CA ACF2™ for z/VM

Installation Guide

r12 SP4

ca®
technologies

CA Technologies Product References

This document references the following CA Technologies products:

- CA ACF2™ for z/VM
- CA Top Secret®
- CA Top Secret® for z/VSE

Contact CA Technologies

Contact CA Support

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

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

Providing Feedback About Product Documentation

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

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

Documentation Changes

The following updates have been made in the latest edition:

- Task M9C0I002: Specify Maintenance Minidisks (see page 65)—Added a note to indicate that if you leave the USERID, VADDR, and PWD fields blank, you must specifically ACCESS the required SFS directories in the CAIMAINT PROFILE EXEC.
- Task M9C0I017: Edit CA ACF2 for z/VM $PPF File (see page 91)—Added important information to consider before performing this installation task.

Documentation updates have previously been made to the following topics to reflect support for z/VM Version 6 Release 3.0:

- Operating System Configuration and Service Level
- Task M9C0I005: Specify CP Installation Option
- Task M9C0I006: Modify CP CNTRL File
- Task M9C0I008: Modify CMS CNTRL File
- Task M9C0I015: Create/Edit ACFCP_CAXALOAD File
- Task M9C0I017: Edit CA ACF2 for z/VM $PPF
- Maintenance ID Installation Steps - Step 2: Modify VMXAOPTS
- Sample CP Base Control File
- Sample CMS Base Control File

Contents

Chapter 1: Welcome 11

CA Technology Services: Delivering Business Value On Your Terms ... 11
CA Education Services: Ready When You Are ... 12
CA: Commitment, Quality, Innovation ... 12
For More Information ... 13

Chapter 2: System Requirements 15

Shared File System Prerequisites ... 15
Terminology ... 16
DASD Space Requirements ... 16
- Minidisk Configuration ... 16
- VSAM Database Configuration ... 20
- Required CP Minidisks ... 21

Operating System Configuration and Service Level ... 21
System Prerequisites ... 21
- CA ACF2 for z/VM Virtual Machine Requirements ... 21
- CA-ACTIVATOR Installation Virtual Machine Requirements ... 22
- Storage Requirements ... 22
- User Diagnose Requirements ... 23
- VMDBK Field Requirements ... 23

Maintenance Philosophy ... 23
Genlevel Tapes ... 24
- Program Temporary Fix ... 24

Using CA LMP ... 24

Chapter 3: Implementing CA-CIS 27

CA-ACTIVATOR ... 27
CAICCI ... 28
CAISSF ... 28
Advantage CA-Earl Reporting Service ... 29
CA-ESM ... 29

Chapter 4: Installation Prerequisites 31

Maintenance ID ... 31
PF Keys ... 32

CP or CMS Commands....33
Exiting CA-ACTIVATOR....33
Error Messages....33
System Messages....34
Command Notation....34

Chapter 5: Installation Procedure 37

Preliminary Installation Steps....38
CA ACF2 for z/VM Installation Task Selection Menu....58
Procedure Conventions....59
Test System Installation Tasks....59
Task M9C0I90S: Selection of CA-CIS Services....61
Task M9C0IDOC: Load Documentation Files....62
Task M9C0I000: Specify If New Install or Upgrade....63
Task M9C0I001: CA-ACTIVATOR Product Disk Assignment....64
Task M9C0I002: Specify Maintenance Minidisks....65
Task M9C0I003: Option File Processing....66
Task M9C0I004: Modify CA ACF2 for z/VM CNTRL File....67
Task M9C0I005: Specify CP Installation Option....68
Task M9C0I006: Modify CP CNTRL File....69
Task M9C0I007: Specify CMS Installation Option....72
Task M9C0I008: Modify CMS CNTRL File....73
Task M9C0I009: Specify Miscellaneous Installation Options....76
Task M9C0I010: Copy Files to Test System Disks....79
Task M9C0I011: Create CA ACF2 for z/VM Service Machine Directory Entry....81
Task M9C0I012: Update CP Directory....83
Task M9C0I013: Specify Service Machine Minidisk Links....85
Task M9C0I014: Format the Service Machine Minidisks....86
Task M9C0I015: Create/Edit ACFCP CAXALOAD File....89
Task M9C0I017: Edit CA ACF2 for z/VM $PPF File....91
Task M9C0I018: Modify ACF2ASM EXEC....93
Task M9C0I019: Modify the Service Machine PROFILE EXEC....95
Task M9C0I020: Copy ACF2VM MODULE, ACFSTART EXEC, and CALMP KEYS....98
Task M9C0I021: Modify MLAVM, USERLID, and USERXLID COPY....100
Task M9C0I023: Modify and Assemble ACFFDR....103
Task M9C0I024: Copy ACFFDR TEXT to Service Machine....105
Task M9C0I025: Set Up SMF Minidisks....107
Task M9C0I026: Set Up CMS Databases....109
Task M9C0I027: Convert CMS Databases to VSAM Databases....111
Task M9C0I032: Perform MAINT Tasks....112
Maintenance ID Installation Steps....115

Generate the Production System 143
Final Installation Tasks 152

Chapter 6: Generating a CP and CMS Nucleus 163

Types of CA ACF2 for z/VM Control Files 163
Obtain Control Files 164
Control Statement Syntax 164
Add and Relocate Modules 165
Delete Modules 166
Define Intercepts 167
Rename External References 169
Bypass Intercept Processing 170
Bundle Control Statement into Control Files 170
Base Control Files 171
Component Control Files 172
Generate a Nucleus Using a Base Control File 173
Execute CAXABLD 173
Check the CAXALOAD Control Statements 174
Sample CAXALOAD CHECK Summary Log for CP 175
Sample CAXALOAD Summary Log for CP 175

Chapter 7: Installation Options 177

CAISSF 177
CMS Batch Facility 178
Required BATEXIT2 Exit 178
Installing the CMS Batch Interface 178
Submitting Jobs on Behalf of Another User 179
DASD Dump and Restore (DDR) Support 179
Database Synchronization Component 180
Prerequisites 181
Installing the Database Synchronization Component 181
Using the Database Synchronization Component 184
Database Synchronization Component Processing 190
Complex Synchronizing Environments 192
DIAG 'A0' Subcode '04': CA ACF2 for z/VM Password Validation 193
DirMaint Consideration with ISF/CSE Complexes 194
Support for DirMaint 194
Installing Password Validation support. 195
Installing Command Limiting and Message Logging Support 195
National Language Support 197

Defining Language Messages 198
Implementation 203
Considerations 204
PAM Server Support 204
Configuring the PAM Server 204
Configuring CA ACF2 for z/VM Security for Use With the PAM Server 212
Console Interface 215
PASSTHRU Release 1.3 Support 218
RSCS System Component 219
CA-REGISTER 220
Configuring CA-REGISTER 220
Shared Database Support 222
Activating Shared Databases on VM Systems 222
Shared Database Considerations for VM-Only Sites 231
Expanded Rule Size 236
TCP/IP FTP Interface 237
TCP/IP NFS Interface 240
TCP/IP REXEC Interface 241
Unit Record Logging Support 242
Implementing Unit Record Logging Support 243
Output 243
VM Batch System Support 243
Installing VM Batch Facility Support 244
VSAM AMSERV Support 250

Chapter 8: Troubleshooting 251
Diagnostic Procedures 252
Incorrect Installation Symptoms 254
Collecting Diagnostic Data 256
Interpreting Diagnostic Data 256
Contacting Customer Support 257
Accessing the Online Client Support System 257

Chapter 9: Field Definition Record 259
About the VMXAOPTS Macro 259
ACFFDR Macro Summary 260
Modify and Activate ACFFDR Changes 261
@CFDE-Create Field Definition Entry Macro 262
@CFDEDFT-Generate Default Value for Database Record Field 273
@DDSN-Dynamic Data Set Allocation Macro 274

@GNVMFDR-Field Definition Record Generator Macro 275
@GROUP-Group Names for Logonid Display 275
@HEADER-Heading Formatting for Logonid Display 277
@MLID-Logonid Compression Macro 277
@SETUP-DSECT Map Initialization Macro 279
@SMF-System Management Facility Macro 279
ACFSERVE Subcommands 285
@SRF-System Request Facility Macro 286
@SYSID-System Identification Macro 288
@UID-User Identification String Macro 289
@VALUES-Generate Verification Values 289

Appendix A: Applying Genlevel Updates 291

Refresh Tasks 296
Task M9C0R010: Copy Files to Test System Disks 297
Task M9C0R020: Replace ACF2VM MODULE 299
Task M9C0R023: Reassemble ACFFDR 300
Task M9C0R024: Copy ACFFDR TEXT to Service Machine 301
Task M9C0R032: Perform MAINT Tasks 303

Index 305

Chapter 1: Welcome

This guide introduces you to CA ACF2™ for VM (CA ACF2 for z/VM). By the time you have finished reading this guide, you will have an overview of the wide scope of the product and its usability will be familiar to you. It is important to us that you feel comfortable with CA ACF2 for z/VM before you begin to use it.

This section contains the following topics:

CA Technology Services: Delivering Business Value On Your Terms (see page 11)
CA Education Services: Ready When You Are (see page 12)
CA: Commitment, Quality, Innovation (see page 12)
For More Information (see page 13)

CA Technology Services: Delivering Business Value On Your Terms

CA Technology Services is a global organization of highly trained, experienced professionals who are determined to provide you with the technical expertise you need, when and how you need it. From implementing a CA solution to helping you get the most out of the CA technology that you have, CA Technology Services is committed to delivering business value to you on *your* terms.

Our professionals understand your unique business needs and work closely with you to assess which technology is right for your business. Whether the assignment is large or small or you need a custom, stand-alone, or packaged solution, we tailor our efforts to meet your business demands.

By offering a broad range of flexible services, we help you maximize your investment in our technology, achieve more efficient IT performance, and better manage your infrastructure, security, storage, applications, and data. Such flexibility ensures that you reach your time-to-market goals while improving your business performance.

Why not ask your CA representative for more information about how a CA Technology Services professional can help your organization get the most out of your CA business solutions?

CA Education Services: Ready When You Are

The goal of CA Education Services is to help you realize the full potential of your CA software investment. To meet this goal, our high-quality instructors strive to understand your specific training requirements, and then deliver the right kind of training when, where, and how you need it.

All CA instructors are fully certified and offer a wealth of hands-on enterprise management experience gained in working with today's largest and most complex businesses. Whether your training is web-based, self-paced, or in the traditional classroom, you always receive the most up-to-date instruction and expertise that is available. The knowledge you gain through training prepares you to successfully leverage the capabilities of your CA software.

Why not ask your CA representative how our training and education programs can help you get more out of your CA business solutions?

CA: Commitment, Quality, Innovation

For more than a quarter century, CA has been developing and supporting software solutions that are currently used by more than 99 percent of the Fortune 500 companies in more than 100 countries. CA is committed to offering leading technologies in flexible partnerships to help you derive full value from your software investments.

At CA, we are committed to offering simple and meaningful solutions to your complex problems, and to delivering management solutions that offer security, reliability, availability, and performance. We work hard to achieve the highest levels of quality in our solutions to help you meet your changing business needs.

To meet these needs, CA's world-class solutions address all aspects of process management, information management, and infrastructure management with six focus areas:

- Enterprise management
- Security
- Storage
- Portal and business intelligence
- Database management
- Application life cycle management and application development

In addition, our innovative approach to technology is carried over into our innovative business solutions. From a revolutionary new business model to a dedicated customer relationship organization, CA is responding to your changing business needs.

We know what it takes to deliver and support valuable solutions 24 hours a day, 7 days a week, 365 days a year while maintaining the highest standards for quality and innovation:

- We are the first global enterprise software company to meet the exacting standards for worldwide ISO 9002 certification.
- We have earned over 150 patents for innovative software solutions.
- We have the highest caliber software developers and consultants in the industry.

We also know you expect us to stand by our commitments. And we do.

For More Information

After reading this *Installation Guide*, you can refer to the numerous resources available to you for additional information. In addition, you can obtain procedural information and answers to any questions you may encounter by accessing the CA web site at ca.com.

Chapter 2: System Requirements

This chapter contains information about the requirements necessary for installing CA ACF2 for z/VM on z/VM systems.

CA ACF2 for z/VM r12 uses CA-ACTIVATOR r1.2 at the 0704 or later genlevel for installation. The CA-CIS for VM release 1.0 tape contains CA-ACTIVATOR r1.2.

We assume your site follows standard IBM maintenance procedures. The sample commands we use to describe the CA ACF2 for z/VM installation process also assume your site follows standard IBM procedures and control files.

To install the CMS security feature, you must have the High Level Assembler (5696-234) from IBM since IBM requires the High Level Assembler to assemble CMS modules. The High Level Assembler supports more advanced Assembler language macro functions and is a requirement for several things in z/VM. The High Level Assembler is not part of the base z/VM package at this time. It is a separate product from IBM.

If you do not properly install CA ACF2 for z/VM, you may not be able to use your VM system. To avoid this situation, we recommend you install CA ACF2 for z/VM on a test VM system. This test system can be a processor or LPAR dedicated to testing system software or a guest machine running under VM. After testing, migrate to a production CP.

This section contains the following topics:

Shared File System Prerequisites (see page 15)
Terminology (see page 16)
DASD Space Requirements (see page 16)
Operating System Configuration and Service Level (see page 21)
System Prerequisites (see page 21)
Maintenance Philosophy (see page 23)
Genlevel Tapes (see page 24)
Using CA LMP (see page 24)

Shared File System Prerequisites

To implement CA ACF2 for z/VM SFS file protection, you must install the CA-ESM component of CA-CIS. We discuss specific IBM APARs required for SFS file protection in System Prerequisites section.

Terminology

Throughout this guide, the following terms apply:

VM

Refers to VM Version 5 Release 1.0 and above.

IBM-supplied system generation exec

Refers to the VMFBLD EXEC.

DASD Space Requirements

The following sections contain information on the space requirements for CA-ACTIVATOR, CP minidisks, and CMS minidisks.

Minidisk Configuration

If you are installing CA ACF2 for z/VM for the first time, you need to define and format the following disks.

Required CAIMAINT Minidisks

Disk Description	Owner	Disk	Notes	4 Kb CMS Blocks	3390 Cylinders
CA-ACTIVATOR system disk	CAIMAINT	191		2700	15
CA-ACTIVATOR staging disk	CAIMAINT	291	13	20700	115

Optional CAIMAINT Minidisks

Disk Description	Owner	Disk	Notes	4 Kb CMS Blocks	3390 Cylinders
CA-ACTIVATOR production system disk	CAIMAINT	391	14, 15	360	2

Required Minidisks

Disk Description	Owner	Disk	Notes	4 Kb CMS Blocks	3390 Cylinders
CA ACF2 for z/VM test local options	MAINT	2A0		540	3
CA ACF2 for z/VM test generation files	MAINT	2A1		3960	22
CA ACF2 for z/VM test command files	MAINT	2A3	1,9	360	2
CA ACF2 for z/VM test report generator files	MAINT	2A3	1, 10	720	4
CA ACF2 for z/VM test help files	MAINT	2A3	1, 11	3060	17
CA ACF2 for z/VM test full-screen files	MAINT	2A3	1, 12	900	5

Optional Minidisks

Disk Description	Owner	Disk	Notes	4 Kb CMS Blocks	3390 Cylinders
CA ACF2 for z/VM production local options	MAINT	3A0	15	540	3
CA ACF2 for z/VM production generation	MAINT	3A1	15	3960	22
CA ACF2 for z/VM production command files	MAINT	3A3	1, 9, 15	360	2
CA ACF2 for z/VM production report generator files	MAINT	3A3	1, 10, 15	720	4
CA ACF2 for z/VM production help files	MAINT	3A3	1, 11, 15	3060	17

Disk Description	Owner	Disk	Notes	4 Kb CMS Blocks	3390 Cylinders
CA ACF2 for z/VM production full-screen files	MAINT	3A3	1, 12, 15	900	5

Required Service Machine Minidisks

Disk Description	Owner	Disk	Notes	4 Kb CMS Blocks	3390 Cylinders
Service machine A-disk	ACF2VM	191	2	180	1
Service machine B-disk	ACF2VM	193	3	540	3
Backup file disks	ACF2VM	195	4	2160	12
Logonid database	ACF2VM	301	5	720	4
Rule database	ACF2VM	302	5	360	2
Infostorage database	ACF2VM	303	6	360	2
SMF disks	ACF2VM	200, 201 202	6	900	5/disk

Optional Service Machine Minidisks

Disk Description	Owner	Disk	Notes	4 Kb CMS Blocks	3390 Cylinders
STARTUP OPTIONS file	ACF2VM	197	7		
Alternate Logonid database	ACF2VM	3A1	8		
Alternate Rule database	ACF2VM	3A2	8		
Alternate Infostorage database	ACF2VM	3A3	8		

Notes:

1. You can combine all 2A3 minidisks as a single 26 cylinder minidisk (4680 blocks). You can also combine all 3A3 minidisks as a single 26 cylinder minidisk (4680 blocks). We recommend you leave enough space for future maintenance.
2. This disk contains the PROFILE exec for the CA-ACF2 service machine.

