whether a logonid with AUTH(SECURITY) can be scoped or not. This field only applies to the CA ACF2 for VM SECURITY privilege. SCOPED means that a scoped security administrator can issue the specified ACFSERVE command. NOSCOPED means that only an unscoped security administrator can issue the specified ACFSERVE command.

LOG|NOLOG

Specifies whether CA ACF2 for VM always creates an SMF record for the specified ACFSERVE command. In the case of an authority violation, CA ACF2 for VM always writes an SMF record.

ACFSERVE Privilege Record Names

CA selects the record IDs for ACFSERVE privilege records. You can define or modify these record IDs. These predefined record IDs are listed below with the ACFSERVE commands they control.

CA ACF2 for VM-supplied defaults for each record are shown below. Command synonyms (for those that have them) are provided following this list.

Record ID	AUTH	AUTO- LOGIC	Scoped?	Log?
ARCHIVE.SMF	ACCOUNT, SECURITY	OR	SCOPED	
BACKUP	SECURITY	AND	SCOPED	
CKPT.DATABASE	AUDIT, SECURITY	OR	SCOPED	
CKPT.SMF	AUDIT, SECURITY	OR	SCOPED	
DELETE.RESOURCE	ACCOUNT, SECURITY	AND	SCOPED	
DISABLE.SYNC	ACCOUNT, SECURITY	AND	NOSCOPED	
ENABLE.SYNC	ACCOUNT, SECURITY	AND	NOSCOPED	
QUERY.BATCH	NONE	AND	SCOPED	

Record ID	AUTH	AUTO- LOGIC	Scoped?	Log?
QUERY.DATABASE	AUDIT, SECURITY	OR	SCOPED	
QUERY.DDSN	AUDIT, SECURITY	OR	SCOPED	
QUERY.GROUPID	NONE	AND	SCOPED	
QUERY.SECTRACE	AUDIT, SECURITY	OR	SCOPED	
QUERY.SMF	AUDIT, SECURITY	OR	SCOPED	
QUERY.SOURCE	NONE	AND	SCOPED	
QUERY.STATUS	NONE	AND	SCOPED	
QUERY.SYNC	AUDIT, SECURITY	OR	SCOPED	
RELOAD.COMMAND	SECURITY	AND	SCOPED	

Record ID	AUTH	AUTO- LOGIC	Scoped?	Log?
RELOAD.CONTROL.ACFSERVE	SECURITY	AND	NOSCOPED	
RELOAD.CONTROL.VMO	SECURITY	AND	NOSCOPED	
RELOAD.DIAGNOSE	SECURITY	AND	SCOPED	
RELOAD.FDR	ACCOUNT, SECURITY	AND	NOSCOPED	
RELOAD.LMP	ACCOUNT, SECURITY	AND	NOSCOPED	
RELOAD.PROFILE	SECURITY	AND	SCOPED	
RELOAD.RESOURCE	ACCOUNT, SECURITY	AND	SCOPED	
RELOAD.RULE	SECURITY	AND	SCOPED	
RELOAD.SHIFT	SECURITY	AND	SCOPED	
RELOAD.XREF	SECURITY	AND	SCOPED	
RESET	ACCOUNT, SECURITY	OR	SCOPED	

Record ID	AUTH	AUTO- LOGIC	Scoped?	Log?
RESTART	SECURITY	AND	NOSCOPED	
SECTRACE.DELETE SECTRACE.DISABLE SECTRACE.ENABLE SECTRACE.MODIFY SECTRACE.SET	ACCOUNT, SECURITY	OR	SCOPE	
SET.SYSID	SECURITY	AND	NOSCOPED	
SWITCH.SMF	AUDIT, SECURITY	OR	SCOPED	

ACFSERVE Command Synonyms

The information below provides synonyms for some ACFSERVE commands.

Command Keyword	Command Synonym
ACFSERVE QUERY	ACFSERVE DISPLAY
ACFSERVE RELOAD	ACFSERVE LOAD
ACFSERVE RELOAD	ACFSERVE REFRESH
ACFSERVE RELOAD SHIFT	ACFSERVE NEWSHIFT
ACFSERVE RELOAD XREF	ACFSERVE NEWXREF

Additional Considerations:

- The QUERY.SMF privilege record controls the QUERY SMF and DISPLAY SMF commands.
- CA ACF2 for VM does not create separate privilege records for command synonyms.

SYSID Concepts

The SYSID and MSYSID concepts described for VMO records also apply to the ACFSERVE privilege records. The only difference is that references to CONTROL(VMO) are understood to be CONTROL(ACFSERVE).

Maintaining ACFSERVE Options

After you establish the CONTROL(ACFSERVE) setting, you can insert, change, list, and delete ACFSERVE privilege records. After you process ACFSERVE privilege records, use the ACFSERVE RELOAD command to apply the affected records to the system. Use the SHOW subcommands to verify the values in effect. You do not have to IPL or restart the CA ACF2 for VM service machine.

ACF Subcommands under the CONTROL Setting

You can process ACFSERVE privilege records after you establish the CONTROL(ACFSERVE) setting of the ACF command and specify the type code ACFSERVE through the SET CONTROL(ACFSERVE) command.

After you establish the CONTROL(ACFSERVE) setting, you can issue any of the following subcommands:

- CHANGE
- EXEC
- SET
- CMS
- HELP
- SHOW
- CP
- INSERT
- XEDIT
- DELETE
- LIST
- END
- QUIT

The END, QUIT, and HELP subcommands operate under the CONTROL(ACFSERVE) setting as described earlier in this guide. The other subcommands are described in the following text. SET and SHOW are common subcommands, but we describe them in this section because they function differently under the CONTROL(ACFSERVE) setting than under other ACF command settings.

Setting the CONTROL(ACFSERVE) Mode

Use the SET subcommand to establish the CONTROL(ACFSERVE) setting for the ACF command. It also establishes the active SYSID and other parameters that affect how certain ACF subcommands operate. The syntax for the SET subcommand under the CONTROL setting is:

```
Set Control(ACFSERVE) [ SYSID(sysid)      ]
                      [ MSYSID(sysidmask) ]
```

The parameters for the SET CONTROL(ACFSERVE) subcommand are defined below.

SYSID(sysid)

Specifies the SYSID of the ACFSERVE privilege records that you want to process under this setting. This SYSID remains active for the duration of the CONTROL(ACFSERVE) setting, but you can override it with the SYSID parameter of the individual ACF subcommands. If you do not specify the SYSID parameter when you establish the CONTROL(ACFSERVE) setting, CA ACF2 for VM uses the system default. You can abbreviate SYSID to a minimum of SYS.

MSYSID(sysidmask)

Specifies a mask to indicate the SYSIDs of the ACFSERVE privilege records that you want to process. These SYSIDs remain active for the duration of the CONTROL(ACFSERVE) setting, but you can override them with the SYSID parameter of the individual ACF subcommands. If you do not specify the SYSID parameter when you establish the CONTROL(ACFSERVE) setting, CA ACF2 for VM uses the system default. You can abbreviate MSYSID to a minimum of MSYS.

Displaying ACFSERVE Privilege Records

The SHOW subcommand, under the CONTROL(ACFSERVE) setting, displays the currently active SYSID and the fields of the ACFSERVE privilege records. As it does under other system settings, the SHOW subcommand under the CONTROL(ACFSERVE) setting, also displays many of the system options in effect. (For more information on the ACF SHOW subcommands, see the “Using the ACF Command” chapter in this guide.) The syntax of the SHOW subcommand for ACFSERVE privilege records is:

```
     { ACFSERVE            }
SHow { FIELDS[recordname]  }
     { MODE                }
```

You can enter only one parameter at a time with the SHOW subcommand. All parameters (except FIELDS and MODE) operate under the CONTROL(ACFSERVE) setting as described in the SHOW subcommand description in Chapter 3 of this guide. The FIELDS and MODE parameters are described below.

FIELDS[recordname]

Specifies that the SHOW subcommand displays the names of the fields in the specified ACFSERVE privilege record. The SHOW subcommand display also indicates which fields you can change.

MODE

Specifies that the SHOW subcommand displays the current setting of the ACF command, the ACFSERVE type code, and the currently active SYSID:

```
acf
SET CONTROL(ACFSERVE)
SHow MODE

MODE: CONTROL TYPE: SER SYSID: CPU1
```

Creating ACFSERVE Privilege Records

The INSERT subcommand, under the CONTROL(ACFSERVE) setting, creates an ACFSERVE privilege record to define required ACFSERVE command privileges. The syntax for the INSERT subcommand is:

```
       { *                          }
INsert { recordname                 }
       { USING(oldrecord newrecord) }

       [ SYSid(sysid)               ]
       [ USYSID(sysid)              ]
       [ field1,...,fieldn          ]
       [ ADD|REP|DEL                ]
```

Issue the INSERT subcommand with the SYSID, record ID, and fields defined in the record:

```
ACF
SET CONTROL(ACFSERVE)
INSERT RESET SYSID(CPU1) AUTH(SECURITY) SCOPED LOG
```

This sample subcommand inserts the ACFSERVE RESET record. When CA ACF2 for VM is running under the SYSID CPU1, users must have the SECURITY privilege in their logonid records to issue this command. CA ACF2 for VM logs the command in an SMF record. If you do not specify the SYSID parameter, any options are defined for the currently active SYSID.

The variable recordname specifies the name of the ACFSERVE privilege record you want to insert. You cannot mask this field. See ACFSERVE Privilege Record Names for a list of acceptable record names.

See Creating VMO Records for descriptions of the other parameters of the INSERT subcommand.

Changing ACFSERVE Privilege Records

The CHANGE subcommand, under the CONTROL(ACFSERVE) setting, changes existing ACFSERVE privilege records. The syntax for the CHANGE subcommand is:

```
       { *               }
CHange { recordname      } field1,...,fieldn
       { LIKE(recordmask) } [ADD|DEL|REP]

       [ SYSID(sysid)      ]
       [ MSYSID(sysidmask) ]
```

The following CHANGE subcommand modifies the RESET record so that users who want to issue the ACFSERVE RESET command must have SECURITY, ACCOUNT, or AUDIT in their logonid records:

```
ACF
SET CONTROL(ACFSERVE)
CHANGE RESET AUTH(SECURITY,ACCOUNT,AUDIT) AUTLOGIC(OR)
```

Because we did not specify a SYSID, the subcommand affects the RESET record for the currently active SYSID.

See Changing VMO Records section for descriptions of the parameters for the CHANGE subcommand.

Listing ACFSERVE Privilege Records

The LIST subcommand, under the CONTROL(ACFSERVE) setting, lists the contents of the specified ACFSERVE privilege record. The syntax for the LIST subcommand is:

```
     { *                 }
List { recordname        }
     { LIKE(recordmask)  }

     [ SYSID(sysid)      ]
     [ MSYSID(sysidmask) ]
```

See the Listing VMO Records section for descriptions of the parameters for the LIST subcommand.

Deleting ACFSERVE Privilege Records

The DELETE subcommand, under the CONTROL(ACFSERVE) setting, deletes specified ACFSERVE privilege records. The syntax for the DELETE subcommand is:

```
       { *                }
DELete { recordname       }
       { LIKE(recordmask) }

       [ SYSID(sysid)     ]
       [ MSYSID(sysidmask) ]
```

See the Deleting VMO Records section for complete descriptions of the parameters for the DELETE subcommand.

Chapter 23: Maintaining Profile Records

CA ACF2 for VM Security Databases can contain security information used by other system applications. Profile records contain this information.

This section contains the following topics:

Profile and Segment Information

The information that can be retained in the security database is associated with a user or a VM resource. This general classification is commonly referred to as a *profile*. (For example, a USER profile, a GROUP profile, or other resource profile.) The group of data that is to be extracted is commonly referred to as *segment data*. For example, a SAF call can be issued to extract the OEVM segment of the USER profile.

The following profile and segment information is supported by CA ACF2 for VM:

Profile Record	Segment
GROUP	LINUX, OEVM, OMVS
PTKTDATA	SSIGNON
USER	LINUX, OEVM, OMVS, PASSWORD

Note: The USER profile records are documented in the "Maintaining Logonid Records" Chapter.

CA ACF2 for VM keeps this information in the infostorage database. The records are administered through the CA ACF2 for VM ACF command.

GROUP Profile Records

The GROUP profile records contain information needed by OpenExtensions VM and LINUX/390 applications (CA PAM for Linux for System z) to verify user access. The profile data information segments that can be extracted for a GROUP include OEVM (for OpenExtensions VM) and LINUX (for CA PAM for Linux for System z support). For compatibility, OMVS (for CA PAM for Linux for System z support prior to r8.0) may still be used if linuxdata OMVS is included in the CA PAM for Linux for System z configuration file.

The key of a GROUP profile record is a group name, which cannot be masked.

OEVM Group Profile Record

Record ID	Fields
Recid	GID(*gid*)

recid

Specifies the group name. This value cannot be masked.

GID(*gid*)

A numeric field that accepts values from zero to 2,147,483,647.

The following example shows how to create a GROUP Profile record for group OEVMGRP and assign it a GID value of 20.

```
SET PROFILE(GROUP) DIVISION(OEVM)
INSERT OEVMGRP GID(20)
```

LINUX Group Profile Record

Record ID	Fields
Recid	LINUXGID(*gid*)

recid

Specifies the group name. This value cannot be masked.

LINUXGID(*gid*)

A numeric field that accepts values from zero to 2,147,483,647.

The following example shows how to create a GROUP Profile record for group LINUXGRP and assign it a LINUXGID value of 20.

```
SET PROFILE(GROUP) DIVISION(LINUX)

INSERT LINUXGRP LINUXGID(20)
```

PTKTDATA Profile Records

CA ACF2 for VM PTKTDATA Profile records provide support for the Secured Signon function of the IBM Network Security Program. The Secured Signon feature lets network users sign on to a VM host using a dynamically generated password substitute called a PassTicket. A cryptographic algorithm that uses user ID and profile information, a secret key, and a time value to generate PassTickets. They have a short lifespan and are resistant to password interception and replay attacks. You can generate PassTickets for access to VM, OS/390, TSO, CICS, IMS, or APPC/MVS.

SSIGNON Profile Data Records

To support Qualified PassTicket signon, CA ACF2 for VM finds security key values in Profile records on its Infostorage database. The new SSIGNON Profile Data record of the PTKTDATA profile has a field named SSKEY. It contains the security key used for decrypting the PassTicket and validating the signon. The record ID is usually the VTAM application ID of the application the user is trying to access.

Record ID	Fields
recid	SSKEY(key-value)

recid

A 1- to 26-character profile name that identifies the application where the key-value applies. The record ID is determined by four distinct combinations of an application name, a group name, and a user ID. The four combinations (in PassTicket validation) are:

```
application.group.userid
application.userid
application.group
application
```

For VM systems, the application name is the characters “VM” suffixed with the SMF system ID.

For CICS, IMS, or APPC/MVS applications, this is the VTAM application ID.

For TSO, it is the characters “TSO” suffixed with the SMF system ID as defined in the SMFPRMxx member of SYS1.PARMLIB.

For OS/390 batch jobs, it is the characters “MVS” suffixed with the SMF system ID. In both cases, &acf. ignores any special characters in the SMF system ID (for example, “SY*6” becomes “SY6”).

The group name for VM systems is the GROUP field in the Logonid record.

SSKEY

A 16-character hexadecimal representation of the eight-byte encryption key for this application.

Following is an example of the commands to issue to create a profile record letting a user signon to a host application VM on a system with an SMF ID of XE75) from a workstation:

```
SET PROFILE(PTKTDATA) DIVISION(SSIGNON)

INSERT VMXE75 SSKEY(c237d18425cfe12d)
```

Following is an example of the commands to issue to create a Profile record letting user JOE in group SYSPROGS signon to a host application VM system with an SMF ID of ESA3:

```
INSERT VMESA3.SYSPROGS.JOE SSKEY(C31D4EF1908DF1AF)
```

Following is an example of the commands to issue to create a Profile record letting users in group ADMIN signon to a host application VM system with an SMF ID of PROD:

```
INSERT VMPROD.ADMIN SSKEY(C537D04C03C77F1E)
```

Using the ACF Command

Once within the ACF command, use the SET command to enter profile administration. Enter:

`SET PROFILE(`*`profile-name`*`) DIVISION(`*`profile-data-name`*`)`

Where *profile-name* can be one of the following: USER, GROUP, or PTKTDATA. DIVISION is the profile data name of the data record.

The following commands can be used with profile records:

INSERT *|*recid*|USING(*urecid*) *recid*

LIST *|*recid*|LIKE(*recidmask*)

DELETE *|*recid*|LIKE(*recidmask*)

CHANGE *recid*|LIKE(*urecid*)

The infostorage class for profile records is PROFILE; the letter representing the class is P. CA ACF2 for VM distributes the following CLASMAP resource type definitions with the product; adjustments might be needed if your site uses different types for these resources.

Profile record	Resource Type
GROUP	GRP
PTKTDATA	PTK
USER	USR

Chapter 24: Using the ACFSERVE Commands

With the ACFSERVE commands, authorized users can look at and move files and other data maintained through the CA ACF2 for VM service machine. Only users who have the proper CA ACF2 for VM security privileges can use the ACFSERVE commands. These CA ACF2 for VM privileges are AUDIT, ACCOUNT, SECURITY, or a combination of these privileges (depending on the command).

We also provide the ACFSERVE Command Tracking Log (ACFRPTCT) that generates SMF records for all valid ACFSERVE commands you enter. With this logging capability, you can monitor all attempts to communicate directly with the CA ACF2 for VM service machine. For more information on the ACFRPTCT report, see the *Reports and Utilities Guide*.

When you finish this chapter, you will know how to use the ACFSERVE commands to:

- Archive, display, and switch SMF files
- Back up the CA ACF2 for VM databases
- Force checkpoints
- Remove resource types
- Reload tables, command and diagnose limiting rule sets and models, infostorage records, the ACFFDR, VMO records, ACFSERVE privilege records, resource types, cached access rule sets, and shift information
- Display information on batch machines, database use, group user IDs, source, and infostorage records
- Lower password violation counts
- Restart the CA ACF2 for VM service machine
- Revalidate your logon password.

This section contains the following topics:

ACFSERVE Command Syntax

The ACFSERVE command is easy to use. An explanation of the command syntax follows.

- The verb operand (action) of the command is used first
- The next operand is the object of the action
- You can abbreviate the ACFSERVE command to a minimum of ACFS
- You can abbreviate some operands.