3. This disk contains the ACFFDR TEXT file, the ACF2VM MODULE file, the ACFSTART EXEC, the CALMP KEYS file, and any user-written exits that reside in the CA ACF2 for z/VM service machine. The CA ACF2 for z/VM service machine accesses in read-only to let another user update the ACFFDR while the service machine is running. Whenever you issue the ACFSERVE RELOAD FDR command or the ACFSERVE RELOAD LMP command and you have accessed the disk in read-only, the CA ACF2 for z/VM service machine reaccesses this minidisk to ensure it uses the latest version of the ACFFDR TEXT or CALMP KEYS files.

4. This disk contains backups of the CA ACF2 for z/VM databases. The minidisk must be large enough for all backup files, have enough extra space for the largest backup file, and room for growth. Backup files have a filetype of BACKUPDB. The files have a record format of sequential variable length for both CMS and VSAM databases. The space allocation shown assumes the sample CMS database sizes. After you have successfully taken backups, use the actual size of the backup files to determine the disk size.

5. During installation, these minidisks will be formatted with 4k blocks and reserved with the CMS RESERVE command. We have based the space allocation shown in the chart on the following specification:

   Logonid database

   Holds 2,880 logonid records. Each additional cylinder of 3390 disk (180 4 Kb blocks) holds an additional 720 logonid records.

   Rule database

   Holds 1,440 rule records, assuming an average rule record length of 2 Kb. Each additional cylinder of 3390 disk (180 4 Kb blocks) holds 360 more rule records.

   Infostorage database

   Holds 720 infostorage records, assuming an average infostorage record length of 2K. Each additional cylinder of 3390 disk (180 4 Kb blocks) holds 360 more infostorage records.

6. During installation, these minidisks will be formatted with 4k blocks and reserved with the CMS RESERVE command. The CA ACF2 for z/VM service machine automatically accesses the SMF minimidisks. If you add SMF disks at a later time, they will need to be formatted with 4k blocks and reserved with the CMS RESERVE command with a file id of:

   ```
   SMF 00000000 fm
   ```

7. This disk contains the CA ACF2 for z/VM STARTUP OPTIONS file. This disk must be one cylinder on CKD DASD or a 12 or more block FBA minidisk.

8. These disks contain alternate CMS databases if you want to use them. Set them up using the criteria in number 6.

9. This disk contains the CA ACF2 for z/VM command modules. Let all users who execute the ACF, ACFCOMP, ACFDCMP, and ACFNRULE commands access it.

10. This disk contains the CA ACF2 for z/VM report generator modules and exec. Let all users who generate CA ACF2 for z/VM reports access it.

11. This disk contains the CA ACF2 for z/VM help files. Let all users who can use the CA-ACF2 help facility access it.

12. This disk contains executable CA ACF2 for z/VM full-screen files. Let all users who can use the CA ACF2 for z/VM full-screen feature access it.

13. If you have already installed other CA products on this CAIMAINT disk, you need this much free space on this disk before beginning to install CA ACF2 for z/VM. This ensures you have sufficient space for both the initial CA ACF2 for z/VM installation and future genlevel maintenance. If you want to place the CA ACF2 for z/VM documentation on this disk, increase the size by 30 cylinders (5400 blocks).

14. If you have already installed other CA products on this CAIMAINT disk, you need this much free space on this disk before generating a production CA ACF2 for z/VM system.

15. If you maintain separate test and production systems with separate CAIMAINT machines for each system, this disk is not required.

VSAM Database Configuration

If you are using Shared Database Support (systems shared across different operating systems), you must allocate space for the CA ACF2 for z/VM primary databases. The following table lists the number of records that fit on one cylinder of data component space. It also lists the number of records that fit on one track of index component space. Use the data in the following chart and your projected record count to estimate the space you need for the CA ACF2 for z/VM databases:

Device Type	Logonid Data	Logonid Index	Rules Data	Rules Index	Infostorage Data	Infostorage Index
3380	495	3220	165	3220	165	805
3390	590	3860	195	3860	195	966

We have based these estimates on the following assumptions. Make adjustments as needed for your site.

- The logonid record size is 1,024 bytes and 4,096 bytes for access rule sets and infostorage records.
- The control interval size for all data and index components is 4,096 bytes.

Be sure to allocate space for the backup data sets. Space allocations for the sequential backup data sets should be approximately equal to the size of the catalog component of the corresponding cluster.

Required CP Minidisks

You should access all the minidisks required to assemble CP modules and generate CP. Refer to the IBM documentation for the appropriate disks to generate your system.

Operating System Configuration and Service Level

CA ACF2 for z/VM operates properly at the following levels:

- z/VM Version 6 Release 4.0
- z/VM Version 6 Release 3.0
- z/VM Version 6 Release 2.0
- z/VM Version 6 Release 1.0
- z/VM Version 5 Release 4.0
- z/VM Version 5 Release 3.0
- z/VM Version 5 Release 2.0

System Prerequisites

To install CA ACF2 for z/VM properly, the operating system configuration and service level, the CA ACF2 for z/VM and installation virtual machines, the user diagnose codes, and the VMDBK control block must meet certain requirements.

There are no required IBM APARs at this time.

CA ACF2 for z/VM Virtual Machine Requirements

One of the components of CA ACF2 for z/VM is a disconnected service machine. It internally autologs this virtual machine when the first user logs on after VM IPL is complete. This service machine runs continuously as long as VM is up. This service machine performs all that need CA ACF2 for z/VM database activity or logging. The user ID of the CA ACF2 for z/VM service machine defaults to ACF2VM, but you can change it through the SRVMID operand of the VMXAOPTS macro.

If you use VSAM shared databases, the CA ACF2 for z/VM service machine must have at least 8Mb of virtual storage, defined in the CP directory. If you do not use VSAM databases, 5Mb of storage is sufficient. For each additional user beyond 512, add 1 Kb of storage.

The CA ACF2 for z/VM service machine runs in an ESA mode virtual machine and will exploit storage above the 16 Mb line.

CA-ACTIVATOR Installation Virtual Machine Requirements

The CAIMAINT ID requires a virtual machine size of 6 Mb to install CA ACF2 for z/VM. You can define this size in the CP directory or by CP commands before starting the installation procedure.

Storage Requirements

This section describes the storage requirements for CA ACF2 for z/VM for VM r12. These figures are a guideline. The actual storage requirements vary based on the application of maintenance.

CP Free Storage

CA ACF2 for z/VM dynamically acquires CP free storage for all validations.

It frees the acquired storage when it completes validation. The following is the requirements for CP free storage acquired at initialization:

Type of Storage	Bytes
CA ACF2 for z/VM free storage	1,440
CA ACF2 for z/VM options free storage (HCPAC0)	750
Total free storage	2,190

Options free storage requirements vary from site to site. To estimate the length of your assembly, examine the length of your HCPAC0 assembly. The amount of options free storage you must allocate is approximately the length of HCPAC0. As distributed, HCPAC0 is 750 bytes. (CA ACF2 for z/VM rounds up all storage units because CP only allocates free storage in units of double words.)

CMS file level security, the requirements per logged-on user for CP free storage are:

Storage	Bytes
Base free storage	328
Free storage for every eight linked disks	112
Total free storage (for user with no more than eight disks)	**440**

Free storage for linked disks is allocated in units of 112 bytes.

CP free storage requirements per logged-on user without CMS file level security are:

Storage	Bytes
Base free storage	328
Total free storage	**328**

CP Resident Nucleus

The storage requirement for the CP resident nucleus is:

Type of Module	Kb
Required resident modules	80
Total resident modules	80

User Diagnose Requirements

CA ACF2 for z/VM dynamically implants two user diagnose codes: ACF2 and 0ACF. Inserting these diagnose codes is transparent to the operating system and non-CA ACF2 for z/VM code.

VMDBK Field Requirements

CA ACF2 for z/VM requires a single fullword in the VMDBK. The VMUSRFD operand of the VMXAOPTS macro specifies the name of this field, normally one of the VMDUSER*n* fields. The default is VMDUSER8.

Maintenance Philosophy

CA ACF2 for z/VM supplies a special control file to simplify the CP and CMS maintenance of your system. The CAXALOAD control file has a filetype of CAXALOAD. We refer to it as a CAXALOAD control file. It contains input data for a CMS module name CAXALOAD that is a front-end of HCPLDR. The control file does two things. It implements the base CA ACF2 for z/VM support in object-code-only form, but not entirely. Source code updates in entry points, instead of source code intercepts, implement a number of optional CA ACF2 for z/VM CMS updates. As a result, you do not need to reassemble CA ACF2 for z/VM intercepted modules. Secondly, the control file overrides CP and CMS LOADLIST execs without modifying them.

For details on these special control files and the CA ACF2 for z/VM modules that use them, see Generating a CP and CMS Nucleus Using CA ACF2 for z/VM.

Genlevel Tapes

CA produces and mass mails genlevel tapes for customer support.

CA ACF2 for z/VM genlevel tapes contain fixes to code. We supply them regularly to bring our client base up to a new level in a release.

We label CA ACF2 for z/VM genlevel tapes with a genlevel indicator. For example, the base genlevel for CA ACF2 for z/VM r12 is C00705AM901. Assuming the first genlevel update is delivered in December of 2007, the distribution tape would be labeled C00712AM901. This indicates that it is a r12 tape (**C0**0712M901), that it is dated December 2007 (C0**0712**AM901), that it is CA ACF2 for z/VM (C00712**AM9**01), and that it is the first version of the genlevel update (C00712AM9**01**). This last number is normally 01. See “Applying Genlevel Updates” for more information about genlevel updates.

We usually include a documentation update with a genlevel tape. The documentation update covers any technical changes included on the tape.

Program Temporary Fix

For information about applying fixes, see the CA-ACTIVATOR documentation. In emergency situations, you can use the ACF2FIX utility to apply Program Temporary Fixes (PTFs). See the *Reports and Utilities Guide* for information about running the ACF2FIX utility.

Using CA LMP

CA ACF2 for z/VM requires CA LMP (License Management Program), to initialize correctly. CA LMP provides a standardized and automated approach to the tracking of licensed software. CA LMP is provided as an integral part CA ACF2 for z/VM.

Examine the CA LMP Key Certificate you received with your CA ACF2 for z/VM installation or maintenance tape. That certificate contains the following information:

Fields	Descriptions
Product Name	The trademarked or registered name of the CA software solution licensed for the designated site and CPUs.

Fields	Descriptions
Product Code	A two-character code that corresponds to CA ACF2 for z/VM.
Supplement	The reference number of your license for CA ACF2 for z/VM, in the format nnnnnn - nnn. This format differs slightly inside and outside North America, and in some cases might not be provided at all.
CPU ID	The code that identifies the specific CPU for which installation of CA ACF2 for z/VM is valid.
Execution Key	An encrypted code required by CA LMP for CA ACF2 for z/VM initialization. During installation, it is referred to as the LMP Code.
Expiration Date	The date (*ddmmmyy* as in 01AUG00) your license for CA ACF2 for z/VM expires.
Technical Contact	The name of the technical contact at your site, which is responsible for the installation and maintenance of CA ACF2 for z/VM. This is the person to whom CA addresses all CA LMP correspondence.
MIS Director	The name of the Director of MIS, or the person who performs that function at your site. If the title but not the individual's name is indicated on the Certificate, you should supply the actual name when correcting and verifying the Certificate.
CPU Location	The address of the building where the CPU is installed.

The CA LMP execution key, provided on the Key Certificate, must be added to the CALMP KEYS to ensure proper initialization of CA ACF2 for z/VM. CALMP KEYS reside on the service machine's 193 minidisk.

The parameter structure for member KEYS is as follows:

```
PROD(pp) DATE(ddmmmyy) CPU(tttt-mmmm/ssssss) LMPCODE(kkkkkkkkkkkkkkkk)
```

Descriptions of the variables follow:

pp

Required. The two-character product code. For any given CA LMP software solution, this code agrees with the product code already in use by the CAIRIM initialization parameters for earlier genlevels of that software solution.

ddmmmyy

The CA LMP licensing agreement expiration date.

tttt-mmmm

Required. The CPU type and model (for example: 9672-RB5) on which the CA LMP software solution is to run. If the CPU type and/or model require less than four characters, blank spaces are inserted for the unused characters.

ssssss

Required. The serial number of the CPU on which the CA LMP software solution is to run.

kkkkkkkkkkkkkkkk

Required. The execution key needed to run the CA LMP software solution. This CA LMP execution key is provided on the Key Certificate shipped with each CA LMP software solution.

The following is an example of a control card for the CA LMP execution software parameter. Although this example uses the CA ACF2 for z/VM two-character product code, the CA LMP execution key value is invalid and is provided as an example only!

```
PROD(K5) DATE(01MAR02) CPU(9672-RB5 /370623) LMPCODE(52H2K06130Z7RZD6)
```

Chapter 3: Implementing CA-CIS

CA ACF2 for z/VM uses the following components of CA-CIS, Common Infrastructure Services:

- CA-ACTIVATOR
- CAICCI
- CAISSF
- Advantage™ CA-Earl®
- CA-ESM

Only CA-ACTIVATOR is required to install or use CA ACF2 for z/VM. Advantage CA-Earl is always installed, but you are not required to use it. You have the opportunity during your product's installation to indicate whether you want to install the CA-CIS components. If you indicate that you do want to install the CA-CIS components, CA-ACTIVATOR automatically leads you through the installation process (unless you have already installed these components for use with other products).

See the CA-CIS *Getting Started* and *Reference Guide* for more information about CA-CIS.

This section contains the following topics:

CA-ACTIVATOR (see page 27)
CAICCI (see page 28)
CAISSF (see page 28)
Advantage CA-Earl Reporting Service (see page 29)
CA-ESM (see page 29)

CA-ACTIVATOR

CA-ACTIVATOR provides better control over installation and maintenance procedures to ensure their successful completion.

Note: Only the CA-ACTIVATOR *Reference Guide* contains instructions for installing the CA-ACTIVATOR component. The CA-CIS CA-ACTIVATOR Supplement guides you in using CA-ACTIVATOR to install all **other** CA-CIS components.

The CA-ACTIVATOR features include:

- Interactive, full-screen dialogs and HELP panels
- Customization steps that let you keep control of software installation
- Automatic incorporation of options during installation

- Recording of all software-solution and component tailoring
- Modifying software-solution options whenever needed
- Validation and demonstration of software installation
- Error-free maintenance.

CAICCI

CAICCI, CAI Common Communications Interface, is a communications facility that offers a simple yet flexible approach enabling CA solutions to communicate with one another. This facility provides a layer that isolates application software from the specifics of the communications environment. The routines that make this possible are grouped under the CA z/OS service code, W410. CAICCI features include:

- Single point of control
- Multiple platform support
- Performance optimization
- Peer-to-peer (program-to-program) communication
- Parallel conversations
- Dynamic installation configuration
- Ease of customization
- Error handling

CAISSF

CAISSF, CAI Standard Security Facility, allows CA software to offer standardized security interfaces, regardless of the underlying access control software. CAISSF offers user authentication and resource access validation facilities, and can interface with CA security products (CA ACF2 or CA Top Secret) or compatible non-CA security products. Check your CA-CIS documentation for the CAISSF service code. For CA security products, CAISSF features include:

- A single security mechanism
- Isolation of CA enterprise solutions from CA or vendor mechanisms.

For non-CA security products, some of the CAISSF features include:

- Resource class translation
- Access level translation
- Selective logging of requests

- Request type control
- Message support for failed access.

Advantage CA-Earl Reporting Service

The Advantage CA-Earl reporting component is a user-friendly report definition facility with the power of a comprehensive programming system. Advantage CA-Earl allows you to modify and print the contents and layout of a predefined CA product report using English-like statements. Check your CA-CIS documentation for the Advantage CA-Earl service code. Some of the Advantage CA-Earl Reporting Service features are:

- Page, user, and field headings
- Automatic subtotaling and totaling capabilities
- Automatic data editing
- Full arithmetic computational facilities
- High-level capabilities
- Enhanced printed output control.

For more information about Advantage CA-Earl, see the *Reporting with* Advantage *CA-Earl Guide*.