You must enter at least one operand to use the ACFSERVE command. All the ACFSERVE commands and syntax are listed below with, if applicable, the
CA ACF2 for VM privilege you must have in your logonid record to issue the command.

Command	Syntax	CA ACF2 for VM Privilege Required
ACFServe	ARchive SMF	ACCOUNT or SECURITY
	BAckup [ddsn] [LID|RULE|INFO]	SECURITY
	CKPT DAtabase [CLEAR]	AUDIT or SECURITY
	CKPT SMF	AUDIT or SECURITY
	DELete RESource	ACCOUNT and SECURITY
	DISAble NOAUTO UPdates	SECURITY
	DISAble SYnc	ACCOUNT and SECURITY
	ENable NOAUTO UPdates	SECURITY
	ENable SYnc	ACCOUNT and SECURITY
	Query Batch {id|*|ALL}	None
	Query DAtabase	AUDIT or SECURITY
	Query DDSN	AUDIT or SECURITY
	Query Groupid {id|*|ALL}	None
	Query SECtrace	AUDIT or SECURITY
	Query SMf	AUDIT or SECURITY
	Query Source {id|*|ALL}	None
	Query STatus [VM]	None

Command	Syntax	CA ACF2 for VM Privilege Required
	Query SYnc [CLEAR]	AUDIT or SECURITY
	Reload COMmand	SECURITY
	Reload Control ACFServe {ALL|recid} [SYSID sysid]	SECURITY
	Reload Control Vmo {ALL|recid} [SYSID sysid FORCE]	SECURITY
	Reload DIAGnose	SECURITY
	Reload LMP	ACCOUNT and SECURITY
	Reload FDR	ACCOUNT and SECURITY
	Reload PROfile {OEVM|LINUX}[ALL|USER|GROUP]	SECURITY
	Reload RESource	ACCOUNT and SECURITY
	Reload RULe	SECURITY
	Reload SHift	SECURITY
	Reload XRef {SGP|RGP}	SECURITY
	RESET logonid	ACCOUNT or SECURITY
	RESTART [dump]	SECURITY
	SECtrace	ACCOUNT or SECURITY
	SET SYSID (sysid) [PERM]	SECURITY
	SWitch SMF	AUDIT or SECURITY
	VALidate PASsword	None

The ACFSERVE commands and operands are described throughout this chapter. We also indicate what CA ACF2 for VM privileges you need to use these commands with each command syntax.

Archiving SMF Files

The ACFSERVE ARCHIVE SMF command tells the CA ACF2 for VM ACF2 service machine that an SMF file in unload status has been processed and can be flagged as a history disk. When you specify SWITCH=NOTIFY in the @SMF macro, a different virtual machine unloads the SMF files or processes the data in some other way. The ACFSERVE ARCHIVE SMF command notifies the CA ACF2 for VM service machine that the SMF data is unloaded and you can reuse the file. You must have the ACCOUNT or SECURITY privilege to issue this command. The syntax for the ACFSERVE ARCHIVE SMF command is:

```
ACFServe ARchive SMF [ft]
```

The variables for this command are defined below.

ft

Specifies the filetype (in the format yydddsss), where

yy

Specifies the last two digits of the year the SMF file was created.

ddd

Specifies the Julian date when the SMF file was created.

sss

Specifies the sequence number of the file. The first file created each day has a sequence number of 001.

Backing up the CA ACF2 for VM Databases

The ACFSERVE BACKUP command creates backup copies of one or all three CA ACF2 for VM databases. By default, only users with the SECURITY privilege can issue this command. However, you can modify the AUTH operand of the BACKUP VMO record to change this authority level. The syntax for this command is:

```
                [LID  ]
ACFServe BAckup [ddsn ] [RULE ]
                [INFO ]
```

The operands for this command are defined below.

ddsn

Specifies the name of an ACFFDR @DDSN macro where information about the backup files is located (filenames and minidisk addresses). This operand is optional. By default, the group named in the DDSNID field of the BACKUP VMO record.

LID|RULE|INFO

These operands are optional and specify that only one database should be copied.

LID

Indicates the Logonid database

RULE

Indicates the Rule database

INFO

Indicates the Infostorage database

All three databases are processed through the ACFSERVE BACKUP command by default.

See the “Backup and Recovery” chapter for more information about database backup procedures.

Forcing a Database Checkpoint

The ACFSERVE CKPT command forces a database checkpoint to occur, checking current statistics for any CMS database. You must have the AUDIT or SECURITY privilege to issue this command. The syntax for the ACFSERVE CKPT command is:

```
ACFServe CKPT DAtabase [CLEAR]
```

The operands for this command are defined below.

DATABASE

Specifies the current statistics for any CMS database being used are written out to DASD.

[CLEAR]

Specifies that when you issue an ACFSERVE QUERY DATABASE command the current statistics displayed are cleared.

Forcing an SMF Checkpoint

The ACFSERVE CKPT SMF command forces an SMF checkpoint. It is used when you specify PRIORITY=NO in the @SMF macro. This command runs reports on SMF data up to the point you issue the command. You must have the AUDIT or SECURITY privilege to issue this command. The syntax for the ACFSERVE CKPT SMF command is:

```
ACFServe CKPT SMF
```

Removing a Resource Type

The ACFSERVE DELETE RESOURCE command removes a resource type from the resident type list, even if it was made resident through the RESTYPE VMO record or through the ACFSERVE LOAD RESOURCE command. This command deletes a type list to make room for another resident type list. You must have the ACCOUNT or SECURITY privilege to issue this command. The syntax for the ACFSERVE DELETE RESOURCE command is:

```
ACFServe DELete RESource typ
```

The operand for this command is defined below.

typ

Deletes resources. You can specify typ (three alphanumeric characters) to delete one record type or an asterisk (*) to delete all resource type codes from the resident type list. If you delete a type code and want to rebuild it later, use the ACFSERVE RELOAD RESOURCE command.

Terminating Updates in NOAUTO Mode

The ACFSERVE DISABLE NOAUTO UPDATES command stops the CA ACF2 for VM service machine to terminate updates to the databases in NOAUTO mode. The syntax for this command is:

```
ACFServe DISAble NOAUTO UPdates
```

The ACFSERVE DISABLE NOAUTO UPDATES command:

- Only a noauto update user can issue if the system was initialized in NOAUTO mode
- Is rejected for any user ID that is not assigned as the noauto update user unless the noauto update user ID is not logged on
- Calls the CP service machine logoff routine
- Clears the NOAUTO UPDATE status
- Issues a message when it has completed.

If the CA ACF2 for VM service machine crashes or is forced off, the noauto update user does not change. Only the noauto update user can reactivate the service machine. If the noauto update user logs off without issuing the ACFSERVE DISABLE NOAUTO UPDATES command, another user in the NOAUTOU list can issue the ACFSERVE DISABLE NOAUTO UPDATES command on his behalf. (For an explanation of the NOAUTOU list, turn to Updating the Databases in NOAUTO Mode.) The ACFSERVE ENABLE NOAUTO UPDATES command is explained in Updating the Databases in NOAUTO Mode.

Displaying Information

The ACFSERVE QUERY command displays system information. See the ACFSERVE QUERY commands for more information.

Stopping Database Synchronization

The ACFSERVE DISABLE SYNC command stops the Database Synchronization Component from sending or processing synchronization requests. The syntax for this command is:

```
ACFServe DISAble SYnc
```

Updating the Databases in NOAUTO Mode

The ACFSERVE ENABLE NOAUTO UPDATES command starts the CA ACF2 for VM service machine when the system is in NOAUTO mode. Your user ID must be defined in the NOAUTOU list to issue this ACFSERVE command. The syntax for this command is:

```
ACFServe ENable NOAUTO UPdates
```

The ACFSERVE ENABLE NOAUTO UPDATES command:

- Checks the NOAUTOU list for users who can issue the ACFSERVE ENABLE NOAUTO UPDATES command. Define users to the NOAUTOU list in the SRVMOPTS and VMXAOPTS macros. This list specifies the names of VM users who can log on and update the databases when CA ACF2 for VM is inactive. You must define users in the NOAUTOU list and in the FORCEID list to allow them to log on in NOAUTO mode. If the user is not included in this list, CA ACF2 for VM rejects the command.
- Assigns the user that issued the ACFSERVE ENABLE NOAUTO UPDATES command as the noauto update user.
- Calls the CP routine that starts the CA ACF2 for VM service machine.

If the service machine startup fails, the NOAUTO UPDATE status clears. In any case, CA ACF2 for VM issues a message to the user and the command completes. The noauto update user can also use the ACF command and the ACFCOMP, ACFDCMP, and ACFNRULE commands to update rules and information on the CA ACF2 for VM databases.

The noauto update user has all CA ACF2 for VM security privileges and CA ACF2 for VM creates SMF records for all updates he performs. For more information about NOAUTO mode, see the *Systems Programmer Guide.*

During NOAUTO update mode, database backups defined in the TIME field of the BACKUP VMO record are disabled. Enter the SHOW BACKUP subcommand to display the backup times that are effective for a normal IPL. See the “Defining Structured Infostorage Records” chapter for more information on the BACKUP VMO record.

Allowing Database Synchronization

The ACFSERVE ENABLE SYNC command allows the Database Synchronization Component to process or send synchronization requests. The syntax of this command is:

```
ACFServe ENable SYnc
```

Reloading Information

To reload information use ACFSERVE RELOAD. See the ACFSERVE RELOAD commands for more information.

Reloading Resident Matrix Tables

The ACFSERVE NEWSHIFT command is a synonym for ACFSERVE RELOAD SHIFT. See the Reloading Shift Information section for information and syntax for this command. We will not support this ACFSERVE command in future releases.

Reloading the Input Source Cross-Reference Table

The ACFSERVE NEWXREF command is a synonym for the ACFSERVE RELOAD XREF command. See the Reloading Shift Information section for information and syntax for the ACFSERVE RELOAD XREF command. We will not support the ACFSERVE NEWXREF command in future releases.

Displaying Batch Machine User IDs

The ACFSERVE QUERY BATCH command displays information about batch machines logged onto CA ACF2 for VM. You do not need any special privileges to issue this command. However, you can be restricted through command limiting. The syntax for this command is:

```
                    { id  }
ACFServe Query Batch { *   }
                    { ALL  }
```

The operands for this command are defined below.

id

Indicates the user ID of a specific batch machine.

*****

Indicates the user ID of this virtual machine.

ALL

Indicates all logged on users that are batch IDs. This is the default.

If the ID specified is not a batch machine or has an active job, CA ACF2 for VM issues the following message:

```
ACFpgm44E user ID HAS NO ACTIVE BATCH USER
```

Displaying Database Information

The ACFSERVE QUERY DATABASE command displays information about database usage. You must have the AUDIT or SECURITY privilege in your logonid record to issue this command. The syntax for this command is:

```
ACFServe Query DAtabase
```

Displaying Information about the DDSN at System Startup

The ACFSERVE QUERY DDSN command displays information about the DDSN parameter specified at system startup. The CA ACF2 for VM service machine profile analyzes the command response to link and access the proper databases. You must have the AUDIT or SECURITY privilege in your logonid record to issue this command. The syntax for this command is:

```
ACFServe Query DDSN
```

Displaying Logged on Group IDs

The ACFSERVE QUERY GROUPID command displays information about group IDs that are logged onto CA ACF2 for VM. You do not need any special privileges to issue this command. However, you can be restricted through command limiting. The syntax for this command is:

```
                      { id  }
ACFServe Query Groupid { *   }
                      { ALL }
```

The operands for this command are defined below.

id

Indicates the user ID of a group virtual machine.

*****

Indicates the user ID of this virtual machine.

ALL

Indicates all logged on users that are group IDs. This is the default.

Displaying SECTRACE Requests

The ACFSERVE QUERY SECTRACE command displays the current status of all SECTRACE requests. You must have the AUDIT or SECURITY privilege in your logonid record to issue this command. The syntax for this command is:

```
ACFServe Query SECtrace
```

The following is sample output of the ACFSERVE QUERY SECTRACE request:

```
                SECTRACE STATUS DISPLAY                    11/13/00 00:13:15
TRACEID    STATUS    REQUEST   USER    SERVMACH   DESTID   LIMIT   COUNT
FTPSERVE   ENABLED   ALL       ALL     FTPSERVE   MAINT    100     42561
```

The ACFSERVE QUERY SECTRACE subcommand displays the following information for each SECTRACE request:

TRACEID

Indicates the name of the SECTRACE request.

STATUS

Indicates whether the SECTRACE request is enabled or disabled.

REQUEST

Indicates the RACROUTE REQUEST= to trace.

USER

Indicates the user to trace.

SERVMACH

Indicates the service machine to trace.

DESTID

Indicates the destination user that the trace output is sent to.

LIMIT

Indicates the maximum number of trace events to produce from this SECTRACE request.

COUNT

Indicates the count of SAF events that match this SECTRACE REQUEST. If the number is higher than the limit, no tracing occurs; however, the count is still incremented for each event.

Displaying SMF Files

The ACFSERVE QUERY SMF command shows all SMF files and their current status. You must have the AUDIT or SECURITY privilege in your logonid record to issue this command. The syntax for this command is:

```
ACFServe Query SMf
```

The response to the ACFSERVE QUERY SMF subcommand may show one of the following as status for each SMF disk:

ACTIVE

Indicates the SMF disk that is currently being used to store new SMF records.

HISTORY

Indicates that this is an SMF disk that has SMF records from the last time it was the ACTIVE disk. These records are known as historical SMF records.

READY

Indicates that this is normally the SMF disk that becomes the new ACTIVE disk when the next switch of SMF disks occurs. This disk also has historical SMF data if it was an ACTIVE disk.

UNLOAD

Occurs immediately after a SWITCH SMF. The ACTIVE disk is assigned an UNLOAD status until an SMF dump for this disk completes or when a service machine (site-defined to save SMF data from SMF disks in UNLOAD status) issues the ACFSERVE ARCHIVE SMF command. This disk is then assigned HISTORY or READY status.

The status of SMF disks normally progresses from READY to ACTIVE to UNLOAD to HISTORY and then becomes a READY disk to restart the cycle.

Displaying Logical Source Information

The ACFSERVE QUERY SOURCE command displays the real device and logical source that a user is logged onto. You do not need any special privileges to issue this command. The syntax for this command is:

```
                     { id  }
ACFServe Query Source { *   }
                     { ALL }
```

The operands for this command are defined below.

id

Display an individual user ID.

*****

Display the user ID of the person issuing the command.

ALL

Display all users that are logged onto the system.

Displaying Status of Database Synchronization Options and Counters

The ACFSERVE QUERY SYNC command displays the status of the Database Synchronization Component, selected options, queued request counters, and activity counters. The syntax of this command is:

```
ACFServe Query SYnc [CLEAR]
```

The optional CLEAR keyword clears the incoming, feedback, and outgoing counters to 0 (zero) after the current counters are displayed. The CLEAR operand does not affect the queued request counters.

For more help in understanding the various output to the command, see messages 3AA through 3AE in the *Messages Guide.*

Displaying CA ACF2 for VM Status

The ACFSERVE QUERY STATUS command validates that a compatible release of CA ACF2 for VM is installed on CP. CA ACF2 for VM utilities, reports, and commands use this validation. It displays information about the status of CA ACF2 for VM, including the release number, status, and release level. The release level contains the genlevel for this product and release. You do not need any special privileges to issue this command. However, you can be prevented through command limiting. Prevented users cannot use CA ACF2 for VM utilities, reports, or the ACF command. The syntax for this command is:

```
ACFServe Query STatus [VM]
```

The optional VM keyword requests that this ACFSERVE command check to ensure that CA ACF2 for VM is in use. The VM keyword results in a return code 4 if this is a VSE-only system.

Reloading CA ACF2 for VM Options

The ACFSERVE RELOAD CONTROL command reloads infostorage records to reflect new settings. This ACFSERVE command loads VMO records and ACFSERVE privilege records. You must have the SECURITY privilege in your logonid record to issue either of these commands. The FORCE parameter does not apply to ACFSERVE privilege records because these records are optional. CA ACF2 for VM assumes you want to use default records, but does not add them to the Infostorage database. Do not specify FORCE if you are reloading ACFSERVE privilege records. The syntax for this command is:

```
ACFServe Reload Control{ Vmo      }{ALL|recid} [SYSID *|sysid ][FORCE]
                       { ACFServe }
```

Control VMO|ACFSERVE

Reloads either VM options from VMO records or ACFSERVE privilege records.

ALL|recid

Specifies the VMO records with the CA ACF2 for VM. options to refresh. If you specify ALL, CA ACF2 for VM processes all VMO records and their reloadable VM options. If you specify recid, CA ACF2 for VM only processes those VM options that recid set. If you can qualify the recid, then CA ACF2 for VM processes all qualified records.

SYSID*|sysid

Specifies the SYSID that is used during this reload. Only those VMO records with a SYSID or masked SYSID that match the specified SYSID are processed. If you enter an asterisk (*) for the SYSID, or if you do not enter a SYSID at all, CA ACF2 for VM uses the currently active SYSID.

FORCE

Affects how CA ACF2 for VM handles a NO RECORD FOUND condition. If you do not specify FORCE, a NO RECORD FOUND condition becomes an error, CA ACF2 for VM issues a message, and does not change any system options. If you do specify FORCE, a NO RECORD FOUND condition causes CA ACF2 for VM to build and use a default record. CA ACF2 for VM adds this record to the Infostorage database. Do not specify this parameter if you are reloading ACFSERVE privilege records.

The ACFSERVE REFRESH command is a synonym for the ACFSERVE RELOAD command.

Reloading Command Limiting Rule Sets and Models

The ACFSERVE RELOAD command reloads the specified command limiting rule set and model into the command limiting rule cache. When the rule set is required for validation, CA ACF2 for VM reloads the most current version of the rule set into the cache. If CA ACF2 for VM databases are shared, issue this command on all systems running CA ACF2 for VM after you update a command limiting rule set. You must have the SECURITY privilege in your logonid record to issue this command. The ruleid must also be in your scope. The syntax for this command is:

```
ACFServe Reload COMmand ruleid
```

ruleid

Purge this rule set, identified by its ruleid, as specified by the $KEY control statement.

Reloading Diagnose Limiting Rule Sets

The ACFSERVE RELOAD DIAGNOSE command reloads the specified diagnose limiting rule set to the rule cache. When the rule set is required for validation, CA ACF2 for VM reloads the most current version of the rule set into the cache. If you are sharing CA ACF2 for VM databases, issue this command on all systems running CA ACF2 for VM after a diagnose limiting rule set is updated. You must have the SECURITY privilege in your logonid record to issue this command. The ruleid must also be in your scope. The syntax for this command is:

```
ACFServe Reload DIAGnose ruleid
```

ruleid

Purge this rule set, identified by its ruleid, as specified by the $KEY control statement.

Reloading the ACFFDR

The ACFSERVE RELOAD FDR command reloads the ACFFDR and lets you dynamically change many CA ACF2 for VM system options without having to shutdown and reIPL the CA ACF2 for VM service machine. This includes all CA ACF2 for VM system options that you can change dynamically, defined in the ACFFDR. The exceptions are the @SMF macro parameters (SMF record numbers are changed), and the @ADDRSIZ and @SYSID macro parameters.

After you reload the ACFFDR, we recommend that you issue the appropriate SHOW subcommand to ensure that CA ACF2 for VM accepts the new value.

Issue the ACFSERVE RESTART command to change any of the SMF parameters. You must have the ACCOUNT and SECURITY (unscoped) privileges in your logonid record to issue this command. The syntax for this command is:

```
ACFServe Reload FDR
```

Reloading the LMP Keys from the CALMP KEYS File

The ACFSERVE RELOAD LMP command causes the CALMP KEYS file on the ACF2VM service machine's 193 minidisk to be processed and checked for errors. Messages from LMP key processing are sent to the issuer of the ACFSERVE RELOAD LMP command and also to the ACF2VM service machine console and Operator console. You must have the ACCOUNT and SECURITY (unscoped) privileges in your logonid record to issue this command. The syntax for this command is:

```
ACFServe Reload LMP
```

Reloading the Profile Records

This ACFSERVE command applies only to POSIX (OpenExtensions) and LINUX environments. Whenever you create, change, or delete a USER or GROUP profile record, you must issue the appropriate ACFSERVE command to rebuild the directories so CA ACF2 for VM recognizes these changes.

```
ACFServe Reload PROfile { OEVM  } <ALL|USER|GROUP>
                        { LINUX }
```

The old syntax for OpenExtensions tables is also still supported:

```
ACFServe Reload PROfile { USER  }
                        { GROUP }
```

PROfile OEVM

Indicates that OpenExtensions tables should be rebuilt.

PROfile LINUX

Indicates that Linux tables should be rebuilt.

ALL

Indicates all tables of the requested type of Profile records should be rebuilt. This is the default.

USER

Indicates that only the User Profile records should be reloaded and the corresponding table rebuilt.

GROUP

Indicates that only the Group Profile records should be reloaded and the corresponding table rebuilt.

Reloading Resource Resident Type Lists

The ACFSERVE RELOAD RESOURCE command reloads and creates the specified resource type in the resident type list. It builds or rebuilds the resident type list for the specified resource type. You do not have to define the resource type in the RESTYPE VMO record. Whenever you add a resource rule set, you should also reload the associated resource type in the resident type list to activate the new rule set. You should only issue this command when you add a new resource rule. You do not have to issue the ACFSERVE command if you are updating an existing rule.

You must have the ACCOUNT and SECURITY privileges in your logonid record to issue this command. The resource types not previously defined in the RESCLASS VMO record that are added to the resident type list through this command are temporary. The syntax for this command is:

```
ACFServe Reload RESource typ
```

typ

Specifies the three-character resource type you are reloading into the resident type list.

Reloading Access Rule Sets

The ACFSERVE RELOAD RULE command reloads the access rule set you specified into the rule cache. When the rule set is required for access validation, CA ACF2 for VM reloads the most current version of the rule set into the cache. If you are sharing databases, issue this command on all systems running CA ACF2 for VM after a rule set is updated.

You must have the SECURITY privilege in your logonid record to issue this command. The ruleid must also be in your scope. The syntax for the ACFSERVE RELOAD RULE command is:

```
ACFServe Reload RULe ruleid
```

ruleid

Indicates the reloaded rule set identified by its ruleid, as specified by the $KEY control statement.

Reloading Shift Information

The ACFSERVE RELOAD SHIFT command reloads the resident matrix tables containing shift information. You must issue this command when you modify a shift or zone record to activate the change. If you do not issue this command, changes to shift and zone records do not take effect until you IPL the CA ACF2 for VM service machine. You must have the SECURITY privilege in your logonid record to issue this command. The syntax for this command is:

```
ACFServe Reload SHift
```

Reloading XREF Tables

The ACFSERVE RELOAD XREF command builds or reloads the CA ACF2 for VM cross-reference tables. You must have the CA ACF2 for VM SECURITY privilege to issue this command. The syntax for this command is:

```
ACFServe Reload XRef {SGP|RGP}
```

SGP, the default, dynamically updates the source entry cross-reference table with both E-SGP and X-SGP records. Issue this command whenever you modify an entry source group (ESGP) or a source group XREF (XSGP) record to activate the change. If you do not issue this command, changes to E-SGP or X-SGP records do not take effect until the next IPL of the CA ACF2 for VM service machine.

RGP dynamically updates the resource cross-reference table with X-RGP records. Issue this command whenever you modify a resource group (XRGP) record to activate the change. If you do not issue this command, changes to X-RGP records do not take effect until the next IPL of the CA ACF2 for VM service machine.

To activate new resource rules that correspond to new or changed X-RGP groups, you need to build a resident type list so that resource group validation processing occurs. See the Reloading Resident Matrix Tables.

Lowering Password Violation Counts

The ACFSERVE RESET command lowers the password violation count by one for the specified logonid to enable one more attempt at matching the password. You must have the ACCOUNT or SECURITY privilege in your logonid record to issue this command. The logonid must also be in your scope. The syntax for this command is:

```
ACFServe RESET logonid
```

logonid

Represents the logonid to be reset.

Restarting the CA ACF2 for VM Service Machine

The ACFSERVE RESTART command gradually quiesces the CA ACF2 for VM service machine and restarts it. We recommend this method of quiescing CA ACF2 for VM. We do not recommend using the CP FORCE command or the CMS HX command. You must have the unscoped SECURITY privilege in your logonid record to issue this command. The system operator can also issue this command. The syntax for this command is:

```
ACFServe RESTART [dump]
```

dump

Represents an optional operand that causes the CA ACF2 for VM service machine to perform a dump before restarting.

The CA ACF2 for VM service machine finishes any current requests, puts the requests made after you issued the ACFSERVE RESTART command into a wait mode, closes all database files, then logs off. Then, the CA ACF2 for VM service machine is autologged and restarted. You should only use the ACFSERVE RESTART command in emergency situations. For example:

- Loading a new CA ACF2 for VM service machine module
- Replacing an exit
- Changing the MAXLIDS value
- Changing the SMF parameters.

Changes to rules on one system in a shared database environment do not affect a rule cached on another system. When you restart the CA ACF2 for VM service machine, all caches are cleared and reset. The ACFSERVE RESTART command does not refresh the cache. The refresh is a byproduct of IPLing CMS. The restart function can adversely affect system performance.

You must format and reserve any new disks added to the DISKLST operand of the @SMF macro. See the *Installation Guide* for detailed information.

The ACFSERVE RESTART command does an internal reload of the ACFFDR. All values in the ACFFDR are refreshed except those the @ADDRSIZ and @SYSID macros define.

The CA ACF2 for VM service machine detects unexpected error conditions and performs error processing. This capability reduces the possibility that the CA ACF2 for VM service machine hangs through a VM READ or disabled wait state until CP forces the CA ACF2 for VM service machine off or you issue a CP FORCE command. For example, the CMS HX command stops the service machine and normally causes a VM READ. The CA ACF2 for VM service machine intercepts detect this error condition, take a dump, and logoff.

Tracing SAF Events

The ACFSERVE SECTRACE command creates SECTRACE requests to trace SAF events (RACROUTE requests) that are passed to CA ACF2 for VM by the CA-ESM component.

The syntax for this command is:

```
ACFServe SECtrace SET traceid options
ACFServe SECtrace MODify traceid|ALL options
ACFServe SECtrace DELete traceid|ALL
ACFServe SECtrace DISAble traceid|ALL
ACFServe SECtrace ENable traceid|ALL
```

The operands for this command are defined below.

traceid

Specifies the eight-character name for the SECTRACE request. ALL can be specified for the traceid in all SECTRACE subcommands, except for SET, to perform the action on all SECTRACE requests.

Options

For SET and MODIFY, the options are described below. There is an ACFSERVE command limit of seven words after the SET or MODIFY subcommands. If you need to specify more operands, use SET to create the trace request and MODIFY to further define the trace request.

Request REQ|ALL

Specifies the RACROUTE request function to trace or ALL to trace all requests. If the request option is not specified, all requests are traced.

User username|ALL

Specifies the user to trace or ALL to trace all users. If the user option is not specified, all users are traced.

Servmach machineid|ALL

Specifies the service machine to trace or ALL to trace all service machines. This is a service machine that is doing SAF security requests communicated through the CA-ESM component. If the servmach option is not specified, all machines are traced.

Dest destid

Specifies the destination user that the trace messages are sent to. If the dest option is not specified, the trace messages are sent to the SECTRACE CMD issuer.

Limit limitval

Specifies the maximum number of events to trace. If the limit option is not specified, 100 is the maximum.

RESET

Resets the count of the events for this request to zero, allowing for another set of traces to be done up to the LIMIT value.

For example, to trace all REQUEST=AUTH calls done by the SFSPROD service machine in a trace request called SFSTRACE, issue:

```
ACFSERVE SECTRACE SET SFSTRACE REQUEST AUTH SERVMACH SFSPROD
```

To display the existing SECTRACE requests that have already been defined, use the ACFSERVE QUERY SECTRACE command under Displaying SECTRACE Requests.

Setting the SYSID

The ACFSERVE SET SYSID command changes the currently active CA ACF2 for VM SYSID. CA ACF2 for VM uses the currently active SYSID as the default SYSID for the ACF commands CONTROL mode, and as the default SYSID for the ACFSERVE RELOAD CONTROL VMO command. You must have the SECURITY privilege in your logonid record to issue the ACFSERVE SET SYSID command. The syntax for this command is:

```
ACFServe SET SYSID sysid|* [PERM]
```

The operands for this command are defined below.

SYSID sysid|*

Specifies the value that becomes the new currently active SYSID. If you specify an asterisk (*), CA ACF2 for VM uses the startup SYSID value. A SYSID value is required.

PERM

Specifies that CA ACF2 for VM saves the new SYSID value in the startup options file if you have defined one for your system and are using it. (See the RPROMPT and PRDISK operands of the VMXAOPTS macro in HCPAC0. If a startup options file is not available, CA ACF2 for VM issues a warning message and sets the current SYSID to the sysid value.)

To display the ACFFDR @SYSID macro value, the startup SYSID, the current SYSID, and the last user that issued the ACFSERVE SET SYSID command, enter the following command:

```
SHOW SYSTEMS
```

Switching to a New SMF File

The ACFSERVE SWITCH SMF command stops the CA ACF2 for VM service machine writing to the currently active SMF file and switches to a new SMF file. You must have the SECURITY privilege in your logonid record to issue this command. The syntax for this command is:

```
ACFServe SWitch SMF
```

Revalidating Your Logon Password

The ACFSERVE VALIDATE PASSWORD command prompts the user to enter his logon password for validation. Incorrect passwords are counted as password violations, so the user can be suspended after too many incorrect passwords are entered. The user can also logoff or disconnect from the system at this prompt. You cannot terminate this command unless you enter the correct password or enter LOGOFF or DISConnect. You do not need any special privilege to use this command. The syntax for the ACFSERVE VALIDATE PASSWORD command is:

```
ACFServe VALidate PASsword
```

ACFSERVE Command Return Codes

Shown below is a table of ACFSERVE return codes and associated message information.

Return Code	Message	Message Text
000	ACFpgm373I	<dsn> DDSN selected at startup
	ACFpgm374I	Using default DDSN - none selected at startup

Return Code	Message	Message Text
	ACFpgm432I	Release <rel> - Level <lev>.<seq> - Status <status>
	ACFpgm76AI	Service machine started for updates
	ACFpgm76BI	Service machine stopped for updates
	ACFpgm994I	Processing of LMP keys is complete
004	ACFpgm419E	CA ACF2 for VM not active
	ACFpgm432I	Release <rel> - Level <lev>.<seq> - Status <status>
008	ACFpgm442E	<userid> is not a groupid
	ACFpgm444E	<userid> has no active batch user
012	ACFpgm411E	Invalid ACFSERVE command operand: <operand>
	ACFpgm412E	Excessive number of ACFSERVE command operands
	ACFpgm414E	Invalid ACFSERVE subcommand: <subcommand>
	ACFpgm731E	Required operand missing
	ACFpgm76CE	Service machine in use by <userid>
016	ACFpgm443E	<userid> is not logged on
	ACFpgm445E	No active batch users found
	ACFpgm446E	No groupids found
	ACFpgm76DE	Command valid only in NOAUTO mode
024	ACFpgm449E	“<recid>“ is not a valid ACFSERVE record ID
	ACFpgm460E	Recid “<recid>“ is not reloadable
	ACFpgm462E	“<recid>“ is not valid VMO record ID
	ACFpgm76EE	ENABLE NOAUTO failed
032	ACFpgm4A7E	CP errors occurred setting VM system options, RC=<rc>, EC=<errcode>
	ACFpgm46DE	CP errors occurred setting ACFSERVE privileges, RC=<rc>, EC=<errcode>
040	ACFpgm413E	IUCV error <rc> processing ACFSERVE command

Return Code	Message	Message Text
124	ACFpgm333E	VMDUMP for ACFSERVE RESTART DUMP failed, VMDUMP RC=<rc>
913	ACFpgm416E	ACFSERVE command security violation
	ACFpgm434E	Need unscoped <attribute> privilege for this ACFSERVE subcommand

Using ACFSERVE Commands in NOAUTO Mode

The ACFSERVE command is functionally limited when CA ACF2 for VM is in NOAUTO mode. You can issue only these ACFSERVE commands in NOAUTO mode:

- ACFSERVE DISABLE NOAUTO UPDATES
- ACFSERVE ENABLE NOAUTO UPDATES
- ACFSERVE QUERY DDSN
- ACFSERVE QUERY SMF
- ACFSERVE QUERY STATUS
- ACFSERVE RELOAD FDR
- ACFSERVE RELOAD CONTROL
- ACFSERVE SET SYSID

Chapter 25: Advanced Logon Procedures

This chapter is aimed at more advanced CA ACF2 for VM users who need to maintain logonids for virtual machines other than their own.

When you finish this chapter, you will know about:

- DirMaint and other directory passwords
- Logging on with the AUTOONLY privilege
- Logging onto group machines
- Logging on as a group user
- Logging on as a group machine
- Logging on with LOGON-BY
- Executing AUTOLOG without a password
- How to access the system when CA ACF2 for VM is not active.

This section contains the following topics:

DirMaint Password Information

When you issue a DirMaint command, CA ACF2 for VM validates the CA ACF2 for VM password instead of the directory password.

```
dirm rev
(press enter)

DVHDIR014I FILE CMS1 DIRECT WILL BE SENT TO YOUR READER

DVHDIR005R ENTER CURRENT CP PASSWORD TO VALIDATE COMMAND
OR A NULL TO EXIT:

password  (nondisplay)
(press enter)

PUN FILE xxxx FROM DIRM2  COPY 001  NOHOLD
 DVHMAC035I FILE CMS1 DIRECT SENT VIA CMS PUNCH COMMAND
```

You must enter your CA ACF2 for VM password correctly, even though the system reply asks you to enter the CP password. If you do not, the command is rejected.

Other Password Information

There are other points to consider about CA ACF2 for VM passwords that we explain below.

- The NOLOG directory password prevents a user from logging onto the system.
- Password suppression is an IBM security feature implemented in the PSUPRS operand of the SYSJRL macro of HCPSYS. It forces you to enter your password on a separate line from your logonid when you log on. This password is not displayed. If you turn password suppression off, you can enter the LOGON command, your logonid, and your password on the same line. Let's look at the example below.

  ```
  logon TLCAMS password (enter one space before and after your logonid)

  (press enter)
  ```

  You should not enter all of this information on one line because your password is displayed on the screen and a nearby observer can read it. We recommend that you keep password suppression turned on at all times.

AUTOONLY Information

The AUTOONLY logonid privilege, a field in the PRIVILEGES group of logonid records, prevents a virtual machine from logging on. You should not associate AUTOONLY with a virtual machine. If your logonid has AUTOONLY turned on, the following message appears when you try to log on:

```
ACFpgm279E Logon not allowed - logonid has AUTOONLY attribute
```

The unsuccessful system access attempt is reported in the Invalid Password/Authority Log (ACFRPTPW).

Logging onto CA ACF2 for VM Group Machines

It is sometimes necessary to have a virtual machine that several users can access. This machine is called a group virtual machine (or maintenance machine) and usually has a centralized function. Through CA ACF2 for VM, you can assign the group virtual machine the GRPLOGON privilege (a field in the Privileges group of the logonid record) to define a special kind of group virtual machine.

Many people can use a group virtual machine that has the GRPLOGON privilege, however, only one person can access the machine at any given time. What is special about a GRPLOGON machine is that when someone tries to use it, CA ACF2 for VM validates attempts in a unique way. CA ACF2 for VM determines who is actually using the machine. This way, everyone who uses the group virtual machine (group users) are identified through auditing. In addition, many users can use the group machine without knowing that logonid's password.

How to Log onto a CA ACF2 for VM Group Machine

This method is for CA ACF2 for VM group virtual machines.

```
logon cmsgroup (with one space before group-logonid)
(press enter)

logon
ACFpgm263R Enter ACF2 logonid

TLCAMS
(press enter)

ACFpgm244R Enter your ACF2 password

password  (nondisplay)
(press enter)

ACFpgm137I TLCAMS last system access at 17.38 on 11/09/97 from GRAF-419
ACFpgm137I CMSGROUP last system access at 21.01 on 10/27/97 from GRAF-49
ACFpgm268I Group logon successful to CMSGROUP by TLCAMS
  ...
LOGON AT 17:45:09 CST TUESDAY 09/09/98
```

After entering logon group-logonid to use the group machine, you supply your own logonid and password, **not** the group machine's.

When you are in the system, all validations are performed as if the group virtual machine were logged on as a regular user. Normal data and resource access validation is performed against the group machine, not the group user.
CA ACF2 for VM creates SMF records on behalf of the group machine. SMF records contain information that identifies the group user. The user who logged onto the group virtual machine is identified in the CA ACF2 for VM reports. This enforces individual accountability.

You can also use the ACFSERVE RELOAD RESOURCE command to reload and create new group IDs. You can find more information for using this command in the chapter “Using the ACFSERVE Commands” in this guide.