CA-ESM

CA-ESM release 1.1 is an External Security Manager that allows CA ACF2 for z/VM to provide security to other products with an ESM option that issue Racroute calls. Some of the CA-ESM features include:

- Securing user's ability to manipulate SFS files. Through CA-ESM, CA ACF2 for z/VM can secure the user's ability to create, delete, or change SFS files and directories.
- Consolidating violation reports. By tying ESM calls into your system security package, all security violations are consolidated into your CA ACF2 for z/VM reports.

There are definite benefits to using CA-ESM at your site. These benefits include:

- Can eliminate the need for a product specific interface between another product and CA ACF2 for z/VM.
- Single security environment. CA-ESM ties your other products with ESM options into your regular CA ACF2 for z/VM environment, so that there is only a single security environment. This helps to simplify security administration, because there are no new commands for administrators to learn.

  Greater degree of security protection. By having your other products with ESM options tied into CA ACF2 for z/VM, a greater degree of security protection is guaranteed than would be using only the product's security.

Chapter 4: Installation Prerequisites

The following topics describe the preparation for installing CA ACF2 for z/VM.

This section contains the following topics:

Maintenance ID (see page 31)
PF Keys (see page 32)
CP or CMS Commands (see page 33)
Exiting CA-ACTIVATOR (see page 33)
Error Messages (see page 33)
System Messages (see page 34)
Command Notation (see page 34)

Maintenance ID

You must designate a CA product maintenance virtual machine to install and maintain CA ACF2 for z/VM from. If you installed other CA-Unicenter VM products, you defined a common CA-Unicenter VM product maintenance ID (such as CAIMAINT).

If you are converting from a previous CA ACF2 for z/VM release and are installing r12 in NOAUTO mode, you must add your maintenance ID (CAIMAINT) to the FORCEID list in VMXAOPTS and regenerate your CP nucleus before installation. Define the CA-ACTIVATOR product maintenance virtual machine with the following minidisks:

- A 191 disk to hold the CA-ACTIVATOR files and serve as a work area.
- A 291 disk to act as the CA-ACTIVATOR test staging disk. We recommend that you do not allow general users access to this disk.
- A 391 disk to act as the CA-CIS production disk.

See the chapter "System Requirements" for space requirements for CA-ACTIVATOR and CA ACF2 for z/VM. As an additional safeguard, CA-ACTIVATOR checks that enough disk space is available before installing a new product.

PF Keys

We display PF keys at the bottom of each panel. Use them to perform the indicated panel functions. The table below shows the PF keys as we define them. Generally, you use PF2 for the primary action on a panel. Other actions are usually assigned to PF5 or PF6.

PF Key	Function	Description
PF1	HELP	Displays online HELP and field descriptions.
PF2	SAVE	Saves the values displayed on the panel.
	ACTION	Performs the action detailed on the panel.
	EXECUTE	Performs the action detailed on the panel.
	CONFIRM	Confirms that the task has been performed.
	FORMAT	Formats the specified minidisks.
	PERFORM	Causes the task to perform.
	FILE	Saves the options you specified.
	MODIFY	Lets you modify the values.
	EDIT	Lets you edit the panel.
	XEDIT	Lets you change a file.
	COPY	Copies the files you specified.
	BUILDMAC	Builds the MACLIB.
	ASSEMBLE	Assembles the ACFFDR.
	PROCESS	Processes files.
PF3	END	Ends processing and exits a particular task. The cursor returnsto the CA-ACTIVATOR Task Selection Menu.
	QUIT	Quits current processing and exits a particular task. No actionis taken. The cursor returns to the CA-ACTIVATOR Task Selection Menu.
PF4	RETURN	Exits task without processing. The cursor returns to the Primary Menu.
PF5	ADD USER	Lets you define a user to the CP directory.
	BYPASS	Bypasses all processing. The cursor returns to the CA-ACTIVATORTask Selection Menu.
PF6	XEDIT	Lets you change a file.

PF Key	Function	Description
	VIEW	Displays a file.
	REPLACE	Replaces previously existing files with newly created ones.
PF7	BACKWARD	Scrolls backward three lines.
PF8	FORWARD	Scrolls forward three lines.
PF9	SCRHLPUP	Scrolls the help information towards the top.
PF10	SCRHLPDN	Scrolls the help information towards the bottom.

You tab forward to each field in a panel. Most fields on a panel display default values. To accept a default value, tab to the next field. To change a field entry, type over the existing entry. To delete an entry, use the space bar or the Erase EOF key to erase the entry.

CP or CMS Commands

To issue one or more CP or CMS commands while processing a task, you can respond CMS to most prompts. CA-ACTIVATOR places you into a conversational mode where you can issue the necessary commands. Enter a null response to return to the original CA-ACTIVATOR prompt.

Exiting CA-ACTIVATOR

To exit CA-ACTIVATOR while processing a task, respond EXIT to most prompts. CA-ACTIVATOR returns you to the Task Selection Menu and leaves the status of the processing task INCOMPLETE. You can select that task again at a later time.

Error Messages

See the CA-ACTIVATOR *Reference Guide* for detailed explanations of error messages.

System Messages

System messages indicate the status of the installation tasks. Each task can consist of one or more required actions. When CA-ACTIVATOR successfully completes or executes a task, the following message, indicating it completed all required prerequisites or actions, appears:

```
This task has completed successfully
```

You can proceed to the next task. If you did not perform a prerequisite task or some portion of the task did not successfully complete, CA-ACTIVATOR marks the task as incomplete. You must reexecute the task after completing the failed or missing portion before you can proceed to next task.

Command Notation

This guide uses the following command notation. Enter the following exactly as they appear in command descriptions:

Type of Characters	Description
UPPERCASE	Identifies commands, keywords, and keyword values that you must code exactly as shown.
MIXed Cases	Identify command abbreviations. The uppercase letters are the minimum abbreviation; lowercase letters are optional
Symbols	You must code all symbols, such as commas, equal signs, and slashes exactly as shown.

The following clarify command syntax; do not type these as they appear:

Type of Characters	Description
lowercase	Indicates a variable that you must supply.
[ ]	Identify optional keywords or parameters.
{ }	Require that you choose one or more of the keywords or parameters listed.
<u>underlining</u>	Shows default values that you do not have to specify.
|	Separates alternative keywords and parameters, choose one.

Type of Characters	Description
...	Means you can repeat the preceding items or group of items more than once.

Sample Command	Explanation	
ACFNRULE{ruleid|KEY(ruleid)}	ACFNRULE	Command abbreviation.
TYPE(rsrctype)]-	TYPE	Optional value you can specify.
{[ADD(ruleentry)…]-	ADD	Optional keyword.
[DELETE(ruleentry)…]}-	DELETE	Optional keyword.
[LIST|NOLIST] -	LIST	Default, you do not have to specify.
[VERIFY|NOVERIFY]	VERIFY	Default, you do not have to specify.

Chapter 5: Installation Procedure

This chapter describes the procedures for installing CA ACF2 for z/VM using CA-ACTIVATOR. We have divided these procedures into steps for you to perform. In these steps are operational units called tasks.

You select each task from a CA-ACTIVATOR Task Selection Menu and respond to prompts on the CA-ACTIVATOR panels to perform the task.

The following procedure is for the target system for this CA ACF2 for z/VM installation. It is important that your current CP directory correctly reflect any disks you use in this procedure.

This section contains the following topics:

Preliminary Installation Steps (see page 38)
CA ACF2 for z/VM Installation Task Selection Menu (see page 58)
Procedure Conventions (see page 59)
Test System Installation Tasks (see page 59)

Preliminary Installation Steps

Note: Perform the following steps on your CAIMAINT ID.

To install CA-ACTIVATOR, follow these steps :

1. If you are using actual tapes, mount the CA ACF2 for z/VM product tape as 181 and the CA-CIS tape as 182:

```
q v tape
TAPE 0181 ON DEV 0582 R/W
TAPE 0182 ON DEV 0583 R/W
Ready; T=0.01/0.01 08:36:47
 * 181 = Product tape
 * 182 = CA-CIS tape
 rew 181
Rewind complete
Ready; T=0.01/0.01 08:39:06
 rew 182
Rewind complete
Ready; T=0.01/0.01 08:39:10
```

As an alternative to using actual tapes, you may use CDTAPE, a utility provided by CA as part of electronic distribution. If you have downloaded CA-ACF2, CA-CIS, and the CDTAPE utility, you can place the CDTAPE images on a minidisk. CDTAPE front ends the TAPE command, allowing the CDTAPE images to be read when TAPE commands are issued, instead of the actual tapes.

If you are using CDTAPE images instead of actual tapes, do the following commands after accessing the minidisk(s) that contain the CDTAPE utility and the CDTAPE images for CA-ACF2 and CA-CIS, using filemodes C and above (do not use A or B):

```
cdtape start

cdtape mount acfimage
* where acfimage = Filename of the CA-ACF2 CDTAPE image file

cdtape mount cisimage 182
* where cisimage = Filename of the CA-CIS CDTAPE image file
```

2. Locate the fifth file on the CA-CIS tape:

```
Ready; T=0.01/0.02 08:39:14
 tape fsf 4 (182
```

3. Load the CA-CIS files:

```
Ready; T=0.01/0.01 08:39:28
 tape load * * A (182 disk)
```

   The only disks you need to have access to at this point are the 191 disk, accessed as A, and the CAIMAINT test system disk (usually 291), accessed as B.

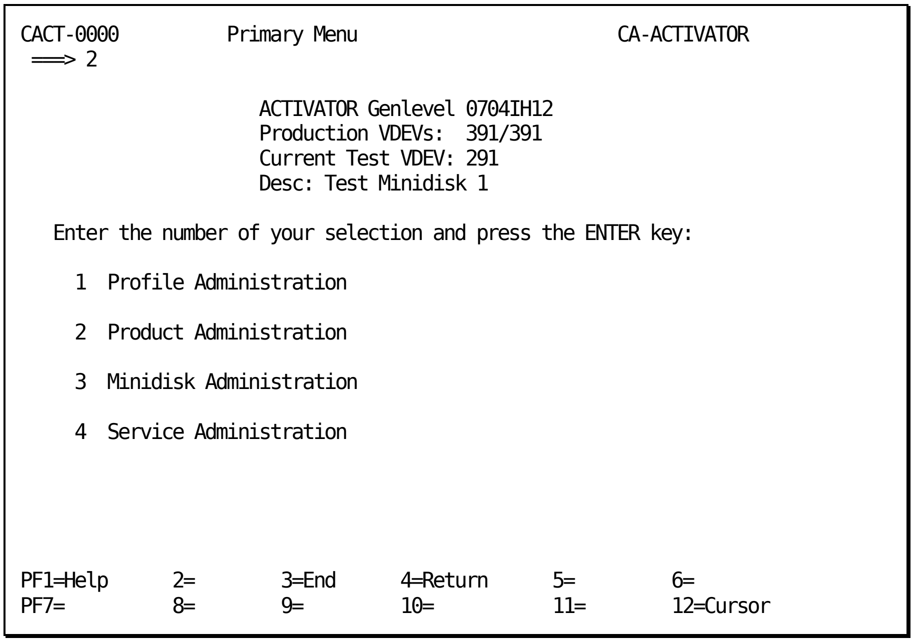
4. Enter **CACT** to start the online CA-ACTIVATOR program.

```
Ready; T=0.01/0.01 08:41:36
 cact
```

   The CAI logo panel appears:

```
CA+++++
   ----------------------------  CCCCCCCC       ---------
                                CC     CC  AAAAAAAAA
                               CC         AA      AA IIIIIIIII
             CA-ACTIVATOR VM  CC         AA      AA     II
              Release 1.2    CC     CC  AAAAAAAAA      II
                             CCCCCCCC  AA     AA      II
                                      AA     AA      II
                                                 IIIIIIII








  (C) Copyright 1987, 1990 by Computer Associates International,Inc.
   ------------------------------------------------------------
        Please press the ENTER key to continue
```

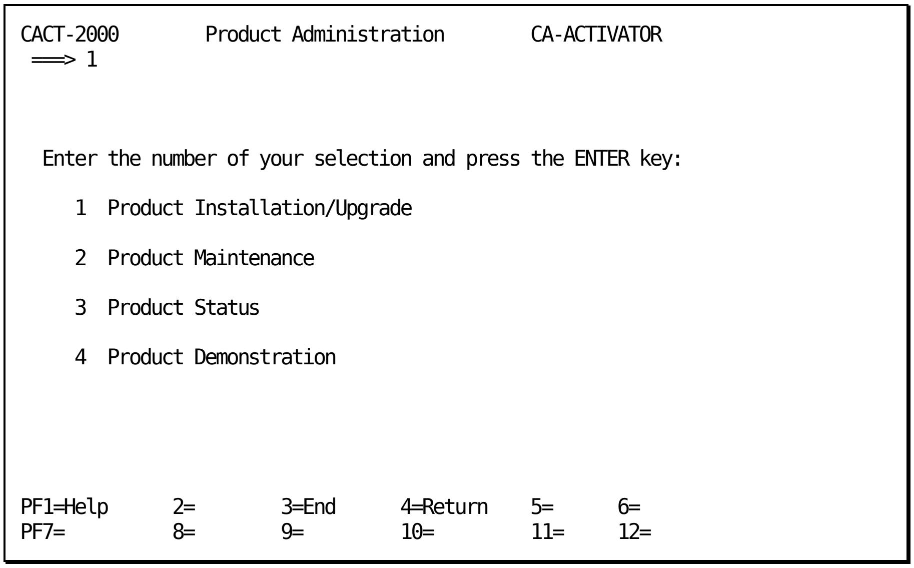
5. Press Enter.



```
CACT-0000         Primary Menu                        CA-ACTIVATOR
 ===> 2

                      ACTIVATOR Genlevel 0704IH12
                      Production VDEVs:  391/391
                      Current Test VDEV: 291
                      Desc: Test Minidisk 1

   Enter the number of your selection and press the ENTER key:

     1  Profile Administration

     2  Product Administration

     3  Minidisk Administration

     4  Service Administration




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

6. Enter 2 to install a CA product.

   The following panel appears:

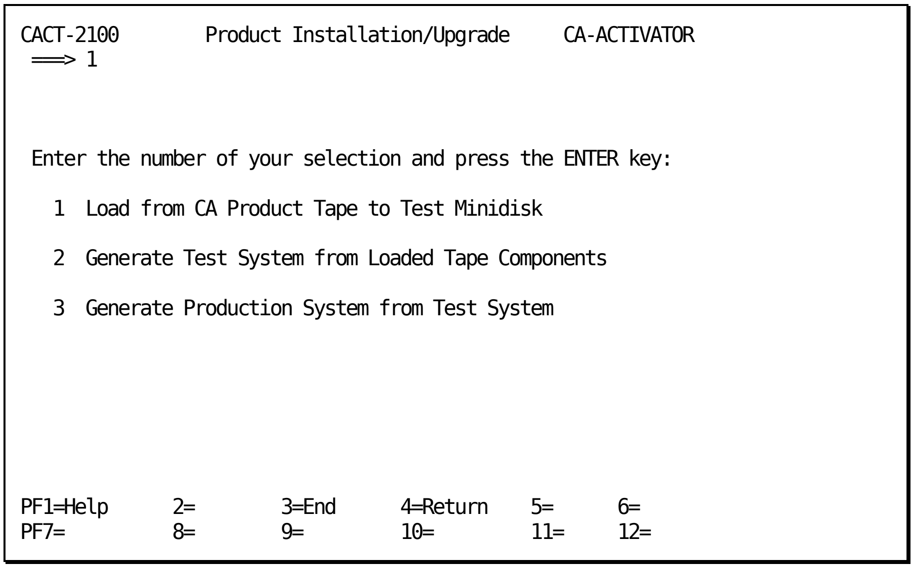


```
CACT-2000       Product Administration       CA-ACTIVATOR
 ===> 1



  Enter the number of your selection and press the ENTER key:

     1  Product Installation/Upgrade

     2  Product Maintenance

     3  Product Status

     4  Product Demonstration




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

7. Enter 1 to install a CA product.

   Note: For information on installing genlevel updates, see Applying Genlevel Updates.