For details on how the CA ACF2 for VM group logonid feature works with CA ACF2 for VM report generators, the *Reports and Utilities Guid*e.

Important factors to consider about this implementation include:

- When you try to log onto a group virtual machine, CA ACF2 for VM checks a resource rule to see if you can use the machine. This occurs regardless of the CA ACF2 for VM data access mode setting and any special privileges you might have. The rule checking and implementation requirements for the rule check are explained in the chapter titled “Writing Resource Rules” of the CA ACF2 for VM *Implementation Planning Guid*e.
- If the OPTS VMO record defines a VMCHK attribute, that attribute is validated for the group machine but not for the group user.
- A CA ACF2 for VM group machine cannot log on or issue the DIAL command. The GRPLOGON attribute does not affect the ability to use the AUTOLOG and XAUTOLOG commands.
- You do not need to define a group user as a user ID in the VM directory. The only prerequisite for a group user is that you must define the user with a logonid record in the Logonid database. You must also define the group machine with a logonid record in the Logonid database and in the VM directory.
- Password suppression is an IBM security feature implemented in the PSUPRS operand of the SYSJRL macro. It forces you to enter your password on a separate line from your logonid when you log on. This password is not displayed. If this feature is turned on, CA ACF2 for VM does not accept a group user's password on the same line as the logonid. For example, if a group user tries to log onto the MAINT group machine as shown below, CA ACF2 for VM sends a message and ignores the password.

  ```
  logon maint password
  (press enter)

  ACFpgm278I Logon for groupid - password entered will be ignored
  ```

- The AUTOONLY privilege prevents you from logging onto a virtual machine. This is also true if the machine is a group virtual machine.

Using LOGON-BY

LOGON-BY lets you log onto another user ID and allows IDs to be optionally shared when a group logon resource rule exists that allows the access.

Optional group IDs allow users to share personal IDs. A logonid record is designated as an optional group ID when the GRP-OPT attribute is present in the logonid record.

Important: In cases where a logonid has both the GRPLOGON attribute and the GRP-OPT attribute, GRPLOGON takes precedence. This means that the user ID cannot log onto itself.

LOGON-BY lets you log onto another user ID if a group logon resource rule exists that allows the access. Enter the LOGON-BY syntax to do this. When you use LOGON-BY to access another user ID, that ID becomes a group ID for the duration of the session.

LOGON-BY provides these benefits:

- Allows authorized users access to VM user IDs without password sharing
- Incorporates VM industry standard LOGON-BY command syntax
- Improves logon sequence for the group logon feature
- Allows user ID owners to designate other users to access their user IDs
- Identifies users in all CA ACF2 for VM audit trails by the user ID and accessing logonid.

The RESCLASS VMO record controls whether a user ID can be a group ID. This lets you gradually phase in the LOGON-BY feature.

Logging on with LOGON-BY

We currently support three syntax variations for LOGON-BY. We recommend that you use number one because it ensures that CA ACF2 for VM does not display your password. Below are examples of the syntaxes:

```
LOGON lid1 BY lid2
```

lid1

Specifies the user ID you are logging onto.

lid2

Specifies your own CA ACF2 for VM user ID.

For example, if Ann Smith (TLCAMS) wants to access Pete Miller's user ID (TLCPJM), she would enter the following:

```
LOGON TLCPJM BY TLCAMS
```

CA ACF2 for VM would then prompt Ann for her own CA ACF2 for VM password:

```
LOGON lid1 BY lid2 password
```

For example, if Ann Smith (TLCAMS) wants to access Pete Miller's user ID (TLCPJM) using this syntax, she would enter the following:

```
LOGON TLCPJM BY TLCAMS password
```

```
LOGON lid1
```

For example, if Ann Smith (TLCAMS) wants to access Pete Miller's user ID (TLCPJM) using this syntax, she would enter the following:

```
LOGON TLCPJM
```

CA ACF2 for VM then prompts Ann for her password and she must enter:

```
BY/TLCAMS/password
```

Enter the LOGON-BY syntax (including slashes) exactly as shown above.

You can also issue the LOGON-BY syntax from the full-screen logo as shown in the following example:

```
z/VM ONLINE

                                                  / VV          VVV MM       MM
                                                 /  VV         VVV  MMM     MMM
                                        ZZZZZZ     /   VV       VVV   MMMM   MMMM
                                           ZZ     /    VV     VVV    MM MM MM MM
                                          ZZ     /     VV  VVV      MV  MMM  MM
                                         ZZ     /      VVVVV       MV   M   MM
                                        ZZ     /       VVV        MV       MM
                                       ZZZZZZ /        V         MV       MM

                                         built on IBM Virtualization Technology









Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)
USERID   ===> USER01
PASSWORD ===> BY/USER02/password

COMMAND  ===>
                                                                    RUNNING   ZVMSYS01
```

Enter the LOGON-BY syntax in the PASSWORD field. Use the same syntax shown in the third example on the preceding page.

With LOGON-BY, a group ID cannot log onto itself. If your site allows this, you must remove the GRPLOGON attribute and specify the GRP-OPT attribute to change those IDs from mandatory group IDs to optional group IDs. Then use the correct LOGON-BY syntax and let the group logon resource rules govern the accesses.

Implementation Considerations

If you are presently allowing users to log onto group IDs as themselves, you must remove the GRPLOGON attribute and specify GRP-OPT to change those IDs from mandatory group IDs to optional group IDs.

To decentralize optional group IDs and let users make their IDs optional group IDs, this is what you need to do:

- Change the SECURITY and ACCOUNT privileges for the GRP-OPT attribute. If you want to allow users to designate the ID as an optional group ID, you might want to change the ALTER= authority on the field definition in the ACFFDR.
- Write resource rules. (Only a security administrator can write these resource rules.) To allow users to write their own rules to let others access their user IDs, a security administrator must write a skeleton rule with the key of the user ID and a %CHANGE control statement with the user ID in it.

Executing AUTOLOG without a Password

With special CA ACF2 for VM AUTOLOG procedures, you can execute the AUTOLOG command without entering a password and not compromise system security. This is useful because it lets you AUTOLOG from uninterrupted execs. To enter AUTOLOG with CA ACF2 for VM, enter the command followed by the name of the machine that is autologged.

```
autolog cmsauto (you can specify a group-logonid as the logonid)

ACFpgm137I CMSAUTO last system access at 17.22 on 11/09/97 from GRAF-490
AUTO LOGON  ***  autologged-logonid USERS = 043
```

Depending on certain CA ACF2 for VM VM logonid privileges defined for the machine performing the autolog or the machine being autologged (as discussed shortly), there is no prompt for a password. The machine being autologged logs on automatically. After you have entered the system, CA ACF2 for VM validates data and resource access against the machine that is autologged, not the machine performing the autolog.

With CA ACF2 for VM, you can specify the name of the target machine and its password to enter the AUTOLOG command. A password prompt does not appear.

CA ACF2 for VM protects the AUTOLOG command in a unique way. System entry for executing the AUTOLOG command through CA ACF2 for VM consists of the following steps:

1. AUTOLOG command limiting validation. This step is optional, just as it is for any other CP command. See the *Command and Diagnose Limiting Guide* for details.
2. AUTOLOG resource rule validation. Whenever you try to autolog a virtual machine, the system automatically checks a resource rule to see if you are allowed access. This occurs regardless of the CA ACF2 for VM data access mode setting and any special privileges you might have.

3. Conditional password validation that determines you can execute AUTOLOG without entering a password. The system does not prompt for a password if the logonid of the machine being autologged has the AUTONOPW privilege (in the PRIVILEGES group of the logonid record) or the logonid of the machine executing the autolog has the AUTOALL privilege (also in the PRIVILEGES group of the logonid record).

   On the other hand, if none of these conditions are met, CA ACF2 for VM prompts for the password of the autologged machine with the standard password prompt:

   ```
   ACFpgm244R Enter CA-ACF2 password
   ```

   The autologged machine can be a group virtual machine (a machine with the GRPLOGON privilege). If this is the case and you receive a password prompt, enter the password of the autologged machine (the group virtual machine).

   For XAUTOLOG: If the same conditions as above are not met when executing XAUTOLOG, CA ACF2 for VM issues error message 1690E:

   ```
   XAUTOLOG failed - password required
   ```

   CA ACF2 for VM then displays:

   ```
   XAUTOLOG logonid PROMPT
   ```

   The standard password prompt follows the message:

   ```
   ACFpgm244R Enter CA-ACF2 password
   ```

   To assign virtual machines the special logonid privileges of AUTONOPW and AUTOALL, see the *Implementation Planning Guid*e.

4. User IDs for group logonids (group machines). At this point of the AUTOLOG validation process, the user who issued the AUTOLOG command is authorized to autolog a virtual machine. This step preserves the individual accountability of group machines (machines with the GRPLOGON privilege). This is done for audit purposes. CA ACF2 for VM checks the logonid of the machine being autologged for the GRPLOGON privilege. If present, the name of the machine performing the autolog appears in all CA ACF2 for VM reports with the name of the autologged group machine. This corresponds with how CA ACF2 for VM generates reports for group virtual machines. For details on how the CA ACF2 for VM group logonid feature works with the CA ACF2 for VM report generators, see the *Reports and Utilities Guid*e.

The name of the original machine that performed the autolog continues to appear in the reports for any machine that an autologged group machine subsequently autologs. The original group user of the group machine is accountable for all of that machine's subsequent actions.

A machine with the GRPLOGON privilege does not undergo group logon resource validation for group machines that are being autologged. (The machine has already gone through autolog resource validation.) Group logon resource validation applies only when group machines are trying to gain system entry through logon procedures. That is where the name GRPLOGON (GRouP LOGON) is derived from.

5. The machine executing the AUTOLOG command must have CP privilege class A or B.
6. Password suppression is an IBM security feature implemented in the PSUPRS operand of the SYSJRL macro. It forces you to enter your password on a separate line from your logonid when you log on. Your password is not displayed when you enter it.

   If password suppression is turned off and the machine you are autologging has the AUTONOPW privilege or your machine has the AUTOALL privilege, enter a dummy password of at least one character.

```
autolog TLCAUTO
(press enter)

AUTO LOGON  *** CMSAUTO USERS = 043
```

   We recommend, however, that you have password suppression turned on. This forces you to enter your password separate from your logonid so your password is not visible on the screen. If password suppression is turned off and you are autologging a group virtual machine, CA ACF2 for VM does not accept any password on the same line as the logonid.

System Access with CA ACF2 for VM Inactive

CA ACF2 for VM is automatically started (autologged) after VM system initialization has completed and a user other than OPERATOR or AUTOLOG1 tries to log on. Generally, this occurs when the disconnected service machines are being autologged through AUTOLOG1's PROFILE EXEC.

If CA ACF2 for VM is not active, VM is not usable because CA ACF2 for VM CP modules refuse all logons and data accesses. If this ever happens, a special
CA ACF2 for VM feature lets systems programmers log onto the system so that they can correct the problem. Users specified in the FORCEID operand of the VMXAOPTS macro can use this facility. See the *Systems Programmer Guid*e for complete information about the VMXAOPTS macro.

These special users can log on and access data while CA ACF2 for VM is inactive if a system emergency has been declared. At the CA ACF2 for VM system initialization message, reply:

```
NOAUTO
```

Consider the example shown below.

```
ACFpgm000R Enter ACF2 startup options or press enter

noauto
```

This response suppresses automatic startup of the CA ACF2 for VM service machine, letting only FORCEID users log on during the emergency. No one else can access the system. Also, activities that would normally require CA ACF2 for VM validation are automatically allowed for FORCEID users.

If NOAUTO=DIRPASS in the VMXAOPTS macro, FORCEID users must enter a valid directory password when they log on. The default setting of NOAUTO=NOPASS does not require a password. After you take corrective action, IPL the system again to resume normal operation and security controls. When CA ACF2 for VM is up and running, CA ACF2 for VM treats FORCEID users like any other user (access privileges are granted according to the user's logonid record privileges and CA ACF2 for VM access rules).

Chapter 26: Advanced Commands

For more experienced CA ACF2 for VM users, we provide three CMS commands for processing rules:

ACFCOMP

Lets you compile one or more rule sets

ACFDCMP

Lets you decompile one or more rule sets

ACFNRULE

Lets you add or delete individual rule entries.

This chapter provides complete information for using these commands.

This section contains the following topics:

Using ACFCOMP to Compile Rules

The ACFCOMP command lets you compile and store access rule sets quickly and easily. You can enter rule text from the terminal or from a CMS file. You can also use ACFCOMP to process resource rule sets. The syntax for the ACFCOMP command is:

```
ACFCOMP {*           }   [(List|(NOList   ]
        {fn {ft fm}  }   [(Store|(NOStore ]
                         [(Force|(NOForce ]
```

The operands for ACFCOMP are defined below.

*****

Specifies that you are entering rule text from a terminal. CA ACF2 for VM prompts you for the rule text. After you enter all the rule text, press Enter to indicate you are done.

fn ft fm

The filename, filetype, and filemode of the CMS file that contains the rule text. If this file contains multiple rule sets, CA ACF2 for VM compiles all the rule sets. You only need to specify the filename (fn). By default, CA ACF2 for VM assumes the filetype is RULE and the filemode is A.

(List|(NOList)

LIST Specifies that CA ACF2 for VM display the compiled rule sets at the terminal. NOLIST suppresses the display. The default is LIST.

Store|(NOStore

STORE specifies that CA ACF2 for VM automatically store the compiled rule sets on the Rule database if the compile is successful. NOSTORE compiles the rule sets, but does not store them. This makes NOSTORE a useful option for testing. STORE is the default.

Force|(NOForce

FORCE specifies that CA ACF2 for VM store the compiled rule sets if a rule with the same $KEY already exists. In effect, CA ACF2 for VM replaces the existing rule with the newer rule. NOFORCE suppresses the store if a rule with the same $KEY exists. When you compile multiple rule sets and specify NOFORCE, CA ACF2 for VM only stores rules that do not exist on the Rule database.

The following sections illustrate how you can use ACFCOMP to compile access rule sets.

Compiling a Rule for Display at the Terminal

Specify the command with the filename of the input file. In this form, ACFCOMP assumes the filetype is RULE and the filemode is A. For example:

```
ACFCOMP DOCTST

ACFCOM526I Processing input file 'DOCTST RULE A'

ACFCMP510I ACF compiler entered

$KEY(DOCTST)
 V0191.JD- UID(*)
 V0191.VOLUME UID(*) READ(A) WRITE(A) EXEC(A)
 V0191.- READ(A) WRITE(A) EXEC(A)
 V0192.- UID(*) READ(A) EXEC(A)
 - UID(****DOC) READ(A) EXEC(A)
ACFCMP551I Total record length=205 bytes - 5 percent utilized
ACFCOM726I Rule DOCTST replaced

Ready; T=0.03/0.17 11:23:13
```

In the above example, the defaults of LIST, STORE, and FORCE list the rule set during the compile and store it after the compile completed successfully. The last line displayed in the example (RULE DOCTST REPLACED) indicates that the compiled rule set replaced the previous rule for DOCTST.

Specifying the Complete CMS File Parameters

You can provide the complete filename, filetype, and filemode of the CMS file. For example:

```
ACFCOMP DOCTST FILE D (NOLIST

ACFCOM526I Processing input file 'DOCTST FILE D'

ACFCMP510I ACF compiler entered

ACFCMP551I Total record length=205 bytes - 5 percent utilized
ACFCOM726I Rule DOCTST replaced

Ready; T=0.04/0.16 11:26:19
```

In the above example, the defaults of STORE and FORCE store the rule set after the compile completed successfully. The last line displayed in the example (RULE DOCTST REPLACED) indicates that the rule set replaced the previous rule for DOCTST. The NOLIST operand suppressed the display of the rule being compiled.

Using NOSTORE to Test Rules

You can use the NOSTORE operand to test whether the rule sets in the CMS file compile successfully. With NOSTORE, CA ACF2 for VM compiles all rules in the input file, but does not store them. For example:

```
ACFCOMP DOCTST FILE D (NOSTORE

ACFCOM526I Processing input file 'DOCTST FILE D'

ACFCMP510I ACF compiler entered

$KEY(DOCTST)
 V0191.JD- UID(*)
 V0191.VOLUME UID(*) READ(A) WRITE(A) EXEC(A)
 V0191.- READ(A) WRITE(A) EXEC(A)
 V0192.- UID(*) READ(A) EXEC(A)
 - UID(****DOC) READ(A) EXEC(A)
ACFCMP551I Total record length=205 bytes - 5 percent utilized

Ready; T=0.04/0.16 11:27:22
```

In the above example, CA ACF2 for VM did not store or replace any rule sets-only compile them. If an error occurs during the compile, CA ACF2 for VM issues an error message.

Using NOFORCE to Store Only New Rules

There can be times when you want to store a rule set only if it does not already exist. You can use the NOFORCE operand to do this. For example:

```
ACFCOMP DOCTST RULE D (NOLIST NOFORCE

ACFCOM526I Processing input file 'DOCTST RULE D'

ACFCMP510I ACF compiler entered

ACFCMP551I Total record length=205 bytes - 5 percent utilized
ACFRUM097E Record already exists - store rejected

Ready(00003); T=0.03/0.15 11:28:53
```

In the above example, CA ACF2 for VM did not store or replace a rule set because a rule set already existed for DOCTST. The NOLIST suppressed display of the rule being compiled.

Entering Rule Input Directly from the Terminal

You can use the ACFCOMP command to enter rule input directly from the terminal. For example:

```
ACFCOMP (STORE

ACFCOM527I ACF2/VM interactive compiler
$KEY(DOCTST)
 (press enter)
 V0191.JD- UID(*)
 (press enter)
 V0191.VOLUME UID(*) READ(A) WRITE(A) EXEC(A)
 (press enter)
 (press enter again to signal end of terminal input)

ACFCMP510I ACF compiler entered

ACFCMP551I Total record length=130 bytes - 3 percent utilized
ACFCOM726I Rule DOCTST replaced

Ready; T=0.03/0.18 11:31:22
```

In the above example, the rule input is entered in interactive mode directly from the terminal.

Using ACFDCMP to Decompile Rules

The ACFDCMP command quickly decompiles access rule sets to the terminal or to a CMS file. ACFDCMP also decompiles resource rule sets. The syntax of the ACFDCMP subcommand is:

```
ACFDCMP  {ruleid       }  [TYPE rsrctype   ]
         {LIKE rulemask }  [INTO fn {ft fm} ]
                           [(APPEND|REPLACE ]
```

ruleid|LIKE rulemask

Specify the $KEY of the rule to decompile a single rule set. To decompile multiple rule sets, use the LIKE rulemask operand and CA ACF2 for VM decompiles all rule sets with a $KEY that matches the rulemask. The rulemask can be any valid mask, an asterisk (*), or a dash (-).

TYPE rsrctype

Processes resource rules. The rsrctype value must be the three-character resource type code where the resource rule is stored. You cannot mask the rsrctype. This operand is not valid for access rules.

INTO fn ft fm

Indicates the filename, filetype, and filemode (respectively) of a CMS file that contains the rule output. Only the filename fn is required. By default, the filetype is RULE and the filemode fm is A.

APPEND|REPLACE

These operands are valid only when decompiling rule sets into a CMS file. APPEND adds the decompiled rule sets to the end of the output CMS file. APPEND is the default. REPLACE overlays the decompiled rule sets to any existing rule sets in the output file.

Decompiling a Rule for Display at the Terminal

Specify the command and the $KEY of the rule set to decompile. For example:

```
ACFDCMP TLCAMS
ACFDC0511I Processing access rule 'TLCAMS'
*. ACFDCA556I ACCESS rule TLCAMS stored by MAINT on 11/02/97-11:31
$KEY(TLCTST)
 V0191.JD- UID(*)
 V0191.VOLUME UID(*) READ(A) WRITE(A) EXEC(A)
 V0191.- READ(A) WRITE(A) EXEC(A)
 V0192.- UID(*) READ(A) EXEC(A)
 - UID(****DOC) READ(A) EXEC(A)
*. ACFDCA551I Total record length=205 bytes - 5 percent utilized
Ready; T=0.02/0.12 11:32:33
```

In the above example, CA ACF2 for VM decompiled and displayed the rule TLCAMS at the terminal.

Decompiling and Adding a Rule Set to a CMS File

Use ACFDCMP to decompile a rule and add it to a CMS file. Use this subcommand to maintain a history of rule changes. For example:

```
ACFDCMP TLCAMS INTO TLCTST
*. ACFDCA551I Total record length=205 bytes - 5 percent utilized
*. ACFDCA556I ACCESS rule TLCAMS stored by MAINT on 11/02/97-11:31
Ready; T=0.03/0.12 11:33:48
```

In the above example, the TLCAMS rule is decompiled and placed in a CMS file named TLCAMS RULES A. If this file had existed before you issued the ACFDCMP command, CA ACF2 for VM would add the rule to the end of the file. If it did not exist, CA ACF2 for VM would create the file.

Decompiling and Replacing a Rule Set in a CMS File

Use ACFDCMP to decompile a rule and replace it if it exists in the file. This saves only the most recent rules. For example:

```
ACFDCMP TLCAMS INTO TLCTST RULES A(REPLACE
ACFDC0511I Processing access rule 'TLCAMS'
*. ACFDCA551I Total record length=205 bytes - 5 percent utilized
*. ACFDCA556I ACCESS rule TLCAMS stored by MAINT on 11/02/97-11:31
Ready; T=0.03/0.12 11:34:19
```

In the previous example, the TLCAMS rule is decompiled and placed into a CMS file named TLCAMS RULES A. If the rule existed in the file before you issued the ACFDCMP command, CA ACF2 for VM would replace the rule.

Decompiling Multiple Rule Sets

Use ACFDCMP to decompile multiple rules at one time. For example:

```
ACFDCMP LIKE TLC- INTO TLCRULES RULES A
ACFDC0511I Processing access rule 'TLC'
*. ACFDCA551I Total record length=205 bytes - 5 percent utilized
*. ACFDCA556I ACCESS rule TLC stored by MAINT on 11/02/97-11:31
ACFDC0511I Processing access rule 'TLCDLW'
*. ACFDCA551I Total record length=210 bytes - 5 percent utilized
*. ACFDCA556I ACCESS rule TLCDLW stored by MAINT on 10/02/97-11:31
ACFDC0511I Processing access rule 'TLCGDP'
*. ACFDCA551I Total record length=200 bytes - 5 percent utilized
*. ACFDCA556I ACCESS rule TLCGDP stored by MAINT on 11/30/97-11:31
Ready; T=0.08/0.35 13:08:03
```

CA ACF2 for VM decompiles all rule sets with a $KEY that match the TLC- mask and adds them to the CMS file.

Using ACFNRULE to Add Rule Entries

The ACFNRULE command adds individual rule entries to access rule sets. The changed rule is automatically compiled, sorted, and stored. ACFNRULE also processes resource rules.

The syntax for the ACFNRULE command is:

```
ACFNRULE {ruleid       } [TYPE(rsrctype)]
         {KEY(ruleid)} }

         {[ADD(ruleentry)]|[DELETE(ruleentry)]}

         [List|NOList      ]
         [Verify|NOVerify  ]
```

The operands for ACFNRULE are defined below.

ruleid|KEY(ruleid)

Indicates the $KEY value. To process a single rule set, specify the $KEY of the rule. Both ruleid and KEY(ruleid) function alike. You must issue this operand.

TYPE (rsrctype)

Processes resource rules. The rsrctype value must be the three-character resource type code where the resource rule is stored. You cannot mask this value. This operand is optional and you must omit it when processing access rules.

ADD(ruleentry)...

Inserts one or more individual rule entries into a rule set. For example, ADD(V0191.volume uid(TEST-)) inserts one rule entry into a rule set. You can ADD and DELETE (see description below) rule entries in a single command. You must specify at least one of these functions.

DELETE(ruleentry)...

Removes one or more individual rule entries from a rule set. For example, DELETE(V0191.volume uid(TEST-) deletes one rule entry from a rule set. You can DELETE and ADD (see description above) rule entries in a single command. You must specify at least one of these functions.

List|NOList

LIST displays the rule set at the terminal. NOLIST suppresses the display. The default is LIST.

Verify|NOVerify

These operands are valid only when you use the DELETE function. VERIFY displays each rule entry you are deleting, followed by a verification request. You must respond **Y** to delete the rule entry or **N** to suppress the delete. VERIFY is the default. NOVERIFY deletes rule entries without user verification.

The next section contains examples of how to use ACFNRULE to add individual rule entries from an access rule set or a resource rule set.

Adding a Rule Entry

To add a rule entry to an existing rule set, specify the rule ID of the rule and the new rule entry. For example:

```
ACFNRULE TLCAMS ADD(V0191.- UID(SHS-) READ(A))
ACFCMP510I ACF compiler entered

$KEY(TLCAMS)
 V0191.JD- UID(*)
 V0191.VOLUME UID(*) READ(A) WRITE(A) EXEC(A)
 V0191.- READ(A) WRITE(A) EXEC(A)
 V0192.- UID(*) READ(A) EXEC(A)
 - UID(****TLC) READ(A) EXEC(A)
*. ACFDCM551I Total record length=205 bytes - 5 percent utilized
V0191.- UID(SHS-) READ(A)
.dc cont ›
ACFCMP550W Execute access set to same value as read access in one›
or more rules
ACFCMP551I Total record length=229 bytes - 5 percent utilized
ACFULE727I Rule TLCAMS stored
Ready; T=0.04/0.18 12:55:01
```

In this example, a single new rule entry is added to the TLCAMS rule set. The default parameter of LIST displayed the updated rule set at the terminal.

Adding Multiple Rule Entries with One Command

You can add more than one rule entry to the rule set.

```
ACFNRULE TLCAMS ADD(V0192.- UID(SHS-)) ADD(V0193.VOLUME UID(*)) NOLIST
ACFCMP510I ACF compiler entered

ACFCMP551I Total record length=273 bytes - 6 percent utilized
ACFULE727I Rule TLCAMS stored
Ready; T=0.04/0.16 12:57:17
```

Two new rule entries were added to the TLCAMS rule set. The updated rule set is not displayed at the terminal because you specified the NOLIST parameter.

Processing Resource Rules

You can also use ACFNRULE to add resource rule entries. To process resource rules, include the three-character resource rule type after the rule ID. For example:

```
ACFNRULE ACF2 TYPE(CKC) ADD(UID(*) ALLOW)
ACFCMP510I ACF compiler entered

$KEY(ACF2) TYPE(CKC)
 UID(*****TLCAMS) VERIFY ALLOW
 UID(*****PAY) ALLOW
*. ACFDCM551I Total record length=176 bytes - 4 percent utilized
UID(*) ALLOW
ACFCMP551I Total record length=192 bytes - 4 percent utilized
ACFULE729I Resource CKCACF2 A stored
Ready; T=0.03/0.16 13:02:30
```

One entry is added to the resource rule stored under the type code CKC.

Using ACFNRULE to Delete Rule Entries

The ACFNRULE command deletes individual rule entries from access rule sets. CA ACF2 for VM automatically compiles, sorts, and stores the rules. You can also process resource rules with ACFNRULE. The syntax of the ACFNRULE command is:

```
ACFNRULE {ruleid      } [TYPE(rsrctype)]
         {KEY(ruleid) }

         {[DELETE(ruleentry)]}

         [ List|NOList     ]
         [ Verify|NOVerify ]
```

ruleid|KEY(ruleid)

Indicates the $KEY value of the rule. To process a single rule set, specify the $KEY of the rule you want to process. Both ruleid and KEY(ruleid) function alike. You must specify this operand.

TYPE rsrctype

Processes resource rules. The rsrctype value must be the three-character resource type code where the resource rule is stored. You cannot mask this value. This operand is optional and you must omit it when you process access rules.

DELETE(ruleentry)...

Removes one or more individual rule entries from a rule set. For example, DELETE(V0191.volume uid(TEST-)) deletes one rule entry from a rule set.

List|NOList

LIST displays the rule set at the terminal. NOLIST suppresses the display. The default is LIST.

Verify|NOVerify

These operands are valid only when you specify DELETE. VERIFY displays each rule entry you are deleting, followed by a verification request. You must respond **Y** to delete the rule entry or **N** to suppress the delete. VERIFY is the default. NOVERIFY deletes rule entries without user verification.

Examples of how you can use ACFNRULE to delete individual rule entries from an access rule set or a resource rule set follow.

Deleting a Rule Entry

ACFNRULE deletes individual rule entries from a rule set.

```
ACFNRULE TLCAMS DELETE(V193.VOLUME UID(*)) NOLIST
ACFULE874E The following entries will be deleted

 V193.VOLUME UID(*)

ACFULE875R Verify delete - reply 'Y', 'N' or '?' for individual prompts
Y
ACFCMP510I ACF compiler entered

ACFCMP551I Total record length=245 bytes - 5 percent utilized
ACFULE727I Rule TLCAMS stored
Ready; T=0.04/0.19 13:03:58
```

In the above example, V0193.VOLUME was deleted. Before it was deleted, it was displayed and a prompt was issued asking for verification to delete it. You can use this feature to delete multiple rule entries with one command.

Chapter 27: Backup and Recovery

CA ACF2 for VM provides complete facilities for the backup and recovery of its databases. These facilities include automatic daily backups and using alternate databases. This chapter explains how to implement the backup and recovery procedures for CA ACF2 for VM.

This section contains the following topics:

Prerequisites

This section covers the CA ACF2 for VM components that are important for backup and recovery.

BACKUP VMO Record

The BACKUP VMO record defines various backup options. Complete information on the BACKUP VMO record is in the chapter “Defining Structured Infostorage Records” of this guide.

ACFFDR @DDSN Macro

The @DDSN macro defines the name of a group of CA ACF2 for VM databases and the filenames (or VSAM data set names) of the three CA ACF2 for VM databases. You can define up to 16 @DDSN macros. This lets you have a primary set of databases and multiple sets of alternate databases. You must also define a @DDSN macro for backups. This could be any @DDSN, including PRIMARY, as long as it is for CMS databases. We recommend you define at least one set of alternate databases.

FORCEID Users

The FORCEID= operand of the VMXAOPTS macro in HCPAC0 defines the user IDs that can log on when CA ACF2 for VM is not active (NOAUTO mode).

Noauto Update Users

Besides FORCEIDs, CA ACF2 for VM also provides a user ID that enables authorized users to log onto CA ACF2 for VM in NOAUTO mode and update the databases. You must identify these user IDs with the NOAUTOU parameter in VMXAOPTS in HCPAC0. These user IDs become part of the NOAUTOU list. All user IDs defined in the NOAUTOU list must also be defined in the FORCEID list. The user IDs defined in both the NOAUTOU list and in the FORCEID list are called noauto update users. Only these users can log onto CA ACF2 for VM in NOAUTO mode and issue the appropriate ACFSERVE commands to update the databases. For more information on these ACFSERVE commands, see the "Updating the Databases in NOAUTO Mode" chapter.

ACFFDR @SMF Macro

The @SMF macro defines options related to SMF processing. This is a very important component for backup and recovery because it defines the number of defined SMF minidisks and their virtual addresses.

SMF Minidisks and Files

Database recovery procedures use historical SMF data. The recovery procedures can access the SMF data in any of the following formats:

- Old format SMF files.
- New format historical SMF files
- Dumped and archived SMF disks.
- ACFSMCOP copied and archived SMF disks. (For additional information on the ACFSMCOP utility, see the *Reports and Utilities Guid*e.
- New format online SMF disks. You must use current (online) SMF disks. They are volatile because a SMF disk can become active and reused for recording at any time unless the system was IPLed in NOAUTO mode.

Primary CA ACF2 for VM Databases

These are the CA ACF2 for VM databases normally in use. The first @DDSN macro in the ACFFDR specifies the primary databases. These databases usually have the @DDSN name of PRIMARY (@DDSN PRIMARY).

Alternate CA ACF2 for VM Databases

The @DDSN macros in the ACFFDR define alternate databases. Use them:

- To bring up the system with a minimum amount of delay if the primary databases are unavailable. If you do not merge SMF data with the alternate database before you bring up the system, you can lose changes to the databases.
- To speed up the recovery process if the primary databases are unavailable. By building the alternate databases from backup files each time you take a backup, you eliminate this step when an emergency situation occurs. Performance issues should determine whether you should build alternate databases after backups or only when recovery is necessary. Having alternates built after each backup gives you more options if a disaster occurs.

ACFSERVE BACKUP Command

Use this command to manually start a backup of one or all CA ACF2 for VM databases. The authority necessary to issue this command is defined in the AUTH field of the ACFSERVE BACKUP privilege record.

Recovery Utilities

There are several utilities to use during recovery. They are defined below.

ACFDBCPY

Copies CMS format databases from one reserved disk to another.

ACFDBRST

Restores a CMS or VSAM database from backup files.

ACFRECVR

Invokes the database recovery program after doing some setup work.

ACF4FRCV

Merges SMF records with a database to recover changes to that database.

For more information about these utilities, see the *Reports and Utilities Guide*.

The Backup Process

Ensure that the VM directory entry for the CA ACF2 for VM service machine includes MDISK entries for all the minidisks that contain the primary and alternate sets of databases.

Automatic backup processing is done as follows:

1. As the backup begins, CA ACF2 for VM sends all users defined in the NOTIFY operand a message indicating that it is taking a backup. It also notifies these users about the status of the backup, such as completion of step, completion of the entire backup, or abnormal termination.
2. CA ACF2 for VM tries to locate the service machine minidisk at the address specified in the MDISK operand of the BACKUP VMO record. This minidisk must be accessed read and write. If CA ACF2 for VM cannot find this minidisk or it is not accessed in read and write mode, CA ACF2 for VM writes the backup copies to the service machine 0191 disk. If the 0191 disk is not accessed in read and write mode, CA ACF2 for VM terminates with an error message.
3. CA ACF2 for VM copies each database and creates a sequential CMS file. The filename is determined through the @DDSN macro. The filetype is DATUT1. After it copies each database, it erases the previous backup copy and renames the DATUT1 file to BACKUPDB. This process ensures that any previous backup copy remains intact until the new copy is created. The minidisk containing the backup files must be large enough to contain all three files, plus a copy of the largest file.
4. The output data from the backup is in sequential format. You cannot use it as a database for the CA ACF2 for VM service machine. (CA ACF2 for VM supplies a facility for rebuilding databases. See the Post Backup Service Machine for more details.)

Placement of Databases, Backup Files, and SMF Minidisks

You must plan carefully to determine which real DASD devices the following groups of minidisks should be located on:

- SMF file minidisks and backup file minidisks
- Primary database minidisks
- Alternate database minidisks (optional).

Place these three groups of minidisks on separate physical disk packs and on separate DASD control units. Separating these groups of minidisks ensures at least one set of databases is available if a hardware error occurs or that the files necessary to rebuild the databases are available.

Sample Backup and Recovery Procedures

There are many ways to recover the databases. Each method has its own advantages and disadvantages. The following sections present a few approaches to database backup and recovery. We present a detailed description of each logical step in the recommended order for database recovery.

Complete Database Recovery

We recommend this approach for database recovery. When a problem occurs that makes the primary databases unavailable or unusable, IPL CA ACF2 for VM in NOAUTO mode (when CA ACF2 for VM is not active) for recovery. This approach enables complete database recovery, assuming all SMF files are available since the last backup was taken. A disadvantage is that only FORCEID users can access CA ACF2 for VM during recovery. Because database recovery is a rare occurrence, complete recovery in NOAUTO mode prevents users from accessing the system until recovery is completed. If you cannot afford the downtime to do complete recovery, review the other recovery methods presented in the following sections.

Recovery Steps

When the primary databases are unavailable because of hardware error or database corruption, follow these steps for complete recovery:

1. IPL the system without starting CA ACF2 for VM (NOAUTO mode).
2. Restore the latest backups to the alternate databases. Skip this step if you use the post backup service machine to restore backups to alternate databases after each backup occurs. Verify that the automatic restore occurred correctly.
3. Identify the SMF files needed for recovery.
4. Run the ACFRECVR utility to merge database changes from the SMF records to the alternate databases. For additional information on this utility, see the *Reports and Utilities Guide*.
5. IPL the system using the alternate databases.

When the primary databases are available and you have enough time, perform the following:

1. IPL the system without starting CA ACF2 for VM (NOAUTO mode).
2. If the databases are corrupted, make a DDR copy of the primary databases for debugging purposes.
3. Copy the alternate databases to the primary databases.
4. IPL the system normally. This causes CA ACF2 for VM to use the primary databases.

5. Manually back up the CA ACF2 for VM databases with the ACFSERVE BACKUP command.
6. Restore the backups just taken to rebuild the alternate databases (or let the post backup service machine do the rebuild).