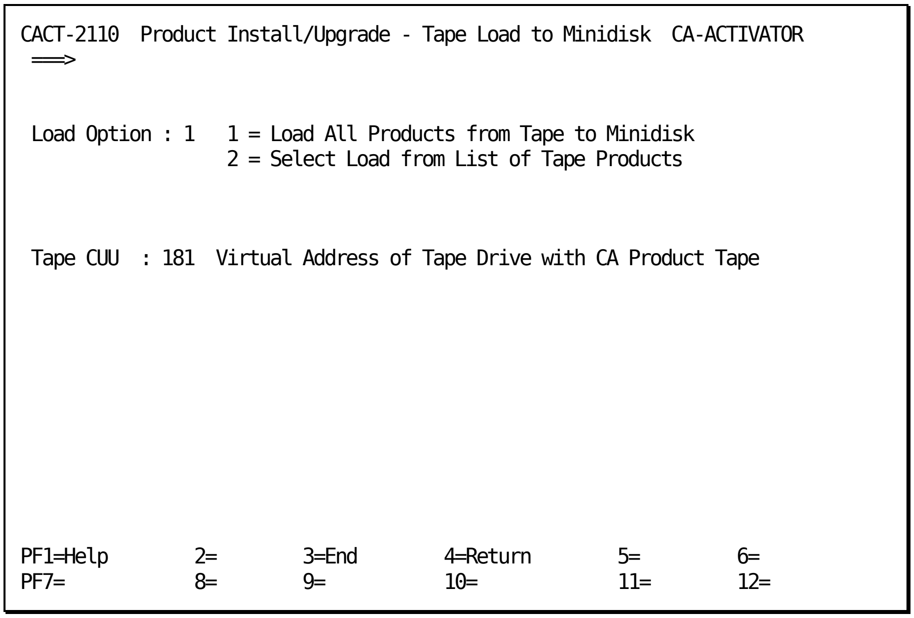
   The following panel appears:



```
CACT-2100       Product Installation/Upgrade     CA-ACTIVATOR
 ===> 1


 Enter the number of your selection and press the ENTER key:

   1  Load from CA Product Tape to Test Minidisk

   2  Generate Test System from Loaded Tape Components

   3  Generate Production System from Test System






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

8. Enter 1 to load CA products to your test minidisk.

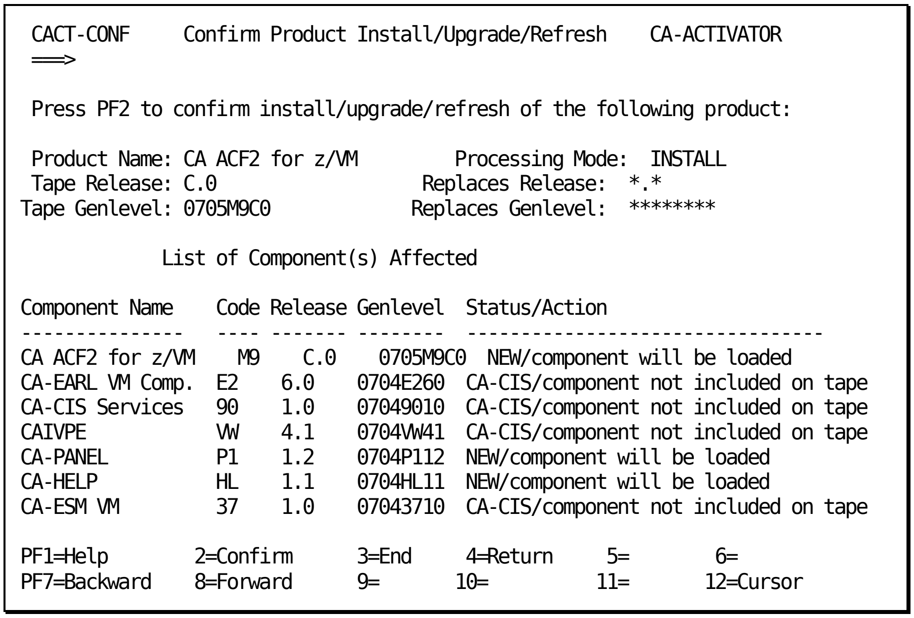
   The following panel appears:



```
CACT-2110  Product Install/Upgrade - Tape Load to Minidisk  CA-ACTIVATOR
 ===>


 Load Option : 1   1 = Load All Products from Tape to Minidisk
                   2 = Select Load from List of Tape Products



 Tape CUU  : 181  Virtual Address of Tape Drive with CA Product Tape









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

9. Enter 1 to load all the CA products to your test minidisk.

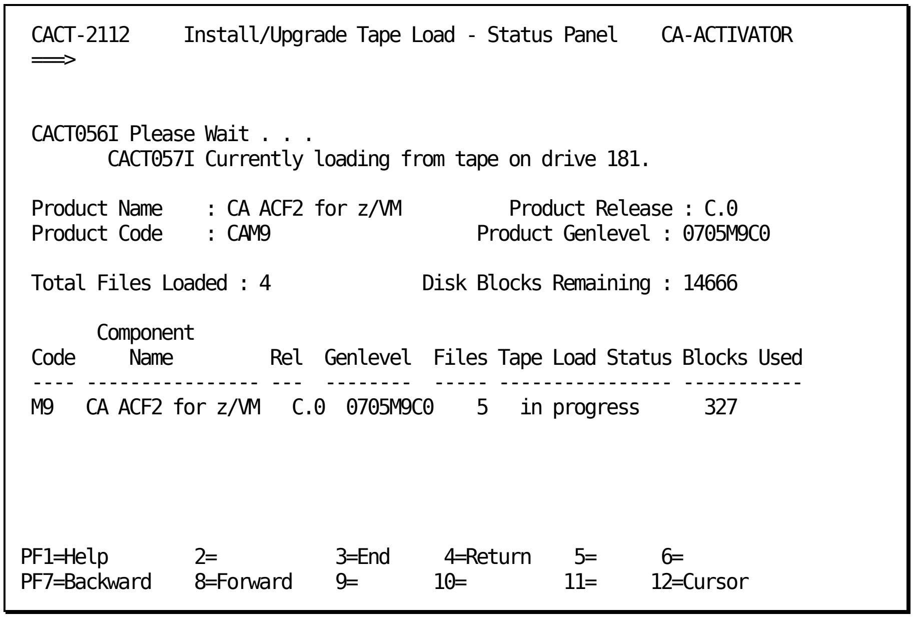
   The following panel appears confirming the CA products that CA-ACTIVATOR can load:



```
 CACT-CONF     Confirm Product Install/Upgrade/Refresh    CA-ACTIVATOR
 ===>

 Press PF2 to confirm install/upgrade/refresh of the following product:

 Product Name: CA ACF2 for z/VM        Processing Mode:  INSTALL
 Tape Release: C.0                   Replaces Release:  *.*
Tape Genlevel: 0705M9C0             Replaces Genlevel:  ********

             List of Component(s) Affected

Component Name    Code Release Genlevel  Status/Action
---------------   ---- ------- --------  ---------------------------------
CA ACF2 for z/VM    M9    C.0    0705M9C0  NEW/component will be loaded
CA-EARL VM Comp.  E2    6.0    0704E260  CA-CIS/component not included on tape
CA-CIS Services   90    1.0    07049010  CA-CIS/component not included on tape
CAIVPE            VW    4.1    0704VW41  CA-CIS/component not included on tape
CA-PANEL          P1    1.2    0704P112  NEW/component will be loaded
CA-HELP           HL    1.1    0704HL11  NEW/component will be loaded
CA-ESM VM         37    1.0    07043710  CA-CIS/component not included on tape

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

10. Press PF2 to confirm that you want to load these products.

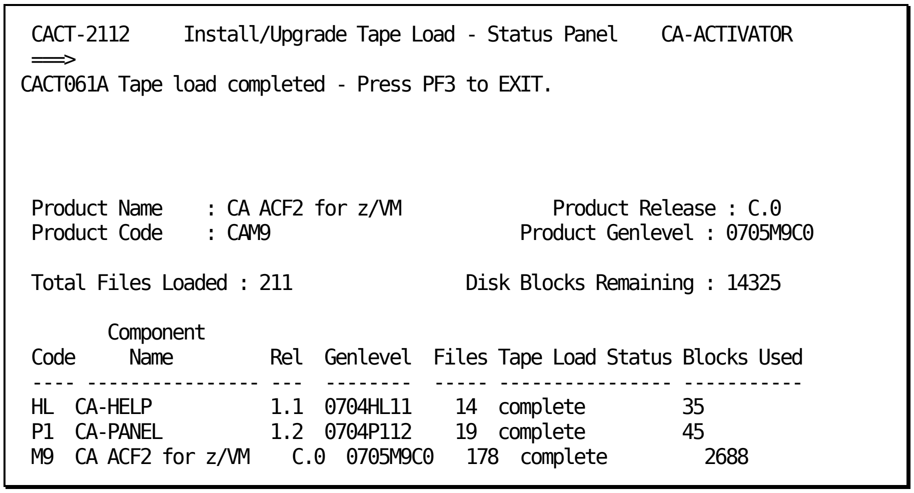
    The following panel appears:



```
 CACT-2112     Install/Upgrade Tape Load - Status Panel    CA-ACTIVATOR
 ===>


 CACT056I Please Wait . . .
        CACT057I Currently loading from tape on drive 181.

 Product Name    : CA ACF2 for z/VM         Product Release : C.0
 Product Code    : CAM9                  Product Genlevel : 0705M9C0

 Total Files Loaded : 4             Disk Blocks Remaining : 14666

       Component
 Code     Name         Rel  Genlevel  Files Tape Load Status Blocks Used
 ---- ---------------- ---  --------  ----- ---------------- -----------
 M9   CA ACF2 for z/VM   C.0  0705M9C0    5   in progress      327




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

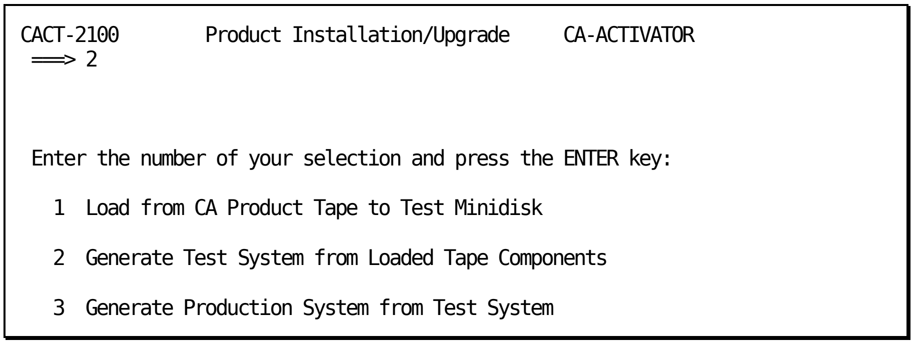
The panel changes as you watch the screen. It indicates how many files have been loaded so far.

When CA ACF2 for z/VM r12 is completely loaded, the panel appears as follows:



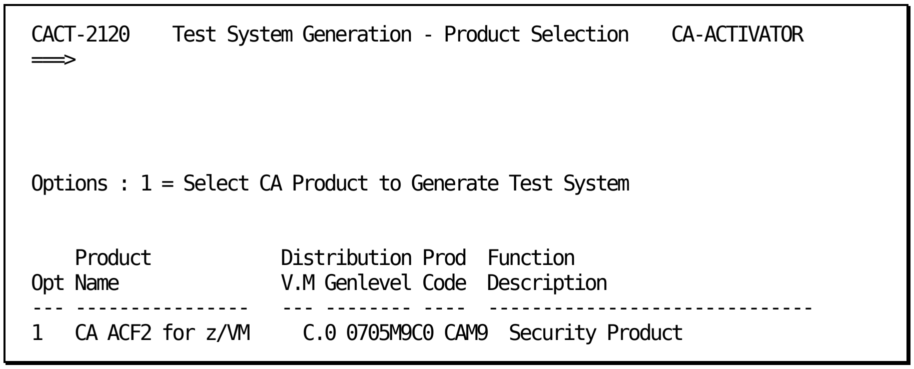
```
 CACT-2112     Install/Upgrade Tape Load - Status Panel    CA-ACTIVATOR
 ===>
CACT061A Tape load completed - Press PF3 to EXIT.



 Product Name    : CA ACF2 for z/VM             Product Release : C.0
 Product Code    : CAM9                      Product Genlevel : 0705M9C0

 Total Files Loaded : 211               Disk Blocks Remaining : 14325

        Component
 Code     Name         Rel  Genlevel  Files Tape Load Status Blocks Used
 ---- ---------------- ---  --------  ----- ---------------- -----------
 HL  CA-HELP          1.1  0704HL11    14  complete        35
 P1  CA-PANEL         1.2  0704P112    19  complete        45
 M9  CA ACF2 for z/VM   C.0  0705M9C0   178  complete        2688
```

When CA ACF2 for z/VM r12 is completely loaded, press PF3 to continue with the installation. CA-ACTIVATOR returns you to the following panel:



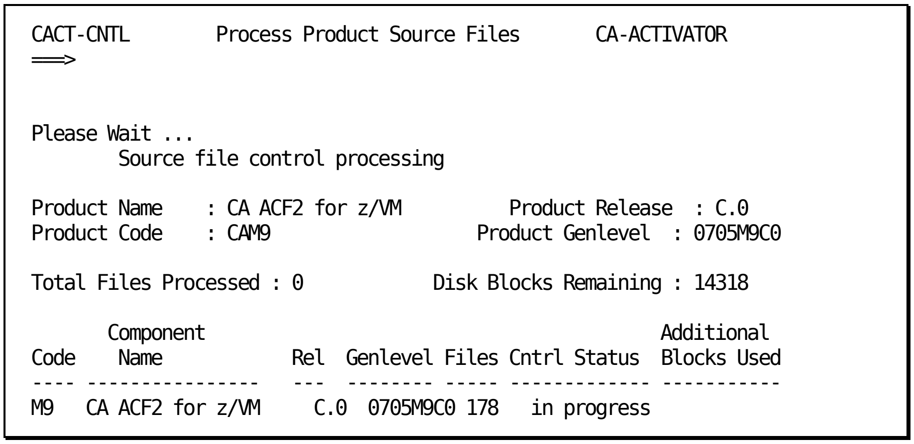
```
CACT-2100       Product Installation/Upgrade     CA-ACTIVATOR
 ===> 2



 Enter the number of your selection and press the ENTER key:

   1  Load from CA Product Tape to Test Minidisk

   2  Generate Test System from Loaded Tape Components

   3  Generate Production System from Test System
```

11. Select option 2 to generate the CA ACF2 for z/VM test system.
12. Type 1 in the Opt column next to the CA ACF2 for z/VM column to indicate you want to install CA ACF2 for z/VM on the test system.



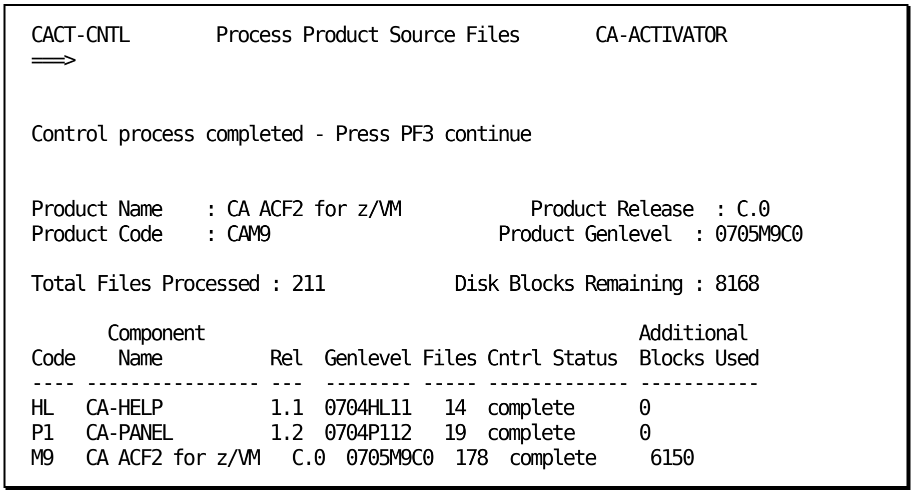
```
CACT-2120    Test System Generation - Product Selection    CA-ACTIVATOR
==>




Options : 1 = Select CA Product to Generate Test System


    Product            Distribution Prod  Function
Opt Name               V.M Genlevel Code  Description
--- ----------------   --- -------- ----  ------------------------------
1   CA ACF2 for z/VM     C.0 0705M9C0 CAM9  Security Product
```

The following panel appears:



```
CACT-CNTL       Process Product Source Files       CA-ACTIVATOR
===>


Please Wait ...
        Source file control processing

Product Name    : CA ACF2 for z/VM         Product Release  : C.0
Product Code    : CAM9                  Product Genlevel  : 0705M9C0

Total Files Processed : 0           Disk Blocks Remaining : 14318

       Component                                        Additional
Code    Name            Rel  Genlevel Files Cntrl Status  Blocks Used
---- ----------------   ---  -------- ----- ------------- -----------
M9   CA ACF2 for z/VM     C.0  0705M9C0 178   in progress
```