Quick System Availability with Partial Recovery

This approach is an alternative if you cannot afford the downtime of a complete recovery. IPL CA ACF2 for VM immediately on the alternate databases and do recovery later, when there is enough time. The advantage is that the IPL makes the databases useable. The disadvantage is that complete recovery is not possible with this approach. SMF records that record changes to the database contain the complete record and have a date and time stamp indicating when the change occurred. Because the recovery utility does not update a record in the database that has a higher date and time stamp than an SMF record, CA ACF2 for VM does not apply SMF changes before the problem occurred to any record updated between the problem and the recovery. Because a database update is made when a user logs on, there is a large potential to lose data.

Recovery Steps

When the primary databases are unavailable due to hardware error or corrupted databases, follow these steps for quick, partial recovery:

1. Restore the latest backups to the alternate databases. (Skip this step if you use the post backup service machine to restore backups to the alternate databases after each backup occurs.) Verify that the automatic restore occurred correctly. If you use this approach, we recommend you use the post backup service machine to do the automatic restores.
2. IPL the system using the alternate databases.

When the primary databases are available and you have time, do the following:

1. IPL the system without starting CA ACF2 for VM (NOAUTO mode).
2. Identify the SMF files needed to recover. You must use current (online) SMF disks. They are volatile because an SMF disk can become active and reused for recording at any time unless you IPL CA ACF2 for VM in NOAUTO mode.
3. Run the ACFRECVR utility to merge database changes from the SMF records into the alternate databases. For additional information on this utility, see the *Reports and Utilities Guide*.
4. If the databases are corrupted, make a DDR copy of the primary databases for debugging.
5. Copy the alternate databases to the primary databases.
6. IPL the system normally. This causes CA ACF2 for VM to use the primary databases.

7. Manually back up the CA ACF2 for VM databases using the ACFSERVE BACKUP command.
8. Restore the backups just taken to rebuild your alternate databases (or you can use the post backup service machine to rebuild the alternate databases).

Quick Availability, Recovery of Previous Change

This approach is an alternative if you cannot afford downtime while complete recovery is done. It is a variation of the previous section, Quick System Availability With Partial Recovery. To use this method, IPL CA ACF2 for VM immediately on the alternate databases and recover later, when downtime is affordable or when it will not have a great impact. Instead of attempting complete recovery of all changes since the backups, which will not work for records updated after the problem, recovery only uses SMF records between the time of the backup used to build the alternate database and when the problem occurred.

This approach's advantage is that, after the problem occurs, only an IPL will activate operations. Also, because complete recovery cannot be done in this case, lost data is limited to changes done while on the alternate databases. You can notify users with a LOGMSG that any password or other changes to the CA ACF2 for VM databases are lost until further notice. The disadvantage is that you cannot recover all of the databases with this approach.

Recovery Steps

When the problem is a hardware error or corrupted databases, follow these steps for recovery:

1. Restore the latest backups to the alternate databases. Skip this step if you use the post backup service machine to restore backups to the alternate databases after each backup occurs. Verify that the automatic restore did occur correctly. We recommend that you use the post backup service machine to do the automatic restores.
2. Record the date and time to use later in selecting the SMF records for recovery.
3. Issue a LOGMSG (or equivalent) notifying users that changes to CA ACF2 for VM are lost until further notice.
4. IPL the system using the alternate databases.

When the primary databases are available and you have time, follow these steps:

1. IPL the system without starting CA ACF2 for VM (NOAUTO mode).
2. If the databases are corrupted, make a DDR copy of the primary databases for debugging purposes.
3. Restore the latest backups to the primary databases. This restore step is done to the PRIMARY databases, not the alternates.

4. Identify the SMF files needed for recovery. You must use current (online) SMF disks. They are volatile because a SMF disk can become active and reused for recording at any time unless the system has been IPLed in NOAUTO mode.
5. Run the ACFRECVR utility to merge database changes from the SMF records to the primary databases. For more information on this utility, see the *Reports and Utilities Guide*. This recover step applies to the PRIMARY databases, not the alternates. Use the recover options to limit the start and end range of dates and times to prevent the utility from using SMF records created after the problem occurred.
6. IPL CA ACF2 for VM normally (defaults to the primary databases).
7. Manually back up the CA ACF2 for VM databases using the ACFSERVE BACKUP command.
8. Restore the backups just taken to rebuild the alternate databases (or let the post backup service machine rebuild the alternate databases).

Detailed Backup and Recovery Steps

This section presents detailed information about the various backup and recovery steps presented in previous sections.

IPLing Without Starting CA ACF2 for VM

There can be situations (such as a catastrophic hardware error) when you must IPL VM without CA ACF2 for VM active. CA ACF2 for VM lets you IPL the system in force mode. When the system runs in force mode, the following restrictions exist:

- Only the users specified in the FORCEID and NOAUTOU lists (in the VMXAOPTS macro) can log onto VM.
- If NOAUTO=DIRPASS (NOAUTO is located in the VMXAOPTS macro), VM directory validation is done when the system is in force mode and FORCEID users must enter their valid VM directory passwords. If NOAUTO=NOPASS, only the user ID is checked to ensure it is a valid FORCEID.
- All CA ACF2 for VM security validation is bypassed. However, CA ACF2 for VM enforces minidisk password protection.

To IPL in force mode:

1. IPL VM according to your site's procedures. Watch for the following message:

   ```
   ACFpgm000R Enter ACF2 startup options or press Enter
   ```

   When it appears, enter the following command:

   ```
   NOAUTO
   ```

   Press Enter to prevent the service machine autolog. VM will be running in FORCE mode. No CA ACF2 for VM security validations will take place.

2. Any user defined in the FORCEID list can now log on without a password. (An exception is if NOAUTO=DIRPASS. FORCEID users must enter their valid VM directory passwords.)

3. Repair the problem as soon as possible. CA ACF2 for VM does no validations while it is in FORCE mode.

4. After the problem is resolved, IPL the system normally autologging the CA ACF2 for VM service machine.

Restoring Backup Files to Usable Databases

Use the ACFDBRST utility to restore backup files to usable databases. You can use the sample exec, ACFREST, to help you run this utility. Use a FILEDEF to define the file ID of the backup file. Use a FILEDEF or a DLBL (for VSAM) to define the database to restore into. The utility expects a parameter of LID, RULE, INFO, or ALL to determine which databases to process. Also, the minidisks you are restoring must be reserved disks and in 4K blocks. If they are existing database disks, this is already done.

Below is an example of running this utility:

```
FILEDEF BKPLIDS DISK BKLID BACKUPDB Z
FILEDEF LIDS DISK LID DATABASE U
FILEDEF BKPRULES DISK BKRULES BACKUPDB Z
FILEDEF RULES DISK RULES DATABASE V
FILEDEF BCPINFO DISK BKINFO BACKUPDB Z
FILEDEF INFO DISK INFO DATABASE W
ACFDBRST ALL
```

In this example, the backup files disk is accessed as Z. The Logonid, Rule, and Infostorage databases reserved disks are accessed as U, V, and W, respectively. After issuing the FILEDEF statements, enter the ACFDBRST ALL command to restore all three database files from backups. You must restore all three databases with this utility. For detailed information about the ACFDBRST utility, see the *Reports and Utilities Guid*e.

The ACFRECVR Utility

This utility merges SMF records with the databases. The next section explains the SMF files needed for recovery.

SMF Files Necessary for Recovery

The SMF files necessary for recovery depend on which approach to recovery you take. Choose the SMF files dated in the date and time interval needed for the desired approach, based on the filetype of the SMF files. The filetype has the format yydddsss.

yy

The year

ddd

The julian date

sss

A sequential suffix from 1 to FFF, indicating the number of the SMF file for the day.

Running the ACFRECVR Utility

To run the ACFRECVR utility, issue the FILEDEFs for each database to recover or put the FILEDEF statements in the ACFRECVR EXEC.

For example:

```
FILEDEF LIDS DISK LID DATABASE U
FILEDEF RULES DISK RULES DATABASE V
FILEDEF INFO DISK INFO DATABASE W
```

Invoke the ACFRECVR EXEC and respond to SMF prompts. For more information on this exec, see the *Reports and Utilities Guide*.

Starting CA ACF2 for VM Using Alternate Databases

The following procedure assumes that at least one set of alternate databases is available. See the Alternate CA ACF2 for VM Database section for more information about the alternate databases.

IPL VM. Specify the name of an alternate set of databases in response to the service machine startup options message.

```
ACFpgm000R Enter ACF2 startup options or press ENTER,

ddsn(altcms) nobackup
```

The user reply DDSN(ALTCMS) NOBACKUP directs CA ACF2 for VM to use the filenames defined in the @DDSN macro named ALTCMS. The NOBACKUP parameter deactivates the CA ACF2 for VM automatic backup facility, preventing automatic backups from writing over backup files. This procedure is completed. CA ACF2 for VM is now running with the alternate databases active.

Copying CA ACF2 for VM Databases

Use the ACFDBCPY utility to copy a CMS format CA ACF2 for VM database. You cannot use COPYFILE because the databases are on reserved disks. Use the ACFDBCPY utility to copy CMS databases for recovery purposes, such as copying alternate databases to primary databases or expanding the size of the minidisk a database is on. To run the ACFDBCPY utility, enter the following:

```
ACFDBCPY ALTLID DATABASE Z LID DATABASE U.
```

If you are using VSAM databases, use the VSAM REPRO utility to copy the databases as follows:

1. REPRO the VSAM databases to sequential files
2. Delete and define the target VSAM clusters
3. REPRO the sequential files to the target clusters just defined.

For more information about the ACFDBCPY utility, see the *Reports and Utilities Guide.*

Taking a Manual Backup

There can be a situation when you want to manually back up the CA ACF2 for VM databases. Use the ACFSERVE command to take manual backups. To back up all the databases, issue the ACFSERVE BACKUP command. You can back up any or all CA ACF2 for VM databases manually. By default, CA ACF2 for VM uses the names in the @DDSN macro, specified in the DDSNID= operand of the BACKUP VMO record. You can specify the name of the @DDSN macro for the filenames you are using to override this default.

To back up all databases using the filenames from the BKWEEKLY @DDSN macro, issue the ACFSERVE BACKUP BKWEEKLY command. To back up only the Logonid database using the filenames from the BKWEEKLY @DDSN macro, issue the ACFSERVE BACKUP BKWEEKLY LID command.

After the ACFSERVE BACKUP command successfully completes, the post backup service machine is autologged if you specified one in the AUTOLOG operand of the BACKUP VMO record.

Disabling the Automatic Backup Facility

You can disable the CA ACF2 for VM automatic backup facility if necessary. CA ACF2 for VM continues functioning normally, even if the alternate databases are not available when automatic backup is initiated. However, there would be a series of error messages to indicate that the backup failed.

The procedure for disabling the automatic backup facility is shown below. After the facility is disabled, you cannot activate it again until you IPL CA ACF2 for VM. To deactivate the CA ACF2 for VM automatic backup facility

1. IPL VM according to your site's procedures.
2. Watch for the following message:

   ```
   ACFpgm000R Enter ACF2 startup options or press Enter
   ```

3. When it appears, enter this command:

   ```
   NOBACKUP
   ```

   Press Enter to disable the backup facility.

   ```
   ACFpgm000R Enter ACF2 startup options or press Enter

   nobackup
   ```

   VM is now running normally. However, no automatic backup of the databases is taken. You can still use the ACFSERVE command to take manual backups. See the Taking a Manual Backup section for information about doing manual backups.

Post Backup Service Machine

Use the post backup service machine to build copies of the database that you can run after a backup.

Post Backup Service Machine Execs

CA ACF2 for VM supplies three sample execs for configuring the post backup service machine and rebuilding the databases:

ACF2PBSM SAMPEXEC

This exec is the CA ACF2 for VM Post Backup Service Machine SAMPEXEC. It initiates the database rebuild process. You have to customize it to your local environment and rename it PROFILE EXEC. Then place it on the post backup service machine 191 minidisk.

This exec processes the program stack and initiates the RESTCMS or RESTVSAM EXECs. When you customize the exec, you should ensure that the appropriate exec is invoked depending on whether you are rebuilding CMS or VSAM CA ACF2 for VM databases.

CA ACF2 for VM autologs the post backup service machine at the end of the backup. You can either schedule an automatic backup or manually initiate a backup with the ACFSERVE BACKUP command. A scheduled backup backs up all three CA ACF2 for VM databases of the active database set. A manual backup backs up one database or a combination of databases in any set defined in the @DDSN macro in the ACFFDR.

When the post backup service machine is autologged, the program stack contains information about which databases were backed up, what database set they belonged to, and what VM user ID initiated the backup if the ACFSERVE command invoked it. The program stack has a line of data in it in the following form:

```
LIDOPT RULEOPT INFOOPT invoker DDSN ddsnname.
```

The variables for the data are described below.

LIDOPT

LIDDB if the LID file was backed up.

NOLID if the LID file was not backed up.

RULEOPT

RULEDB if the RULE file was backed up.

NORULE if the RULE file was not backed up.

INFOOPT

INFODB if the INFO file was backed up.

NOINFO if the INFO file was not backed up.

invoker

SYSTEM, if the backup was automatic.

USERID that issued the command, if the backup was manually generated.

DDSN

The keyword indicating the ACFSERVE BACKUP command was invoked with an alternate DDSN name (optional).

ddsnname

The name the user specified when issuing the ACFSERVE command (optional).

RESTCMS SAMPEXEC

Sample exec for restoring alternate CMS databases. This exec rebuilds copies of the databases.

RESTVSAM SAMPEXEC

Sample exec for restoring alternate VSAM databases, and also rebuilds copies of the CA ACF2 for VM databases. This exec, in turn, calls the following sample execs:

- DDEFLIDS SAMPEXEC
- DDEFRULE SAMPEXEC
- DDEFINFO SAMPEXEC

These sample execs perform the IDCAMS DELETE and DEFINE functions for each VSAM database. There are also three AMSERV control statement files, one each of the above DDEFxxx sample execs uses:

- DDEFLIDS SAMPAMS
- DDEFRULE SAMPAMS
- DDEFINFO SAMPAMS

You must rename all of the above sample exec files to have a filetype of EXEC. You must rename all of the SAMPAMS files to an AMSERV filetype. Modify VOL IDs, data set names, minidisk addresses, and VSAM extents to tailor each of these files.

For more information on the post backup service machine, see Installation Guide.

Chapter 28: OpenExtensions VM Support

Security is a major concern in environments where users move across hardware platforms and operating systems to access numerous applications. Sites want and need the same control over, and accountability for, data and resources accessed in an open system as they are used to having in their mainframe environment. CA ACF2 for VM offers security for such open environments by supporting OpenExtensions VM and the standards developed for a Portable Operating System Interface (POSIX). Specifically, CA ACF2 for VM supports the following services in an OpenExtensions VM environment:

- Byte File System (BFS)
- New User (UID) and Groupid (GID) definitions.

This chapter explains:

- Controlling access to OpenExtensions VM
- Controlling access to the Byte File System
- CA ACF2 for VM records for OpenExtensions VM.

This section contains the following topics:

Controlling Access to OpenExtensions VM

When a user tries to enter the OpenExtensions VM shell, CA ACF2 for VM verifies that he is an OpenExtensions VM user before the system initializes the shell. CA ACF2 for VM also verifies that the user associated with a program trying to access OpenExtensions VM resources is an OpenExtensions VM user before allowing the access to the requested resource.

To define a user as an OpenExtensions VM user, you must:

- Define the user to CA ACF2 for VM.
- Assign the user to a group.
- Assign an OpenExtensions VM UID to the user.
- Assign an OpenExtensions VM GID to the group.

Defining OpenExtensions VM Users

OpenExtensions VM recognizes users by their assigned user identification (UID) and group identification (GID) numbers. (The UID referenced in relation to OpenExtensions VM is not the CA ACF2 for VM User Identification String (UID)). UIDs and GIDs can have numeric values of zero to 2,147,483,647. To set the UID for a user, create a USER Profile record. The new OEVM segment of the USER Profile record defines a user's UID, the user's home directory, the initial program that the user will run, and the initial Byte File System for the user. The initial program is generally the shell program that the user invokes.

For more information about creating USER Profile records, see CA ACF2 for VM Records for OpenExtensions VM.

Superusers

A superuser is a special user under OpenExtensions VM. The superuser is a trusted user who can maintain the OpenExtensions VM system and administer security in the BFS. A superuser's UID has a zero. Use caution when assigning the superuser authority. A superuser passes all security checks, meaning the superuser can access any file in the file system. This type of authority is similar to an unscoped security officer. However, assigning superuser authority to a user does not give the user any added authority outside of OpenExtensions VM.

Defining OpenExtensions VM Groups

OpenExtensions security is based on user and group ownership of files and processes. CA ACF2 for VM uses the GROUP field of the logonid to assign the user to an OpenExtensions VM group. Users can also specify the group they want to be associated with at system entry time. A new GROUP Profile record assigns the GID to the group. For more information on GROUP Profile records, see CA ACF2 for VM Records for OpenExtensions VM.

We recommend that you assign a unique GID to each group. If you assign the same GID value to multiple groups, the groups share ownership of and access to the same files. This can cause unreliable results. For example, if you assign multiple groups the same GID and a Query Group Database service request is made, only one group name is returned in response to the request. CA ACF2 for VM searches its cross-reference tables and returns the first group that matches the GID. It does not return all the groups associated with that GID, nor can it distinguish the specific group that you intend to make the request for.

Supplemental Groups

Under OpenExtensions VM and CA ACF2 for VM, a user is a member of the group defined in the GROUP field of his logonid and a member of any group that he has access to through a resource rule. A user can change to another group with the OpenExtensions VM setgid() or setegid() functions as with the shell newgrp command. These groups are called supplemental groups.

To grant a user access to a group other than the group defined in his logonid record, create a TYPE(PGR) resource rule. The $KEY of the rule identifies the one- to eight-character group name. This field is maskable, but you must create an entry for the PGR type code in the RESTYPE VMO record. Following is an example of a resource rule that grants access to group OEVMGRP to users in the payroll and audit departments:

```
$KEY(OEVMGRP) TYPE(PGR)
 UID(*****pay-) ALLOW
 UID(*****aud-) ALLOW
```

For more information about defining TYPE(PGR) resource rules, see Recource(PGR) POSIX Group Rules.

For more information about setting the owner, group, and other permissions for a file, see the OpenExtensions for VM *User's Guid*e.

Controlling Access to the Byte File System

The Byte File System (BFS) is a tree-structured file system consisting of directories and files. It resembles the DOS file system, although the direction of the slashes is reversed from DOS.

Security for the file system directories and files is based on a UNIX model of security. Each file and directory is assigned an owning UID and an owning GID. This assignment is defined and saved in the file system, not in the external security product.

Three categories of users can access each directory and file in the BFS. They are:

- The file owner
- The group that owns the file
- All other users defined to OpenExtensions VM.

Different access levels can be set for any of these three categories. For example, permissions can be defined so that the file owner gets READ and WRITE access, a member of the file&rs.s group gets only READ access, and all other users can neither READ nor WRITE to the file.

Under CA ACF2 for VM, you must define a UID for each OpenExtensions VM user and a GID for every group used to access OpenExtensions VM. You must also assign a default group in all OpenExtensions VM user ids and give these users access to any supplemental groups needed.

For more information about the Byte File System and setting file permissions, see the IBM documents entitled *OpenExtensions for VM* User*'s Guid*e and *VM CMS File Pool Planning, Administration, and Operation.*

CA ACF2 for VM Records for OpenExtensions VM

Superuser Administrator Logonid

The OpenExtensions VM Shell and Utilities installation process creates directories in the Byte File System. To perform the installation steps, the user must have superuser authority.

To create a Superuser Administrator logonid and give it the authority it needs, follow these directions:

1. Define the logonid as a superuser by issuing the following CA ACF2 for VM subcommands:

   ```
   SET PROFILE(USER) DIV(OEVM)

   INSERT SYSPROG1 UID(0)
   ```

   Logonid SYSPROG1 is defined as a superuser by setting the UID value to zero.

2. Define the logonid as a member of a group by issuing these CA ACF2 for VM subcommands:

   ```
   SET LID

   CHANGE SYSPROG1 GROUP(SYSPROG)
   ```

   The example shows logonid SYSPROG1 changed so that this user can signon and be validated as a member of group SYSPROG. The members of group SYSPROG are a special subset of users who perform system-related tasks.

3. Assign the group a GID value by issuing these CA ACF2 for VM subcommands:

   ```
   SET PROFILE(GROUP) DIV(OEVM)

   INSERT SYSPROG GID(20)
   ```

   In this example, the SYSPROG group is assigned a GID of 20.

4. Rebuild the USER and GROUP Profile directories as documented under the section entitled “System Commands for OpenExtensions VM” later in this chapter.

USER Profile Records

OpenExtensions VM UIDs are defined to CA ACF2 for VM by new USER Profile records in the CA ACF2 for VM Infostorage database. Specifically, you define the UID information in the OEVM segment of this Profile record. The OEVM segment of the USER Profile record contains four fields: UID, HOME, FILESYS, and PROGRAM.

UID is a numeric field that accepts values from zero to 2,147,483,647. A UID defined with a value of zero indicates that this user is a superuser. For a definition of the term superuser, see the *OpenExtensions for VM User's Guide.* This field does not have to be unique, but we recommend that you make it unique; otherwise, individual accountability and control are lost. **This field is required.**

The HOME field defines the initial directory pathname. This is the initial directory used when a user enters the OPENVM command. The HOME field accepts from one to 1023 characters. Both upper and lower case characters are allowed. If HOME is not defined, OpenExtensions VM sets the initial directory for the user to the root directory. This field is optional.

The FILESYS field defines the Byte File System that owns the user's initial directory. The FILESYS field accepts 1 to 1,023 characters. Both upper and lower case characters are allowed. If you do not define FILESYS, OpenExtensions VM requires the user to issue the OPENVM MOUNT command to mount a Byte File System. This field is optional.

The PROGRAM field defines the user's OpenExtensions VM shell program, which is the first program started when the OPENVM command is entered. The PROGRAM field accepts from one to 1023 characters. Both upper and lower case characters are allowed. If PROGRAM is not entered, OpenExtensions VM gives control to the default shell program. This field is optional.

The maximum length of the HOME, PROGRAM, and FILESYS fields is 1,023 bytes. The CMS command input area is only 140 characters. To enter more than 140 characters into a record, you can use the standard CA ACF2 for VM continuation character “-” (blank, dash) between fields and the UNIX style continuation inside a field. The UNIX style continuation closes a field when a matched parenthesis is found. For example, if you are defining the HOME field, you start with **HOME(/../VMBFS** and you are at the end of the CMS input area, you just press Enter. The ACF command knows that it has only found an open parenthesis and posts a VM READ to let you finish the field. You can enter as many characters as needed for the field, as long as you do not exceed the field maximum length. CA ACF2 for VM terminates the field when it finds the close parentheses character **fpoolnam:path/)**.

If you do not define a User profile record in the OEVM segment, the users default to a UID of -1 and the user name DEFAULT when they invoke any POSIX functions.

The example below shows how to define user OEVMUSR as a superuser. Since HOME, FILESYS, and PROGRAM are not explicitly specified, the defaults are taken for these fields.

```
SET PROFILE(USER) DIV(OEVM)

INSERT OEVMUSR UID(0)
```

This example shows how to define user OEVMU2 as a regular user. The HOME, FILESYS, and PROGRAM fields are defined.

```
SET PROFILE(USER) DIV(OEVM)

INSERT OEVMU2 UID(199) HOME(/u/oevmu2) PROGRAM(/bin/sh) -
 FILESYS(/../VMBFS:fpoolname.ROOT/)
```

The following five profile records are system IDs for POSIX. You must insert them into the CA ACF2 for VM database.

```
SET PROFILE(USER) DIV(OEVM)

 INSERT ROOT UID(0)
 INSERT DAEMON UID(1)
 INSERT BIN UID(2)
 INSERT SYS UID(3)
 INSERT ADM UID(4)
```

You must issue the ACFSERVE RELOAD PROFILE USER subcommand after inserting or changing any USER Profile records to rebuild the USER Profile directory. See the "Using the ACFSERVE Commands" chapter for syntax and additional information on ACFSERVE commands.

GROUP Profile Records

OpenExtensions VM groups are defined to CA ACF2 for VM by new GROUP Profile records in the CA ACF2 for VM Infostorage database. The new GROUP Profile record contains the OEVM segment which consists of one field: the GID field.

GID is a numeric field that accepts values from zero to 2,147,483,647. This value does not need to be unique, but we recommend that you make the GID unique; otherwise, control over a particular group is lost.

This example shows how to insert an OEVM GROUP Profile record for a group called OEVMGRP and assign it a GID of 20.

```
SET PROFILE(GROUP) DIV(OEVM)

INSERT OEVMGRP GID(20)
```

Assigning Users to Groups Under CA ACF2 for VM

You assign a user's default group by setting the GROUP field in that user's CA ACF2 for VM logonid. Each user's primary group name is defined in the GROUP field of the LID record. If you do not define a GROUP name, it defaults to DEFAULT and the GID defaults to -1 when the user invokes any POSIX functions.

The following example shows you how to assign logonid OEVMU2 to group OEVMGRP:

```
SET LID

CHANGE OEVMU2 GROUP(OEVMGRP)
```

The following seven profile records are the standard POSIX groups. You must insert them into the CA ACF2 for VM database.

```
SET PROFILE(GROUP) DIV(OEVM)

 INSERT SYSTEM GID(0)
 INSERT STAFF GID(1)
 INSERT BIN GID(2)
 INSERT SYS GID(3)
 INSERT ADM GID(4)
 INSERT MAIL GID(6)
 INSERT SECURITY GID(7)
```

You must issue the ACFSERVE RELOAD **PROFILE GROUP subcommand after inserting or changing any GROUP Profile records to rebuild the GROUP Profile directory.** See the "Using the ACFSERVE Comands" chapter for syntax and additional information on ACFSERVE commands. The user must logon to the system after this directory is rebuilt. Otherwise,
CA ACF2 for VM does not recognize the change and denies access.

Primary Logon Groups

A user can belong to more than one group. The user's primary group is set in their LIDREC in the GROUP field. You can identify all other groups that a user can access (**setgid()|setegid()**) through resource rules. Each user **must** be permitted to their primary group through resource rules. See the "About Resource Rules" chapter for more information about resource rules.

The following example sets user MAINT's primary group to SYSTEM and inserts other required POSIX logonids:

```
SET LID

 CHANGE MAINT GROUP(SYSTEM)
 INSERT ROOT GROUP(SYSTEM) PASSWORD(password)
 INSERT DAEMON GROUP(STAFF) PASSWORD(password)
 INSERT BIN GROUP(BIN) PASSWORD(password)
 INSERT SYS GROUP(SYS) PASSWORD(password)
 INSERT ADM GROUP(ADM) PASSWORD(password)
```

Resource(PGR) POSIX Group Rules

Users are allowed to belong to multiple groups in the OpenExtensions VM environment. A user's primary or default group is defined in the GROUP field of his logonid. A user can change to another group with the OpenExtensions VM setgid() or setegid() functions or with the shell **newgrp** command. These additional groups are called supplemental groups.

To allow a user to use his primary group and groups other than the default group, you must create POSIX Supplemental Group resource rules for the group. The default resource type of POSIX Group rules is PGR.

The POSIXGRP field of the RESCLASS VMO record defines the type code for the POSIX group rules. You can modify this RESCLASS field.

The following example gives the previously mentioned users access to their primary groups:

```
COMPILE
 $KEY(ROOT) TYPE(PGR)
 UID(MAINT) ALLOW
 UID(ROOT) ALLOW
 $KEY(STAFF) TYPE(PGR)
 UID(DAEMON) ALLOW
 $KEY(BIN) TYPE(PGR)
 UID(BIN) ALLOW
 $KEY(SYS) TYPE(PGR)
 UID(SYS) ALLOW
 $KEY(ADM) TYPE(PGR)
 UID(ADM) ALLOW
```

Control(VMO) OPTS Record

In a VM environment, you can maintain the OpenExtensions VM Database information in the VM directory or with an External Security Manager (ESM) such as CA ACF2 for VM. The POSIXDB option in the OPTS VMO record identifies whether CA ACF2 for VM activates OpenExtensions VM Database support. See the “Defining Structured Infostorage Records” chapter for the syntax and more information on the OPTS VMO record.

If you specify POSIXDB, CA ACF2 for VM informs VM that it is maintaining the OpenExtensions VM Databases. If this option is on, CA ACF2 for VM responds to the DIAGA0 subcode X'08' and takes over OpenExtensions VM Database management. To activate CA ACF2 for VM OpenExtensions VM Database support, you must have the appropriate PROFILE records written, the OPTS VMO POSIXDB option set, and you must IPL VM.

If you specify NOPOSIXDB (the default), CA ACF2 for VM stays out of the POSIXDB business and responds negatively to the DIAGA0 subcode X'08'.

MAXPGRPS tells CP what the maximum number of POSIX groups you are allowing. Be conservative in your estimate because CP allocates storage based on this number for every user that logs on to VM. MAXPGRPS has a minimum value of 32 and a maximum value of 125. The following example illustrates how to set POSIXDB and MAXPGRPS:

```
SET CONTROL(VMO)
 CHANGE OPTS POSIXDB MAXPGRPS(32)
```

Control(VMO) RESCLASS Record

The type code for POSIX group rules is defined in the RESCLASS VMO record. The default is PGR. Be sure that the appropriate type code is set. If not, you can use the following command to change it:

```
SET CONTROL(VMO)
 CHANGE RESCLASS POSIXGRP(PGR) REP
```

Control(VMO) RESTYPE Record

You must define POSIX Supplemental Group resource rules in the resident resource rule cache. The resident resource cache can have several resource types in it. The Supplemental Group resource type is defined in the TYPES field of the RESTYPE VMO record. See the “Defining Structured Infostorage Records” chapter for syntax information.

Use the ACF command subcommands as shown below to add the PGR resource type to the resident cache:

```
SET CONTROL(VMO)
 CHANGE RESTYPE TYPE(PGR) ADD
```

CA ACF2 for VM Diagnose Limiting

CP allows users to inquire on the contents of the OpenExtensions VM Database through new CP diagnose calls. The new OpenExtensions VM diagnose calls extract OpenExtensions VM Database and runtime information or issue setid() requests. See the *Command and Diagnose Limiting Guid*e for more information on these diagnose calls.

Chapter 29: Understanding SAF

The IBM System Authorization Facility (SAF) provides an interface that can direct control to all external security products, including CA ACF2 for VM. These developments make CA ACF2 for VM compliant with IBM SAF. CA ACF2 for VM processes all SAF calls by default. In the VM implementation of SAF, a service machine that is running a product that performs SAF calls invokes a router module called RPIUCMS. The RPIUCMS module used by CA ACF2 for VM is supplied by the CA-ESM Release 1.1 or above component of CA-CIS.

This chapter discusses the following topics:

- How CA ACF2 for VM uses SAF and RPIUCMS
- Components of the CA ACF2 for VM SAF interface
- Translating resource classes (CLASMAP)
- Defining environments for SAF calls (SAFDEF)
- Solving problems with SAF calls
- Data set loggings or violations
- Resource loggings or violations
- How CA ACF2 for VM processes SAF calls

This section contains the following topics:

How CA ACF2 for VM Uses SAF and the RPIUCMS Module

CA ACF2 for VM uses the CA SAF interface in the CA-ESM RPIUCMS module. When a system product invokes the RPIUCMS module to request the services of an external security product, CA ACF2 for VM gets control. When a security event occurs, CA ACF2 for VM intercepts the call or processes a SAF call.

When another system product makes a request for security information, it uses the RACROUTE macro. The CA SAF interface intercepts these requests and processes them in terms that CA ACF2 for VM can understand. Some common requests with their CA ACF2 for VM translations are:

RACROUTE REQUEST	CA ACF2 for VM Translation
AUDIT	CA ACF2 for VM journals a Type=V SMF record for the specified audit event. This results in type TRC entries in the ACFRPTRV report. No SVC calls can be issued from this environment. For example, VTAM, APPC, and PSF make use of these types of calls.
AUTH, CLASS=DATASET	CA ACF2 for VM performs data set validation for the request. You may need to define a SAFDEF record for the call.
AUTH, CLASS=others	CA ACF2 for VM performs a resource validation. The default CLASS is SAF. If another CLASS is specified in the RACROUTE macro, you must create a CLASMAP record to define the three-character resource type of the resource that you want to validate.
DEFINE, CLASS=DATASET	CA ACF2 for VM performs data set validation for the request.

RACROUTE REQUEST	CA ACF2 for VM Translation
EXTRACT	CA ACF2 for VM executes the SAF call to extract the requested information from the CA ACF2 for VM databases, where applicable in CA ACF2 for VM. Standard SAF and SAF product return and reason codes are returned with some exceptions. If a RACROUTE REQUEST=EXTRACT,TYPE=REPLACE SAF request fails, CA ACF2 for VM may return a special SAF product reason code if the failure was due to a non-zero return code from the CA ACF2 for VM TYPE=A SVC call issued to effect the database update. In this instance, the SAF return code is 4, the SAF product return code is 16 (decimal), and the SAF product reason code is the low three numbers of the corresponding ACF00nnn message ID associated with the error. This helps you identify the reason for the update failure.
FASTAUTH	CA ACF2 for VM FASTAUTH processing retrieves the rule in storage if one exists and performs a resource validation. The validation takes into consideration both NEXTKEY and XREF processing. If access is allowed, CA ACF2 for VM sets an allow return code. If access is denied or no rule exists, CA ACF2 for VM checks for unscoped SECURITY or NON-CNCL. If these privileges are on, CA ACF2 for VM sets an “allow but log” return code. The caller is responsible for redriving the validation as a regular AUTH call. CA ACF2 for VM performs a FASTAUTH call only if resident rules exist. If the rules are not resident, the call gets a RC=8. See Part I: Defining VM System Options for information about the RESTYPE record and how to activate an infostorage record. If you have not made the rules resident, the FASTAUTH call creates a violation.
LIST	CA ACF2 for VM builds a resource rule directory for the specified resource class. The default CLASS is SAF. This type code must be specified in a RESTYPE record. If the SAF call is for another CLASS, you must specify a CLASMAP record to translate the resource class into a three-character resource type. CA ACF2 for VM stores the resource type in the resource rule directory.
STAT	CA ACF2 for VM verifies that security is active, that the class exists, and that the class is active. CA ACF2 for VM processes all SAF classes as ACTIVE.
TOKENBLD TOKENMAP TOKENXTR	CA ACF2 for VM TOKEN processing routines build, map, or extract TOKENs.

RACROUTE REQUEST	CA ACF2 for VM Translation
VERIFY	CA ACF2 for VM performs all VERIFY requests to build the user control block (ACEE and ACUCB) for the service machine. CA ACF2 for VM performs system entry validation on the logonid associated with the service machine. VERIFY requests are also supported for MUSASS environments.
VERIFYX	CA ACF2 for VM performs all VERIFYX requests to validate a user, build a TOKEN, and return it to the caller.

Components of the CA ACF2 for VM SAF Interface

To understand how CA ACF2 for VM processes SAF calls, you must be familiar with these components of the CA ACF2 for VM SAF interface:

- CLASMAP record
- SAFDEF record
- ACFSERVE SECTRACE command
- CA ACF2 for VM access and resource rules

Since CA ACF2 for VM is SAF compliant, the CA ACF2 for VM SAF interface is always active. Even though CA ACF2 for VM processes all SAF calls by default, you can decide whether you want CA ACF2 for VM to validate particular SAF calls.

This section explains the function of each component. Individual components are explained in detail later in this chapter with examples that you can tailor to the unique calls you must define and validate.

CLASMAP Record

The VMO CLASMAP record translates an eight-character SAF resource class into a three-byte CA ACF2 for VM resource type code. CA ACF2 for VM checks the CLASMAP record for this type code for all SAF calls.

To see the general CLASMAP record of RESOURCE(********) RSRCTYPE(SAF) defined by CA ACF2 for VM, issue the following command:

```
SHOW CLASMAP
```

CA ACF2 for VM searches CLASMAP records in the following manner to determine the type code for a resource call:

- When a SAF general CLASMAP record exists, CA ACF2 for VM searches the external (site-defined) CLASMAP records. If the general CLASMAP record is the only match, CA ACF2 for VM searches the internal CLASMAP records provided with CA ACF2 for VM. CA ACF2 for VM matches on the most specific CLASMAP regardless of it being an internal or external CLASMAP record. If no match is found, CA ACF2 for VM uses the general type code of SAF.
- When no SAF general CLASMAP exists, CA ACF2 for VM searches the external (site-defined) CLASMAP records. If no match is found,
  CA ACF2 for VM searches the internal CLASMAP records provided with CA ACF2 for VM. If no match is found, CA ACF2 for VM uses the first three characters of the resource name as the type code.