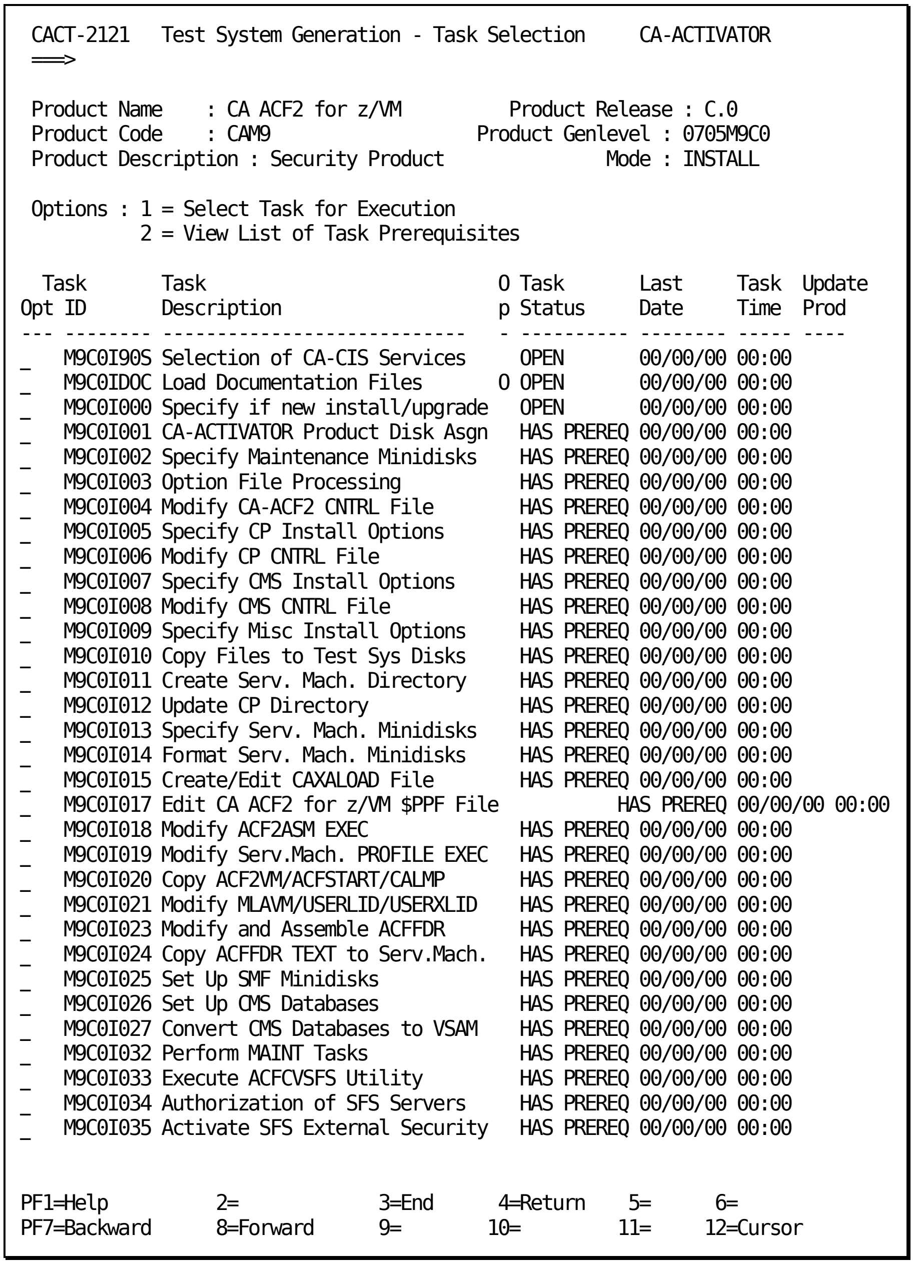
This processing can take quite a while to complete. When processing is complete, a message appears on the panel and CA-ACTIVATOR marks the Cntrl Status column as complete, as shown in the following panel:



```
CACT-CNTL       Process Product Source Files       CA-ACTIVATOR
===>


Control process completed - Press PF3 continue


Product Name    : CA ACF2 for z/VM           Product Release  : C.0
Product Code    : CAM9                    Product Genlevel  : 0705M9C0

Total Files Processed : 211           Disk Blocks Remaining : 8168

       Component                                      Additional
Code    Name          Rel  Genlevel Files Cntrl Status  Blocks Used
---- ---------------- ---  -------- ----- ------------- -----------
HL   CA-HELP          1.1  0704HL11   14  complete      0
P1   CA-PANEL         1.2  0704P112   19  complete      0
M9   CA ACF2 for z/VM   C.0  0705M9C0  178  complete     6150
```

13. Press PF3.

    The following panel appears:



```
 CACT-2121   Test System Generation - Task Selection     CA-ACTIVATOR
 ===>

 Product Name    : CA ACF2 for z/VM          Product Release : C.0
 Product Code    : CAM9                   Product Genlevel : 0705M9C0
 Product Description : Security Product               Mode : INSTALL

 Options : 1 = Select Task for Execution
           2 = View List of Task Prerequisites

  Task       Task                           O Task       Last     Task  Update
Opt ID       Description                    p Status     Last Date     Time  Prod
--- -------- ----------------------------   - ---------- -------- ----- ----
_   M9C0I90S Selection of CA-CIS Services     OPEN       00/00/00 00:00
_   M9C0IDOC Load Documentation Files       O OPEN       00/00/00 00:00
_   M9C0I000 Specify if new install/upgrade   OPEN       00/00/00 00:00
_   M9C0I001 CA-ACTIVATOR Product Disk Asgn   HAS PREREQ 00/00/00 00:00
_   M9C0I002 Specify Maintenance Minidisks    HAS PREREQ 00/00/00 00:00
_   M9C0I003 Option File Processing           HAS PREREQ 00/00/00 00:00
_   M9C0I004 Modify CA-ACF2 CNTRL File        HAS PREREQ 00/00/00 00:00
_   M9C0I005 Specify CP Install Options       HAS PREREQ 00/00/00 00:00
_   M9C0I006 Modify CP CNTRL File             HAS PREREQ 00/00/00 00:00
_   M9C0I007 Specify CMS Install Options      HAS PREREQ 00/00/00 00:00
_   M9C0I008 Modify CMS CNTRL File            HAS PREREQ 00/00/00 00:00
_   M9C0I009 Specify Misc Install Options     HAS PREREQ 00/00/00 00:00
_   M9C0I010 Copy Files to Test Sys Disks     HAS PREREQ 00/00/00 00:00
_   M9C0I011 Create Serv. Mach. Directory     HAS PREREQ 00/00/00 00:00
_   M9C0I012 Update CP Directory              HAS PREREQ 00/00/00 00:00
_   M9C0I013 Specify Serv. Mach. Minidisks    HAS PREREQ 00/00/00 00:00
_   M9C0I014 Format Serv. Mach. Minidisks     HAS PREREQ 00/00/00 00:00
_   M9C0I015 Create/Edit CAXALOAD File        HAS PREREQ 00/00/00 00:00
_   M9C0I017 Edit CA ACF2 for z/VM $PPF File           HAS PREREQ 00/00/00 00:00
_   M9C0I018 Modify ACF2ASM EXEC              HAS PREREQ 00/00/00 00:00
_   M9C0I019 Modify Serv.Mach. PROFILE EXEC   HAS PREREQ 00/00/00 00:00
_   M9C0I020 Copy ACF2VM/ACFSTART/CALMP       HAS PREREQ 00/00/00 00:00
_   M9C0I021 Modify MLAVM/USERLID/USERXLID    HAS PREREQ 00/00/00 00:00
_   M9C0I023 Modify and Assemble ACFFDR       HAS PREREQ 00/00/00 00:00
_   M9C0I024 Copy ACFFDR TEXT to Serv.Mach.   HAS PREREQ 00/00/00 00:00
_   M9C0I025 Set Up SMF Minidisks             HAS PREREQ 00/00/00 00:00
_   M9C0I026 Set Up CMS Databases             HAS PREREQ 00/00/00 00:00
_   M9C0I027 Convert CMS Databases to VSAM    HAS PREREQ 00/00/00 00:00
_   M9C0I032 Perform MAINT Tasks              HAS PREREQ 00/00/00 00:00
_   M9C0I033 Execute ACFCVSFS Utility         HAS PREREQ 00/00/00 00:00
_   M9C0I034 Authorization of SFS Servers     HAS PREREQ 00/00/00 00:00
_   M9C0I035 Activate SFS External Security   HAS PREREQ 00/00/00 00:00


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

Use the Task Selection Menu to select tasks. Enter 1 or 2 in the Opt field next to the task ID as follows:

1

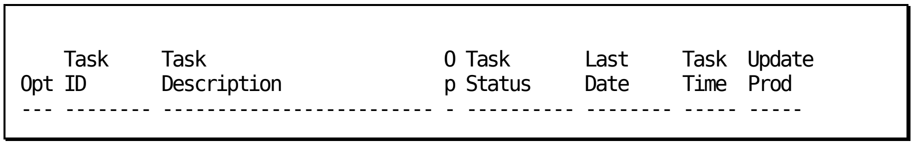
To select for execution task in OPEN or INCOMPLETE status

2

To view a list of task prerequisites

14. To begin, enter 1 in the Opt field to select OPEN tasks to complete in the order shown. When you install CA ACF2 for z/VM with CA-ACTIVATOR, the Task Selection Menu (CACT-2121) lists the tasks you need to complete the installation.

    There is a field to indicate whether the task is optional.



```
    Task     Task                     O Task       Last     Task  Update
Opt ID       Description              p Status     Date     Time  Prod
--- -------- ------------------------ - ---------- -------- ----- -----
```

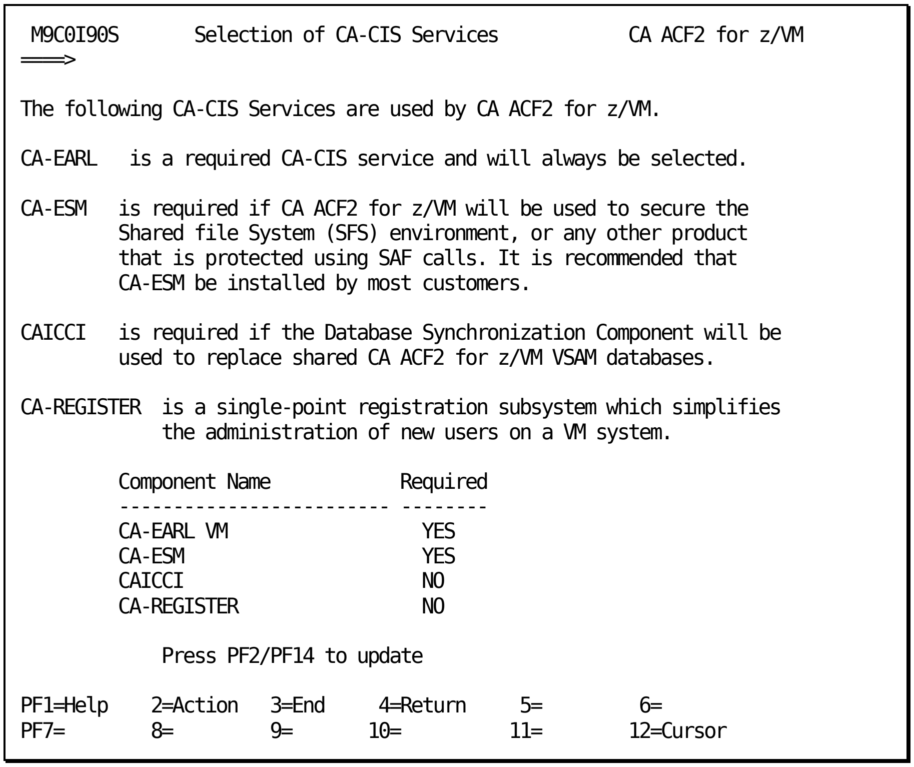
Op

An O in this field indicates the task is optional.

We explain the installation procedures to generate the test and production systems in the rest of this chapter. Choose the appropriate item from the Product Installation Menu to generate both the test and production systems.

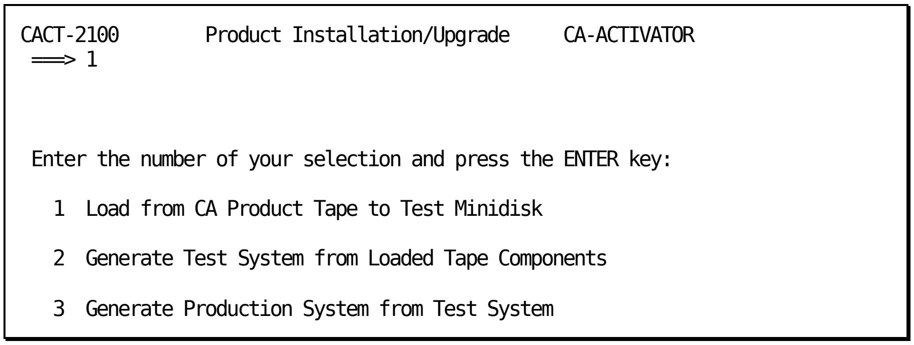
You should only type 1 **in the Opt column of M9C0I90S at this time. Press Enter.**

The M9C0I90S panel appears:



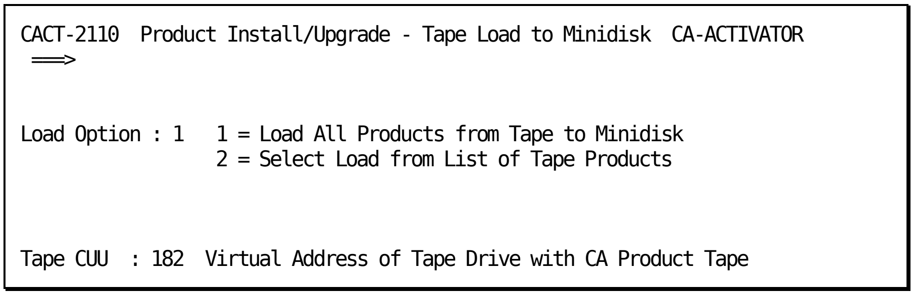
```
 M9C0I90S      Selection of CA-CIS Services           CA ACF2 for z/VM
===>

The following CA-CIS Services are used by CA ACF2 for z/VM.

CA-EARL   is a required CA-CIS service and will always be selected.

CA-ESM   is required if CA ACF2 for z/VM will be used to secure the
         Shared file System (SFS) environment, or any other product
         that is protected using SAF calls. It is recommended that
         CA-ESM be installed by most customers.

CAICCI   is required if the Database Synchronization Component will be
         used to replace shared CA ACF2 for z/VM VSAM databases.

CA-REGISTER  is a single-point registration subsystem which simplifies
             the administration of new users on a VM system.

         Component Name           Required
         ------------------------ --------
         CA-EARL VM                 YES
         CA-ESM                     YES
         CAICCI                     NO
         CA-REGISTER                NO

             Press PF2/PF14 to update

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

- Type YES to select a particular CA-CIS Service or NO to indicate you do not want to install a particular CA-CIS Service. The default is NO.
- Press PF2 or PF14 to perform the update.
- Press PF3 until you return to the Product Installation/Upgrade panel (CACT-2100):



```
CACT-2100       Product Installation/Upgrade     CA-ACTIVATOR
 ==> 1



 Enter the number of your selection and press the ENTER key:

   1  Load from CA Product Tape to Test Minidisk

   2  Generate Test System from Loaded Tape Components

   3  Generate Production System from Test System
```

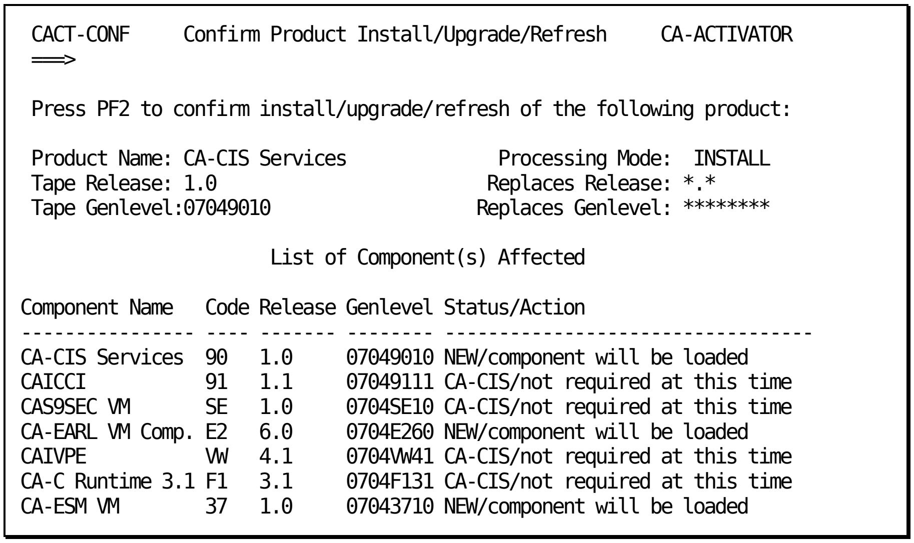
15. Enter 1 to return to the Product Install/Upgrade - Tape Load to Minidisk panel (CACT-2110).



```
CACT-2110  Product Install/Upgrade - Tape Load to Minidisk  CA-ACTIVATOR
 ==>


Load Option : 1   1 = Load All Products from Tape to Minidisk
                  2 = Select Load from List of Tape Products



Tape CUU  : 182  Virtual Address of Tape Drive with CA Product Tape
```

- Enter 182 in the Tape CUU field to point to the CA-CIS tape.
- Press Enter.