SAFDEF Record

The VMO SAFDEF record identifies a SAF request and its environment to CA ACF2 for VM. CA ACF2 for VM provides its own SAFDEF records for internal functions. Your site may have to provide SAFDEF records for IBM program products and any other system or application products that make SAF calls. CA ACF2 for VM defines general SAFDEF records to process SAF calls. These internal SAFDEF records provide CA ACF2 for VM protection by default. CA ACF2 for VM searches for a SAFDEF record that defines the specific request and the action you want CA ACF2 for VM to take when it validates the request.

Note: There are SAFDEF restrictions with FASTAUTH processing. FASTAUTH does not allow the use of ENTITY on the RACROUTE field of the SAFDEF.

ACFSERVE SECTRACE Command

The ACFSERVE SECTRACE command is a diagnostic tool that enables you to capture, format, and display the RACROUTE parameter list passed by requests for SAF services. For complete details and command syntax, See Tracing SAF events in the “Using the ACFSERVE Commands” chapter.

CA ACF2 for VM Access and Resource Rules

Access and resource rules enable you to control how CA ACF2 for VM validates access to data sets and resources based on SAF requests. These requests can be made by other program products or system services.

Translating Resource Classes (CLASMAP)

The CLASMAP record translates eight-character resource classes into three-byte CA ACF2 for VM resource type codes. CLASMAP records are not required, but CA ACF2 for VM checks the CLASMAP record for the type code for all SAF calls. When no matching CLASMAP record is found during validation,
CA ACF2 for VM uses the first three characters of the resource class as the resource type. The three-character resource type code can enable you to write specific resource rules to validate security calls for a specified class.

A description of the CLASMAP record format and fields follows:

Record ID	Fields
CLASMAPqual	RESOURCE(class) RSRCTYPE(typecode) MUSID(musassid|********) ENTITYLN(0|entitylength)

Fields

RESOURCE(class)

Specifies the explicit eight-character resource class from the CLASS keyword on the RACROUTE macro. Standard CA ACF2 for VM resource name masking conventions apply.

RSRCTYPE(typecode)

Specifies the explicit three-character resource type code associated with the class. If you define a RESOURCE but do not define a RSRCTYPE, CA ACF2 for VM uses the first three characters of the RESOURCE as the RSRCTYPE. Use this type code to write resource rules to perform validation. This value cannot be masked. If you want to mask the name of the resource in your resource rule key, add this type code to the VMO RESTYPE record. For more information, see the “About Resource Rules” chapter.

MUSID(musassid******)**

Identifies the MUSASS to which the CLASMAP record applies. This lets several MUSASSes that share the same resource class use different type codes. Standard CA ACF2 for VM resource name masking conventions apply.

ENTITYLN(0|entitylength)

Specifies the entity length of the specified SAF class. The default is 0. Zero causes CA ACF2 for VM to search its internal CLASMAP definitions; non-zero causes the VMO CLASMAP to be used. The resultant CLASMAP record, VMO or internal, is used for RSRCTYPE and ENTITYLN. If the resultant ENTITYLN is zero, CA ACF2 for VM assigns a length of 39, the IBM default.

Creating Multiple CLASMAP Records

To create multiple CLASMAP records, append a qualifier to the record name in the format CLASMAPqual to generate a unique record ID (for example, CLASMAPVMAN or CLASMAP.DATASET). The total recid length is 16 bytes. The optional qualifier can be up to nine characters and must immediately follow the characters CLASMAP. If you use a period (.) as part of the qualifier string for the record name, CA ACF2 for VM counts it as one of the nine characters.

Viewing Internal and External CLASMAP Records

You can view the internal (CA ACF2 for VM-defined) and external (site-defined) CLASMAP records by issuing the SHOW CLASMAP subcommand.

```
show clasmap
 -- INTERNAL CLASMAP DEFINITIONS --
 MUSASS      RESOURCE   TYPE   ENTITY
  ID         CLASS      CODE   LENGTH
 ======      ========   ====   ======
 CICS        FILE       CFC      8
 CICS        PROGRAM    CPC      8
 CICS        TRANS      CKC      4
 CICS        TRANDATA   CTD      8
 CICS        TEMPSTRG   CTS      8
 CICS        DL/I       CPB      8
 -           PROGRAM    PGM      8
 -           UNVRPRT    UNR      0
 -           UNVPGM     UNP      8
 -           ACAPPL     ACA      0
 -           ACDIALOG   ACD      0
 -           DIRECTRY   SAF    153
 -           FILE       SAF    171
 -           SFSCMD     SAF    171
 -- EXTERNAL CLASMAP DEFINITIONS --
 MUSASS      RESOURCE   TYPE  ENTITY
  ID         CLASS      CODE  LENGTH
 ========    ========   ====  ======
 ********    TSTPROD1   TPI      0
```

Validating SAF RACROUTE Calls

You must specify a CLASMAP record for the following type of SAF RACROUTE call that you want to validate.

```
REQUEST=AUTH,CLASS=DATASET|others
```

AUTH calls with a CLASS specification of DATASET result in a data set validation. AUTH calls with any other CLASS specified result in a resource validation.

For more information, see Part I: Defining VM System Options.

Defining Environments for SAF Calls (SAFDEF)

The SAFDEF record defines the SAF environment and how you want CA ACF2 for VM to process the SAF call. CA ACF2 for VM provides internal SAFDEFs for SAF default protection. To see a list of the SAFDEF records active on your system, see the Viewing SAFDEF Records.

CA ACF2 for VM processes all SAF calls by default. To override the default SAF processing for a specific security event, you can specify a SAFDEF record. We provide examples of how to override SAFDEF records in Solving Problems with SAF Calls later in this chapter. CA ACF2 for VM performs validation based on the environment you define in this record. SAFDEF records are processed in a specific sorted order based on fields in the record, whether it is an external or internal SAFDEF, and whether the SUBSYS is defined as ACF2 or VMO. The sort criteria is as follows:

1. Record key
2. JOBNAME (Service machine name)
3. USERID
4. PROGRAM (Not active for VM sites)
5. RB (Not active for VM sites)
6. REQSTOR (RACROUTE parameter)
7. SUBSYS (RACROUTE parameter)
8. REQUEST (RACROUTE parameter)
9. SAFDEF (INTERNAL or EXTERNAL)
10. SUBSYS (ACF2 or VMO)

The only RACROUTE parameter that is used on the sort is REQUEST. See the following pages for specific information on each of these fields.

A description of the record format and fields of the SAFDEF record follows:

Record ID	Fields
SAFDEFqual	ID(name) FUNCRET(4|retcode) FUNCRSN(0|rsncode) JOBNAME(mask|********) MODE(IGNORE|GLOBAL|LOG|QUIET) PROGRAM(mask|********) RACROUTE(keyword=value,...,keyword=value) RB(mask|********) RETCODE(0|4|8) USERID(mask|********)

Fields

ID(name)

Specifies an ID name associated with the record. You can specify up to eight characters. This field is optional. We recommend you specify an ID because this name will appear as the first field displayed in the SHOW SAFDEF output. The ID is also the key used for the SHOW SAFDEF subcommand.

Select a name that is unique and that conveys meaning about the type of SAF call you are defining. For example, VERSMS would be an appropriate ID for a SAFDEF record that defines the environment for a REQUEST=VERIFY call from DFSMS.

FUNCRET(4|retcode)

Specifies the SAF function-dependent return code returned to the caller making the RACROUTE request when MODE is specified as IGNORE. For detailed descriptions of possible return codes, see the IBM document entitled *External Security Interface (RACROUTE) Macro Reference Guide*. The default is four.

FUNCRSN(0|retcode)

Specifies the SAF function-dependent reason code to be returned to the caller making the RACROUTE request when MODE is specified as IGNORE. For detailed descriptions of possible reason codes, see the IBM document entitled *External Security Interface (RACROUTE) Macro Reference Guide*. The default is zero.

JOBNAME(mask|******)**

Specifies the user ID of the service machines that apply to this SAFDEF record. You can specify an eight-character user ID or a mask. The default is all service machines.

MODE(IGNORE|GLOBAL|LOG|QUIET)

Specifies the mode you want CA ACF2 for VM to use to process this SAF request. This field defaults to GLOBAL; a value is required. You can specify any one of the following values:

IGNORE

Bypass processing this SAF request

GLOBAL

Process this SAF request using the mode specified in the VMO OPTS record. For generalized resource validations, use the CA ACF2 for VM recommendation to allow or deny the SAF request.

LOG

Process this REQUEST=AUTH call in LOG mode.

QUIET

Process this REQUEST=AUTH call in QUIET mode.

PROGRAM(mask|******)**

This field is not active for VM sites. For more information for this field, see the CA ACF2 for VM for z/OS *Administrator Guid*e.

RACROUTE(keyword=value,...,keyword=value)

Identifies the SAF request being made. Use this field to specify any valid RACROUTE parameters and values. This is a multi-value field. The maximum length that you can specify for the parameter keyword, operator, and value is 64 characters. Separate the entries with commas or blanks. The REQUEST= keyword is required.

You can specify the following relational operators to indicate the presence of a particular value (for example, ENVIR=CREATE) or the presence of a pointer address (ACEE=<). You can use the following operators depending on your type of keyboard:

```
=   Equal to
¬=  Not equal to
><  Not equal to
!=  Not equal to
=<  Pointer value
¬=<  No pointer value
!=<  No pointer value
```

Pointer values are valid only if the keyword operand is specified as a pointer to a data area or data structure (for example, ACEE). When you specify a pointer value, do not also specify a value for the operand. For example, the following request defines a VERIFY request for all user IDs except JOHN, where an ACEE is supplied:

```
RACROUTE(REQUEST=VERIFY,ACEE=<,USERID>notsym.=JOHN)
```

You can mask character data types using the standard CA ACF2 for VM masking characters (asterisk and dash). You can mask other types of data only if the mask is complete. A complete mask indicates that the parameter matches all values. For example, you can specify the following to indicate that this parameter matches all values of USERID:

```
USERID=-
```

Whereas, the following indicates that the USERID option does not apply to this RACROUTE request:

```
USERID>notsym.=-
```

Note: There are SAFDEF restrictions with FASTAUTH processing. FASTAUTH does not allow the use of ENTITY on the RACROUTE field of the SAFDEF.

RB(mask|******)**

This field is not active for VM sites. For more information for this field, see the CA ACF2 for VM for z/OS and OS/390 *Administrator Guid*e.

RETCODE(0|4|8)

Specifies the SAF return code to be returned to the caller making the RACROUTE request when MODE is specified as IGNORE.

0

Allow the request.

4

Let the caller decide how to process the request. This is the default.

8

Deny the request.

USERID(useridmask|******)**

Specifies the user ID that applies to this SAFDEF record. The default is all user IDs.

Creating Multiple SAFDEF Records

To create multiple SAFDEF records, append a qualifier to the record name in the format SAFDEFqual so that you can define a unique record ID for that SAFDEF record for a particular SYSID. The recid can be a maximum of 16 bytes. Therefore, you can specify a qualifier of up to ten characters. It must immediately follow the characters SAFDEF. If you use a period (.) as part of the qualifier string for the record name, CA ACF2 for VM counts it as one of the ten available characters.

For example, if you want to create a SAFDEF record for a VERIFY call from HSM and an AUTH call from HSM, you must use a qualifier to distinguish these two records. You could name the SAFDEF record for the VERIFY call SAFDEF.VERHSM and the SAFDEF record for the AUTH call SAFDEF.AUTHHSM. Naming records using qualifiers enables you to describe multiple unique environments and lets CA ACF2 for VM add the records to the database with unique identifiers.

Viewing SAFDEF Records

The SHOW SAFDEF and SHOW ALL subcommands display the values in the SAFDEF record. The SHOW SAFDEF output displays the following information:

```
SFSFILEA  JOBNAME=********  USERID=********  PROGRAM=********  RB=********
          RETCODE=0          SAFDEF=INTERNAL  MODE=IGNORE       SUBSYS=ACF2
```

Here is how to interpret the output of the SHOW SAFDEF command, from left to right:

SFSFILEA

The ID field of the SAFDEF record is defined as ID(SFSFILEA).

JOBNAME=******

The JOBNAME field of the SAFDEF record was not specified so CA ACF2 for VM used the default, all service machines.

USERID=******

The USERID field of the SAFDEF record was not specified so CA ACF2 for VM used the default, all user IDs.

PROGRAM=******

This field is not active for VM sites. For more information for this field, see the CA ACF2 for VM for z/OS and OS/390 Administrator Guide.

RB=******

This field is not active for VM sites. For more information for this field, see the CA ACF2 for VM for z/OS and OS/390 Administrator Guide.

RETCODE=0

The RETCODE field of the SAFDEF record is defined as RETCODE(0) so that CA ACF2 for VM allows the request when MODE=IGNORE is specified.

SAFDEF=INTERNAL

This field indicates that this SAFDEF record was created by CA. If this record was created by your site in a SAFDEF record, the display would read SAFDEF=VMO.

MODE=IGNORE

The MODE field of the SAFDEF record is defined as MODE(IGNORE) so that CA ACF2 for VM bypasses processing of this request.

SUBSYS=ACF2

This field indicates that this record is for the CA ACF2 for VM subsystem. User-defined SAFDEF records are always targeted to all SAF processing subsystems (****).

Ignoring SAF Calls

Here is how to create a SAFDEF record when you want CA ACF2 for VM to ignore the SAF request.

```
set control(vmo)
CONTROL(VMO)
list like(safdef-)
PRD1 / SAFDEF.XYZ LAST CHANGED BY USER01 ON 07/03/00-12:13
           ID(AUTHXYZ) MODE(IGNORE) RETCODE(0)
           PROGRAM(********) JOBNAME(XYZ-)
           RACROUTE(REQUEST=AUTH,CLASS=DATASET)
```

In this example, the SAFDEF record is for the XYZ product. The product makes a SAF call that CA ACF2 for VM intercepts. In this case, suppose the SAF call is a RACROUTE REQUEST=AUTH,CLASS=DATASET. The site decided to instruct CA ACF2 for VM to bypass processing of this request because it did not want CA ACF2 for VM to validate these calls.

By specifying MODE(IGNORE) and RETCODE(0), CA ACF2 for VM allows product XYZ to access the data set without creating a logging record. The site does not have to create a rule.

CA ACF2 for VM SAF Return Codes

The return codes issued by CA ACF2 for VM for the SAF interface are similar to those documented in the IBM E*xternal Security Interface (RACROUT*E*) Macro Reference Guid*e. Two sets of return codes are returned to the caller. These are the SAF return code and the RACF return and reason codes.

SAF Return Code

The SAF return code is set to a value of 0, 4, or 8. The standard meaning of these codes is:

0

The requested function was processed and has completed successfully.

4

The requested function has not been processed.

8

The requested function was processed and has failed.

If a Logonid Not Found condition, CA ACF2 for VM returns a SAF return code of 8 instead of 4 as documented in the IBM guide referenced in the previous section.

RACF Return and Reason Codes

The RACF return and reason codes are more variable and are based on the type of call being processed. The RACF codes that are returned normally follow the codes documented in the IBM Reference Guide. There may be discrepancies where there is no code matching a specific CA ACF2 for VM function. This is especially true with the VERIFY and VERIFYX calls. The CA ACF2 for VM SAF interface sets RACF return and reason codes based on the message returned by the CA ACF2 for VM SVC when a failure occurs.

The following table shows how the internal CA ACF2 for VM logon validation (ACVALD) return code is translated into the RACF return and reason code. If no table entry is found, a value of 24(18)/00 (return and reason code) is used. The value in parenthesis is the equivalent hex value as shown in the IBM RACROUTE Guide.

CA ACF2 for VM ACVALD Return Code	RACF Return Code	RACF ReasonCode	Message Text
004	04(04)	0	Logonid not found
006	08(08)	0	Password not allowed for logonid
007	08(08)	0	Password required for logonid

CA ACF2 for VM ACVALD Return Code	RACF Return Code	RACF ReasonCode	Message Text
008	48(30)	0	Unauthorized input source for logonid
010	28(1C)	0	Logonid canceled
011	28(1C)	0	Logonid suspended
012	08(08)	0	Password not matched
013	28(1C)	0	Logonid suspended, password violations
014	28(1C)	0	Logonid expired
015	08(08)	0	Invalid password syntax
017	12(0C)	0	Password expired
018	16(10)	0	Invalid new password syntax
019	16(10)	0	Password less than minimum length
020	16(10)	0	New password less than minimum length
021	12(0C)	0	Password expired, cannot be altered
023	16(10)	0	New=old password and old one is expired
025	28(1C)	0	Logonid not active
026	24(18)	0	Access denied by EXPPXIT exit
030	48(30)	0	Logonid has STC attribute
031	48(30)	0	Logonid does not have STC attribute
032	48(30)	0	Logonid/Source combo not valid
033	16(10)	0	Invalid new password syntax for NJE
034	16(10)	0	New password less than minimum length NJE
035	24(18)	0	SEVPRE/SEVPOST exit failed request
036	24(18)	0	Invalid RC from SEVPRE/SEVPOSt exit

CA ACF2 for VM ACVALD Return Code	RACF Return Code	RACF ReasonCode	Message Text
037	16(10)	0	New password denied by NEWPXIT exit
039	48(30)	0	Invalid grouping structure
045	52(34)	0	Not authorized for access to MUSASS
060	48(30)	4	Zone record not found
061	48(30)	4	Logonid time not within shift
062	48(30)	4	Error in shift processing routines
063	48(30)	4	Shift record not found
097	04(04)	0	No default logonid specified
098	32(20)	0	CA ACF2 for VM not initialized
100	20(14)	0	Not authorized for group
115	16(10)	0	New password cannot equal logonid
116	16(10)	0	New password cannot be all numeric
117	16(10)	0	New password has reserved word prefix
118	16(10)	0	New password matches a previous password
128	16(10)	0	Invalid syntax for new password
130	16(10)	0	New password less than minimum length
131	16(10)	0	New password equals old password
132	16(10)	0	New password not allowed
136	16(10)	0	Mindays violation
200	08(08)	0	Invalid password/authority
255	16(10)	0	Password rejected by NEWPXIT exit

Translating SAF Access Levels

The following table shows how CA ACF2 for VM translates SAF access levels into CA ACF2 for VM service levels for resources and into data set accesses. The CA ACF2 for VM translated service levels are not hierarchical as described in the *IBM External Security Interface (RACROUTE) Macro Reference Guide*. This means that the CA ACF2 for VM resource rule you create must explicitly specify all appropriate service values.

SAF Access Level	CA ACF2 for VM Service Level	CA ACF2 for VM Data Set Access
Read	Read	Read
Update	Update	Write
Control	Delete	Write
Alter	Add	Allocate

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z