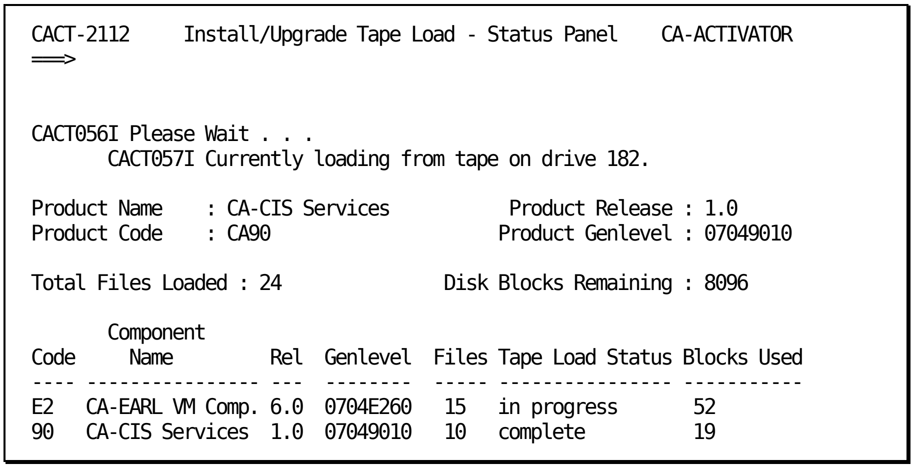
The following panel appears:



```
CACT-CONF     Confirm Product Install/Upgrade/Refresh     CA-ACTIVATOR
===>

Press PF2 to confirm install/upgrade/refresh of the following product:

Product Name: CA-CIS Services             Processing Mode:  INSTALL
Tape Release: 1.0                        Replaces Release: *.*
Tape Genlevel:07049010                  Replaces Genlevel: ********

                      List of Component(s) Affected

Component Name   Code Release Genlevel Status/Action
---------------- ---- ------- -------- ---------------------------------
CA-CIS Services  90   1.0     07049010 NEW/component will be loaded
CAICCI           91   1.1     07049111 CA-CIS/not required at this time
CAS9SEC VM       SE   1.0     0704SE10 CA-CIS/not required at this time
CA-EARL VM Comp. E2   6.0     0704E260 NEW/component will be loaded
CAIVPE           VW   4.1     0704VW41 CA-CIS/not required at this time
CA-C Runtime 3.1 F1   3.1     0704F131 CA-CIS/not required at this time
CA-ESM VM        37   1.0     07043710 NEW/component will be loaded
```

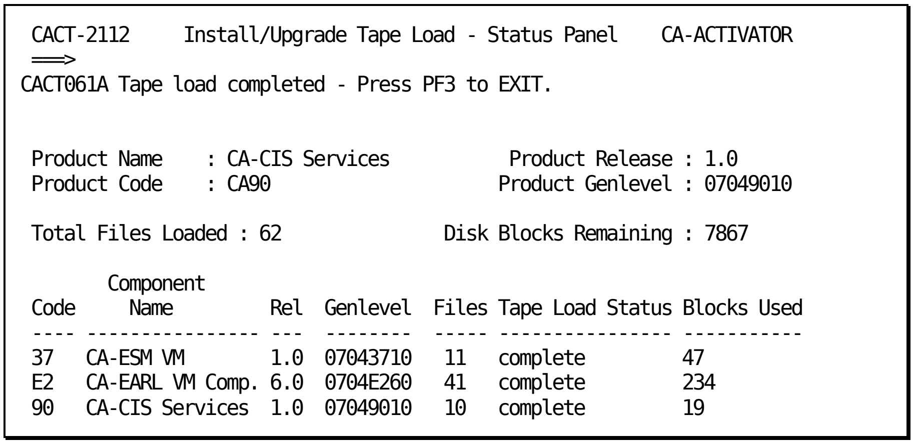
16. Press PF2 to confirm that you want to load these products.

    The following panel appears:



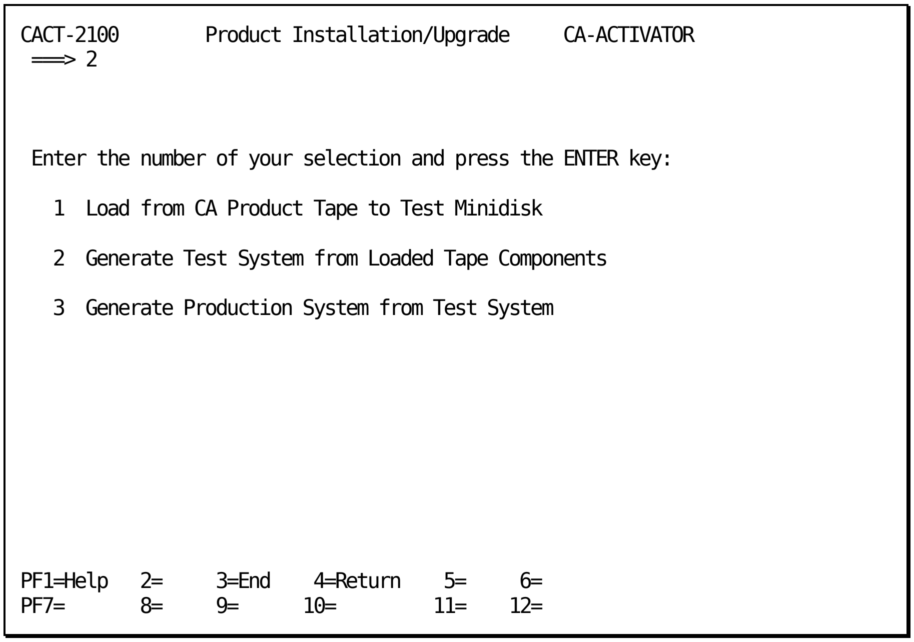
```
CACT-2112     Install/Upgrade Tape Load - Status Panel    CA-ACTIVATOR
===>


CACT056I Please Wait . . .
       CACT057I Currently loading from tape on drive 182.

Product Name    : CA-CIS Services           Product Release : 1.0
Product Code    : CA90                     Product Genlevel : 07049010

Total Files Loaded : 24             Disk Blocks Remaining : 8096

       Component
Code     Name         Rel  Genlevel  Files Tape Load Status Blocks Used
---- ---------------- ---  --------  ----- ---------------- -----------
E2   CA-EARL VM Comp. 6.0  0704E260   15   in progress       52
90   CA-CIS Services  1.0  07049010   10   complete          19
```

CA-ACTIVATOR updates this panel every time it completes a load of a CA-CIS component.



```
 CACT-2112     Install/Upgrade Tape Load - Status Panel    CA-ACTIVATOR
 ===>
CACT061A Tape load completed - Press PF3 to EXIT.


 Product Name    : CA-CIS Services          Product Release : 1.0
 Product Code    : CA90                    Product Genlevel : 07049010

 Total Files Loaded : 62              Disk Blocks Remaining : 7867

        Component
 Code     Name          Rel  Genlevel  Files Tape Load Status Blocks Used
 ---- ---------------- ---  --------  ----- ---------------- -----------
 37   CA-ESM VM        1.0  07043710   11   complete         47
 E2   CA-EARL VM Comp. 6.0  0704E260   41   complete         234
 90   CA-CIS Services  1.0  07049010   10   complete         19
```

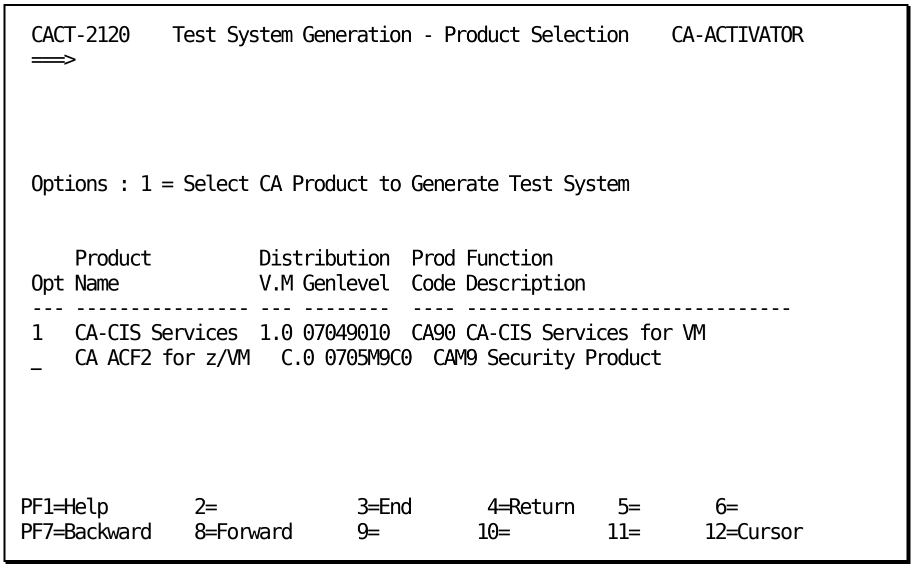
17. Press PF3 to return to the following panel:



```
CACT-2100        Product Installation/Upgrade     CA-ACTIVATOR
 ===> 2



 Enter the number of your selection and press the ENTER key:

   1  Load from CA Product Tape to Test Minidisk

   2  Generate Test System from Loaded Tape Components

   3  Generate Production System from Test System









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

18. Enter 2 to generate the test system.

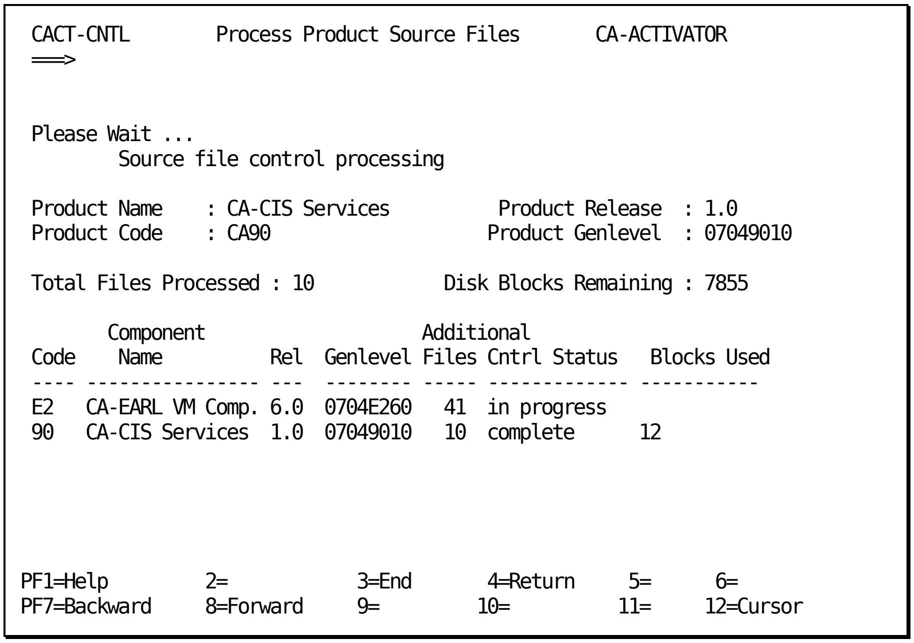
    The following panel appears:



```
CACT-2120    Test System Generation - Product Selection    CA-ACTIVATOR
===>



Options : 1 = Select CA Product to Generate Test System


    Product          Distribution  Prod Function
Opt Name             V.M Genlevel  Code Description
--- ---------------- --- --------  ---- ------------------------------
1   CA-CIS Services  1.0 07049010  CA90 CA-CIS Services for VM
_   CA ACF2 for z/VM   C.0 0705M9C0  CAM9 Security Product




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

19. Type 1 in the Opt column next to the CA-CIS Services to indicate you want to install CA-CIS Services on the test system. Press Enter.

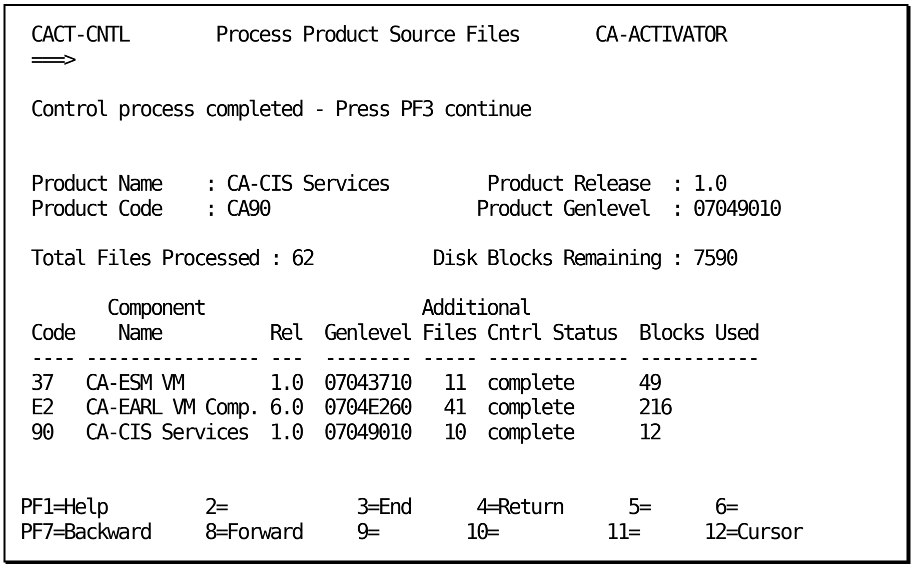
    If this is the first time you selected the product after loading the install tape, the panel below appears. Otherwise, processing continues with the next step.



```
CACT-CNTL       Process Product Source Files       CA-ACTIVATOR
===>


Please Wait ...
        Source file control processing

Product Name    : CA-CIS Services         Product Release  : 1.0
Product Code    : CA90                   Product Genlevel  : 07049010

Total Files Processed : 10           Disk Blocks Remaining : 7855

       Component                     Additional
Code    Name          Rel  Genlevel Files Cntrl Status   Blocks Used
---- ---------------- ---  -------- ----- ------------- -----------
E2   CA-EARL VM Comp. 6.0  0704E260   41  in progress
90   CA-CIS Services  1.0  07049010   10  complete      12




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

The above panel indicates that the CA-CIS Services files have been processed, and that CA-ACTIVATOR is currently processing the CA-EARL VM files.

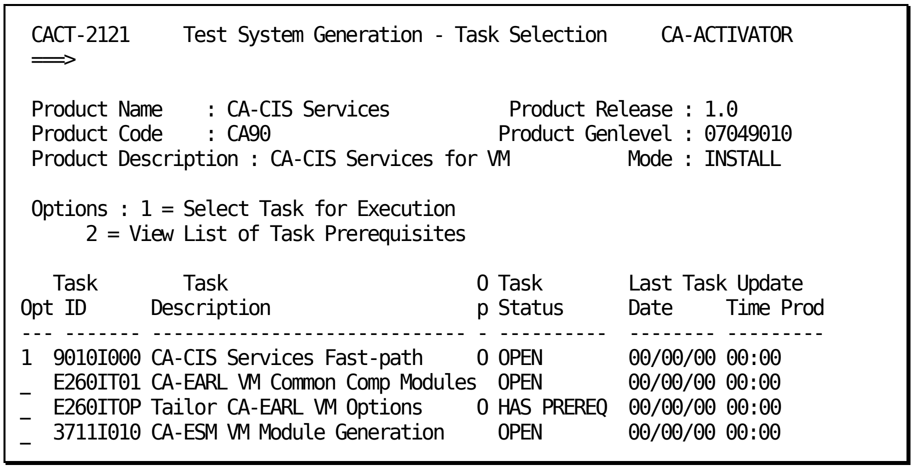
When processing is complete, the following panel appears:



```
 CACT-CNTL       Process Product Source Files       CA-ACTIVATOR
 ===>

 Control process completed - Press PF3 continue


 Product Name    : CA-CIS Services         Product Release  : 1.0
 Product Code    : CA90                   Product Genlevel  : 07049010

 Total Files Processed : 62          Disk Blocks Remaining : 7590

        Component                     Additional
 Code    Name          Rel  Genlevel Files Cntrl Status  Blocks Used
 ---- ---------------- ---  -------- ----- ------------- -----------
 37   CA-ESM VM        1.0  07043710   11  complete      49
 E2   CA-EARL VM Comp. 6.0  0704E260   41  complete      216
 90   CA-CIS Services  1.0  07049010   10  complete      12


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

20. Press PF3 to continue with the installation.

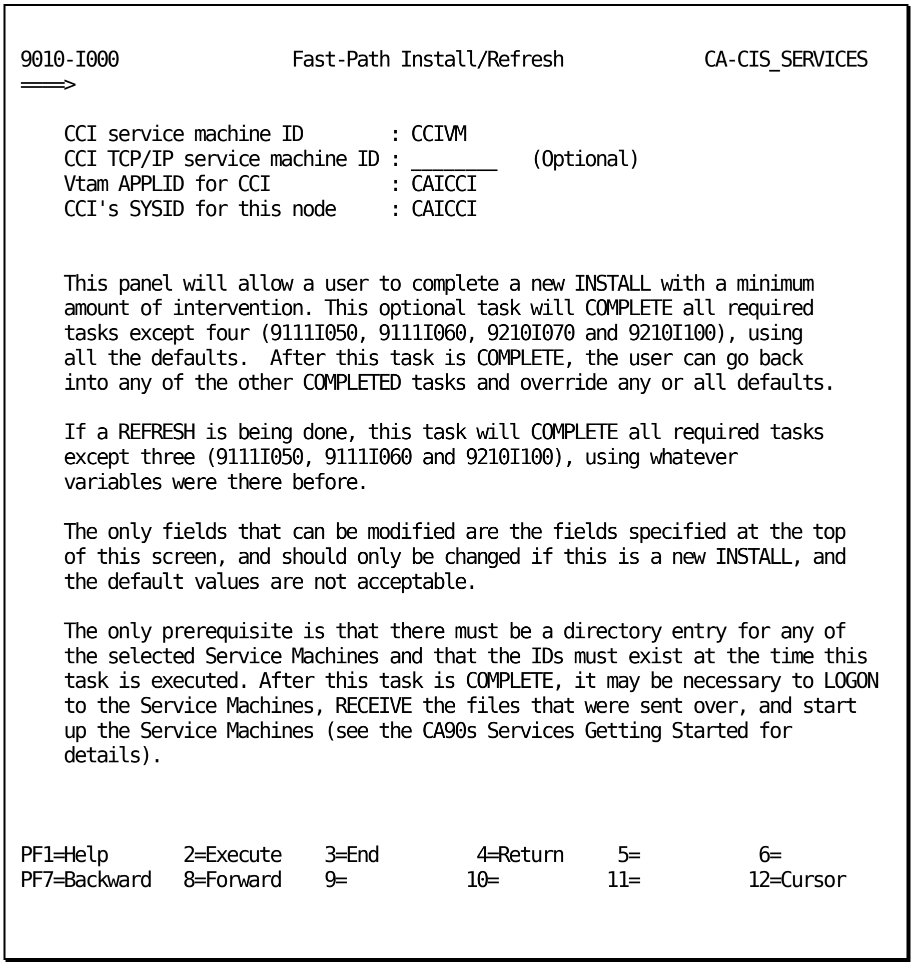
    The following panel appears:



```
 CACT-2121     Test System Generation - Task Selection     CA-ACTIVATOR
 ===>

 Product Name    : CA-CIS Services           Product Release : 1.0
 Product Code    : CA90                     Product Genlevel : 07049010
 Product Description : CA-CIS Services for VM          Mode : INSTALL

 Options : 1 = Select Task for Execution
      2 = View List of Task Prerequisites

   Task       Task                         O Task       Last Task Update
Opt ID      Description                    p Status      Date     Time Prod
--- ------- ---------------------------- - ----------  -------- ---------
1  9010I000 CA-CIS Services Fast-path     O OPEN        00/00/00 00:00
_  E260IT01 CA-EARL VM Common Comp Modules  OPEN        00/00/00 00:00
_  E260ITOP Tailor CA-EARL VM Options     O HAS PREREQ  00/00/00 00:00
_  3711I010 CA-ESM VM Module Generation     OPEN        00/00/00 00:00
```

21. Type 1 in the Opt column to indicate which tasks you want to perform.

    If you choose to install the CA-CIS Services Fast-path, the following panel appears:

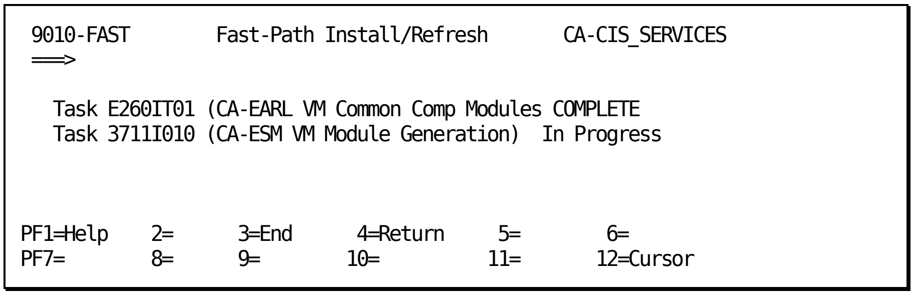


```
9010-I000                Fast-Path Install/Refresh           CA-CIS_SERVICES
===>

    CCI service machine ID        : CCIVM
    CCI TCP/IP service machine ID : ________   (Optional)
    Vtam APPLID for CCI           : CAICCI
    CCI's SYSID for this node     : CAICCI


    This panel will allow a user to complete a new INSTALL with a minimum
    amount of intervention. This optional task will COMPLETE all required
    tasks except four (9111I050, 9111I060, 9210I070 and 9210I100), using
    all the defaults.  After this task is COMPLETE, the user can go back
    into any of the other COMPLETED tasks and override any or all defaults.

    If a REFRESH is being done, this task will COMPLETE all required tasks
    except three (9111I050, 9111I060 and 9210I100), using whatever
    variables were there before.

    The only fields that can be modified are the fields specified at the top
    of this screen, and should only be changed if this is a new INSTALL, and
    the default values are not acceptable.

    The only prerequisite is that there must be a directory entry for any of
    the selected Service Machines and that the IDs must exist at the time this
    task is executed. After this task is COMPLETE, it may be necessary to LOGON
    to the Service Machines, RECEIVE the files that were sent over, and start
    up the Service Machines (see the CA90s Services Getting Started for
    details).



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

Note: If you are installing CCI, but you do not have a CCI TCP/IP Service Machine (not used by CA ACF2 for z/VM) you may need to blank out this field.

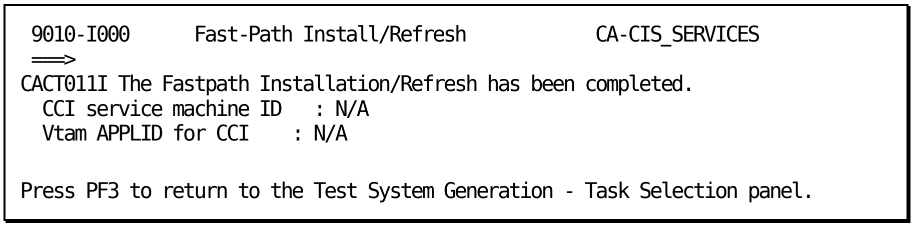
22. Press PF2.

    The next panel displays the status of the modules. In the example below, the CA-EARL VM common component modules have completed, and the CA-ESM VM module generation is in progress.



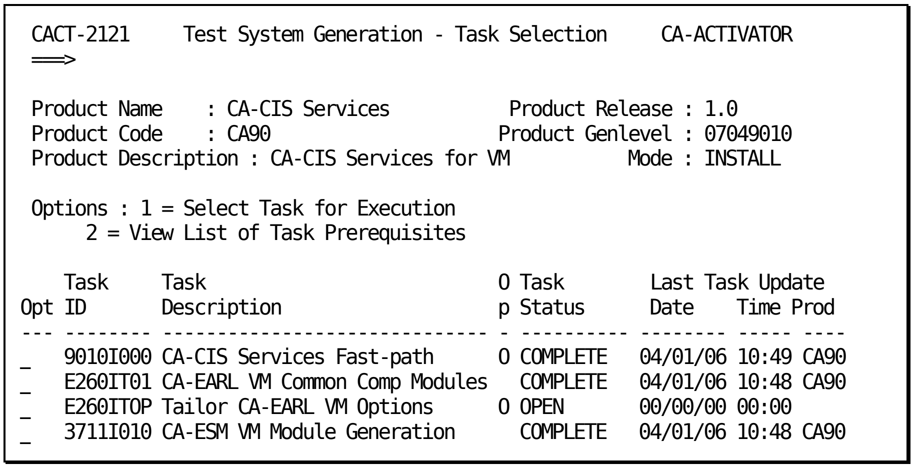
```
 9010-FAST       Fast-Path Install/Refresh       CA-CIS_SERVICES
 ===>

   Task E260IT01 (CA-EARL VM Common Comp Modules COMPLETE
   Task 3711I010 (CA-ESM VM Module Generation)  In Progress



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

When all the modules are generated, a message displays on the Fast-Path Install/Refresh panel:



```
 9010-I000      Fast-Path Install/Refresh            CA-CIS_SERVICES
 ===>
CACT011I The Fastpath Installation/Refresh has been completed.
  CCI service machine ID   : N/A
  Vtam APPLID for CCI    : N/A

Press PF3 to return to the Test System Generation - Task Selection panel.
```

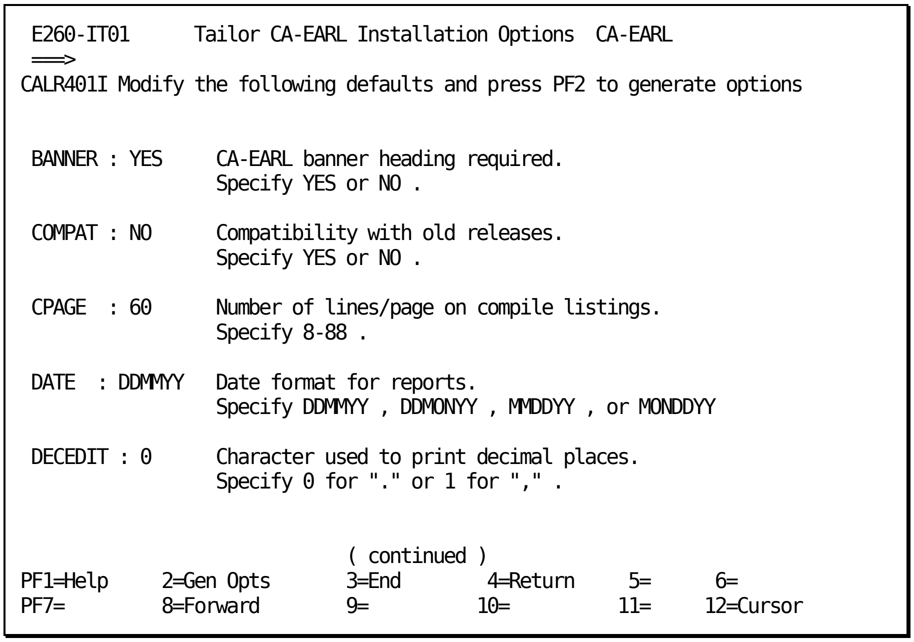
The panel displays:



```
 CACT-2121     Test System Generation - Task Selection     CA-ACTIVATOR
 ===>

 Product Name    : CA-CIS Services            Product Release : 1.0
 Product Code    : CA90                      Product Genlevel : 07049010
 Product Description : CA-CIS Services for VM           Mode : INSTALL

 Options : 1 = Select Task for Execution
      2 = View List of Task Prerequisites

    Task     Task                           O Task        Last Task Update
Opt ID       Description                    p Status      Date     Time Prod
--- -------- ------------------------------ - ---------- -------- ----- ----
_   9010I000 CA-CIS Services Fast-path      O COMPLETE   04/01/06 10:49 CA90
_   E260IT01 CA-EARL VM Common Comp Modules   COMPLETE   04/01/06 10:48 CA90
_   E260ITOP Tailor CA-EARL VM Options      O OPEN       00/00/00 00:00
_   3711I010 CA-ESM VM Module Generation      COMPLETE   04/01/06 10:48 CA90
```

23. To tailor the CA-Earl VM options, enter 1 next to task E260ITOP (as shown on the above panel).

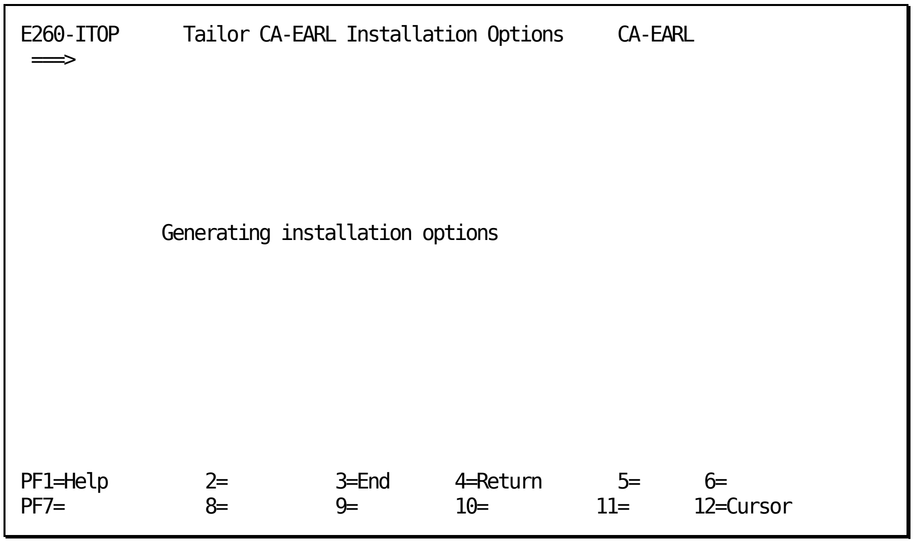
    The following panel displays. Make any changes applicable to your site.



```
 E260-IT01     Tailor CA-EARL Installation Options  CA-EARL
 ===>
CALR401I Modify the following defaults and press PF2 to generate options


 BANNER : YES     CA-EARL banner heading required.
                  Specify YES or NO .

 COMPAT : NO      Compatibility with old releases.
                  Specify YES or NO .

 CPAGE  : 60      Number of lines/page on compile listings.
                  Specify 8-88 .

 DATE  : DDMMYY   Date format for reports.
                  Specify DDMMYY , DDMONYY , MMDDYY , or MONDDYY

 DECEDIT : 0      Character used to print decimal places.
                  Specify 0 for "." or 1 for "," .


                              ( continued )
PF1=Help     2=Gen Opts       3=End        4=Return     5=      6=
PF7=         8=Forward        9=          10=          11=     12=Cursor
```

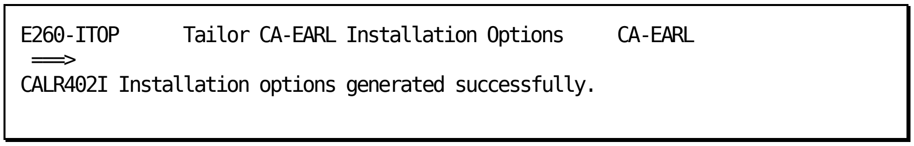
24. Press PF8 to continue to the next section of the Tailor CA-EARL Installation Options panel. After you have selected the installation options, press PF2 to generate them.

    The following panel appears:



```
E260-ITOP      Tailor CA-EARL Installation Options     CA-EARL
 ===>




             Generating installation options








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

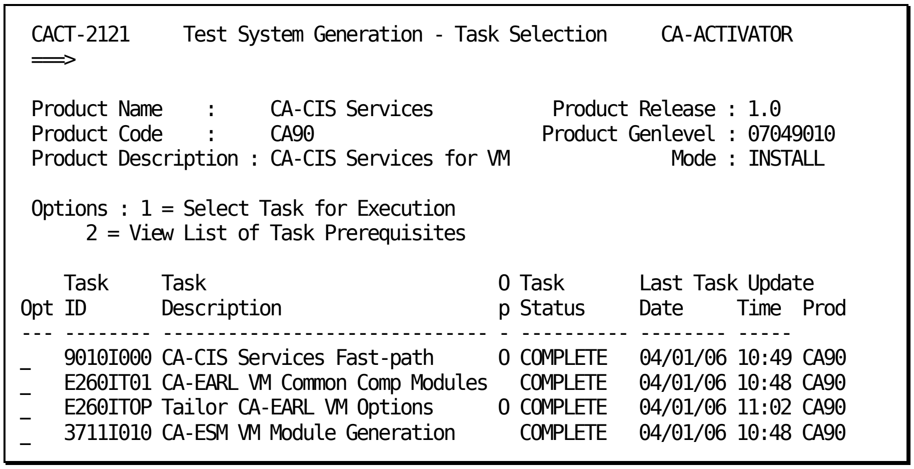
When processing is done, a message appears on the panel:



```
E260-ITOP      Tailor CA-EARL Installation Options     CA-EARL
 ===>
CALR402I Installation options generated successfully.
```

25. Press PF3

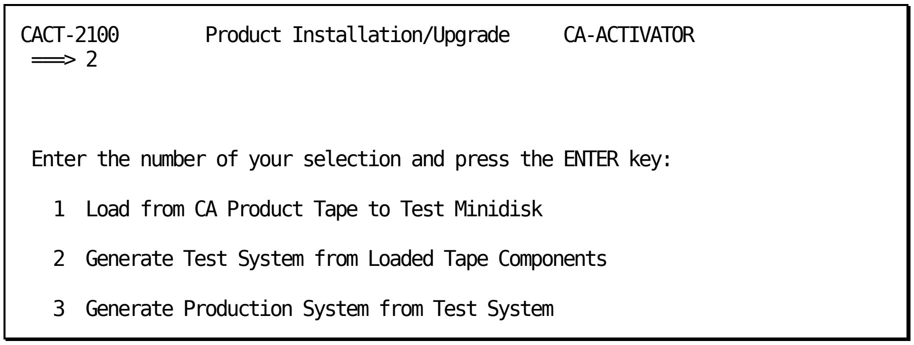
    The following panel displays:



```
 CACT-2121     Test System Generation - Task Selection     CA-ACTIVATOR
 ===>

 Product Name    :     CA-CIS Services          Product Release : 1.0
 Product Code    :     CA90                    Product Genlevel : 07049010
 Product Description : CA-CIS Services for VM              Mode : INSTALL

 Options : 1 = Select Task for Execution
      2 = View List of Task Prerequisites

    Task     Task                           O Task       Last Task Update
Opt ID       Description                    p Status     Date     Time  Prod
--- -------- ------------------------------ - ---------- -------- -----
_   9010I000 CA-CIS Services Fast-path      O COMPLETE   04/01/06 10:49 CA90
_   E260IT01 CA-EARL VM Common Comp Modules   COMPLETE   04/01/06 10:48 CA90
_   E260ITOP Tailor CA-EARL VM Options      O COMPLETE   04/01/06 11:02 CA90
_   3711I010 CA-ESM VM Module Generation      COMPLETE   04/01/06 10:48 CA90
```

26. Press PF3 until you return to the Product Installation/Upgrade panel (CACT-2100).

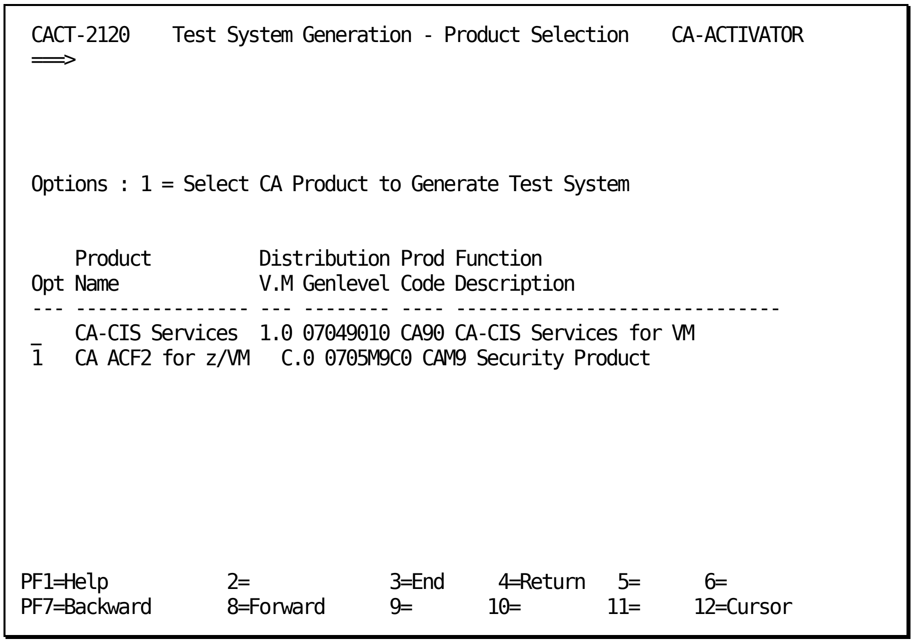
    The panel appears:



```
CACT-2100       Product Installation/Upgrade     CA-ACTIVATOR
 ===> 2



 Enter the number of your selection and press the ENTER key:

   1  Load from CA Product Tape to Test Minidisk

   2  Generate Test System from Loaded Tape Components

   3  Generate Production System from Test System
```

27. Enter 2.

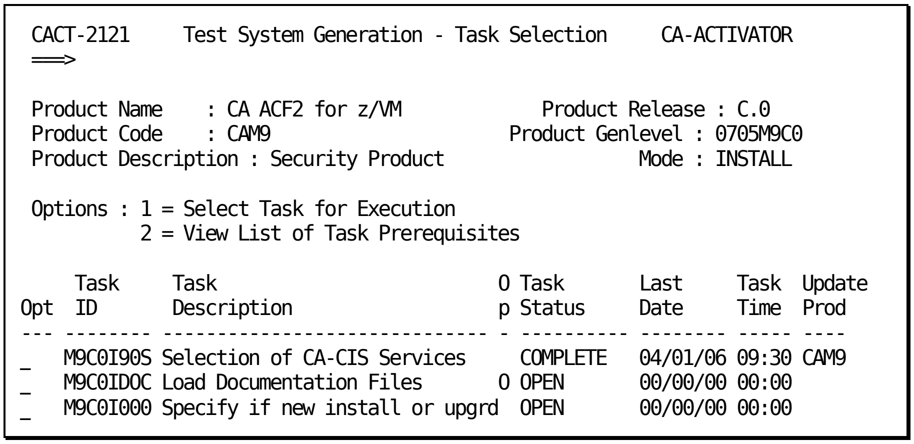
    The following panel appears:



```
 CACT-2120    Test System Generation - Product Selection    CA-ACTIVATOR
 ===>




 Options : 1 = Select CA Product to Generate Test System


     Product          Distribution Prod Function
 Opt Name             V.M Genlevel Code Description
 --- ---------------- --- -------- ---- ------------------------------
 _   CA-CIS Services  1.0 07049010 CA90 CA-CIS Services for VM
 1   CA ACF2 for z/VM   C.0 0705M9C0 CAM9 Security Product









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

28. Type 1 next to CA ACF2 for z/VM and press Enter.

    The Test System Generation - Task Selection panel (CACT-2121) appears:



```
  CACT-2121     Test System Generation - Task Selection     CA-ACTIVATOR
  ===>

  Product Name    : CA ACF2 for z/VM           Product Release : C.0
  Product Code    : CAM9                     Product Genlevel : 0705M9C0
  Product Description : Security Product                 Mode : INSTALL

  Options : 1 = Select Task for Execution
            2 = View List of Task Prerequisites

      Task     Task                          O Task       Last     Task  Update
 Opt  ID       Description                   p Status     Date     Time  Prod
 --- -------- ----------------------------- - ---------- -------- ----- ----
 _   M9C0I90S Selection of CA-CIS Services    COMPLETE   04/01/06 09:30 CAM9
 _   M9C0IDOC Load Documentation Files      O OPEN       00/00/00 00:00
 _   M9C0I000 Specify if new install or upgrd OPEN       00/00/00 00:00
```

29. To complete the CA ACF2 for z/VM installation, resume executing the tasks, beginning with task M9C0IDOC.

CA ACF2 for z/VM Installation Task Selection Menu

For both systems (test and production), CA-ACTIVATOR displays a list of tasks on the CA-ACTIVATOR Installation Task Selection Menu (CACT-2121). The Installation Task Selection Menu includes product information, including name, code, description, and genlevel. Task information and the status of the current task appear at the bottom of the Task Selection Menu. Task information includes the task ID, description, status, and the last used or modified date and time. It also displays the product ID. Possible status descriptions are:

OPEN

Task is ready to execute.

COMPLETE

Task successfully executed.

INCOMPLETE

Part of the task did not successfully execute.

HAS PREREQ

Cannot open this task until a prerequisite task executes successfully.

O - OPTIONAL

Execution of task is optional, depending on previous task selection.

Procedure Conventions

The next few sections describe the procedural conventions in the installation tasks for the test and production systems. Read these conventions to become familiar with the CA-ACTIVATOR panels.

Test System Installation Tasks

You must follow the steps in this section to install CA ACF2 for z/VM on your test system. Refer to the *CA-CIS CA-ACTIVATOR Reference Guide* if you need additional information on starting CA-ACTIVATOR. Following is a list of the installation tasks for the CA ACF2 for z/VM test system. Refer to the individual tasks in this chapter for detailed information.

Task #	Task Title	Description
M9C0190S	Selection of CA-CIS Services	Installs the CA-CIS Services that CA ACF2 for z/VM requires.
M9C0IDOC	Load Documentation Files	Lets you load the documentation files from the tape and place them on a specified mindisk.
M9C0I000	Specify if new install or upgrade	Indicates if you are installing CA ACF2 for z/VM for the first time or are upgrading an existing installation.
M9C0I001	CA-ACTIVATOR Product Disk Assignment	Defines the minidisks CA ACF2 for z/VM uses on your system maintenance ID.
M9C0I002	Specify maintenance minidisks	Specifies the system MAINT virtual machine disks and other disks required for certain installation activities.
M9C0I003	Option file processing	Specifies the CA ACF2 for z/VM option file used for this installation.
M9C0I004	Modify CA ACF2 for z/VM CNTRL file	Lets you modify the CNTRL file for correct CA ACF2 for z/VM file updating and assembly.
M9C0I005	Specify CP installation option	Specifies the current CP release at your site.
M9C0I006	Modify CP CNTRL file	Lets you create the CP CNTRL file that CA ACF2 for z/VM uses to assemble the HCPAC0 macro.

Task #	Task Title	Description
M9C0I007	Specify CMS installation options	Specifies the current CMS release at your site.
M9C0I008	Modify CMS CNTRL file	Lets you create your CMS CNTRL file.
M9C0I009	Specify miscellaneous installation options	Specifies whether you want to install full-screen, ACF help, full-screen help, and message help. Also specifies whether you are using VSAM shared database support and whether this is a new install or a migration from an earlier CA ACF2 for z/VM release.
M9C0I010	Copy files to test system disks	Copies the files necessary to generate your system to the test system disks.
M9C0I011	Create CA-ACF2 service machine directory entry	Copies the directory entry for the CA ACF2 for z/VM service machine.
M9C0I012	Update CP directory	Applies updates to the CP directory.
M9C0I013	Specify service machine minidisk links	Specifies the service machine minidisks you want to access.
M9C0I014	Format the service machine minidisks	Formats the minidisks that CA ACF2 for z/VM uses.
M9C0I015	Create/edit ACFCP CAXALOAD file	Lets you create or edit the ACFCP CAXALOAD file.
M9C0I017	Edit CA ACF2 for z/VM $PPF file	Lets you modify the $PPF file.
M9C0I018	Modify ACF2ASM EXEC	Lets you modify the ACF2ASM EXEC, used for all assemblies.
M9C0I019	Modify the service machine PROFILE EXEC	Lets you create or modify the PROFILE EXEC.
M9C0I020	Copy ACF2VM MODULE, ACFSTART EXEC, and CALMP KEYS	Lets you copy the ACF2VM MODULE, ACFSTART EXEC, and CALMP KEYS files. You also can modify the ACFSTART EXEC and CALMP KEYS files, if necessary.
M9C0I021	Modify MLAVM, USERLID COPY, and USERXLID COPY	Builds the ACF2USER MACLIB.

Task #	Task Title	Description
M9C0I023	Modify and assemble ACFFDR	Lets you modify and assemble the ACFFDR and copies the new ACFFDR to the CA ACF2 for z/VM service machine minidisk.
M9C0I024	Copy ACFFDR TEXT to service machine	Copies the ACFFDR TEXT file from the local options disk to the service machine disk.
M9C0I025	Set up SMF minidisks	Sets up SMF minidisks for SMF recording.
M9C0I026	Set up CMS databases	Loads CA ACF2 for z/VM CMS databases or converts existing CA ACF2 for z/VM CMS databases to proper format.
M9C0I027	Convert CMS databases to VSAM databases	Uses the ACF2VSAM EXEC to convert CA ACF2 for z/VM CMS databases to VSAM databases.
M9C0I032	Perform MAINT tasks	Processes the ACF2TASK EXEC.
M9C0I033	Execute ACFCVSFS Utility	Converts existing SFS grants.
M9C0I034	Authorization of SFS Service Machines	Prompts you to establish IUCV communication authorization with *RPI CP system service.
M9C0I035	Activate SFS External Security	Prompts you to activate SFS external security.

Task M9C0I90S: Selection of CA-CIS Services

This task selects the CA-CIS Services that CA ACF2 for z/VM requires. You have already completed this task. Proceed to task M9C0IDOC.

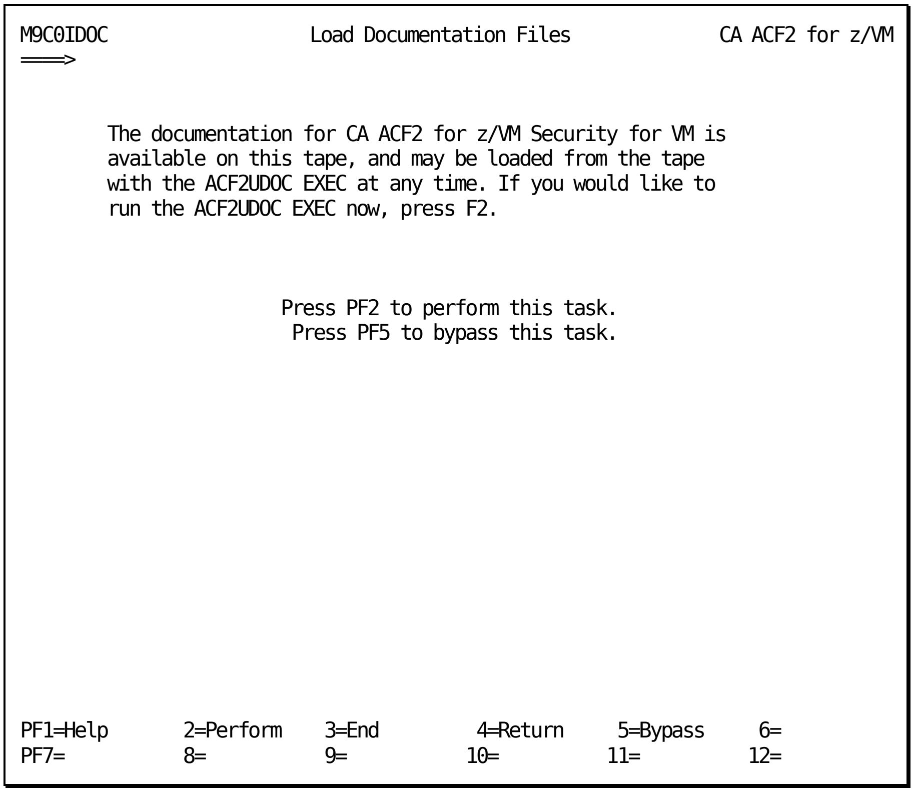
Task M9C0IDOC: Load Documentation Files

This task lets you load the documentation files from the tape and place them on a specified minidisk. This task is optional.

To complete this task:

1. Select panel M9C0IDOC from the Task Selection Menu.

   The following panel appears:



```
M9C0IDOC                  Load Documentation Files              CA ACF2 for z/VM
===>


        The documentation for CA ACF2 for z/VM Security for VM is
        available on this tape, and may be loaded from the tape
        with the ACF2UDOC EXEC at any time. If you would like to
        run the ACF2UDOC EXEC now, press F2.



                        Press PF2 to perform this task.
                         Press PF5 to bypass this task.

















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

2. Press PF2 to perform the task.

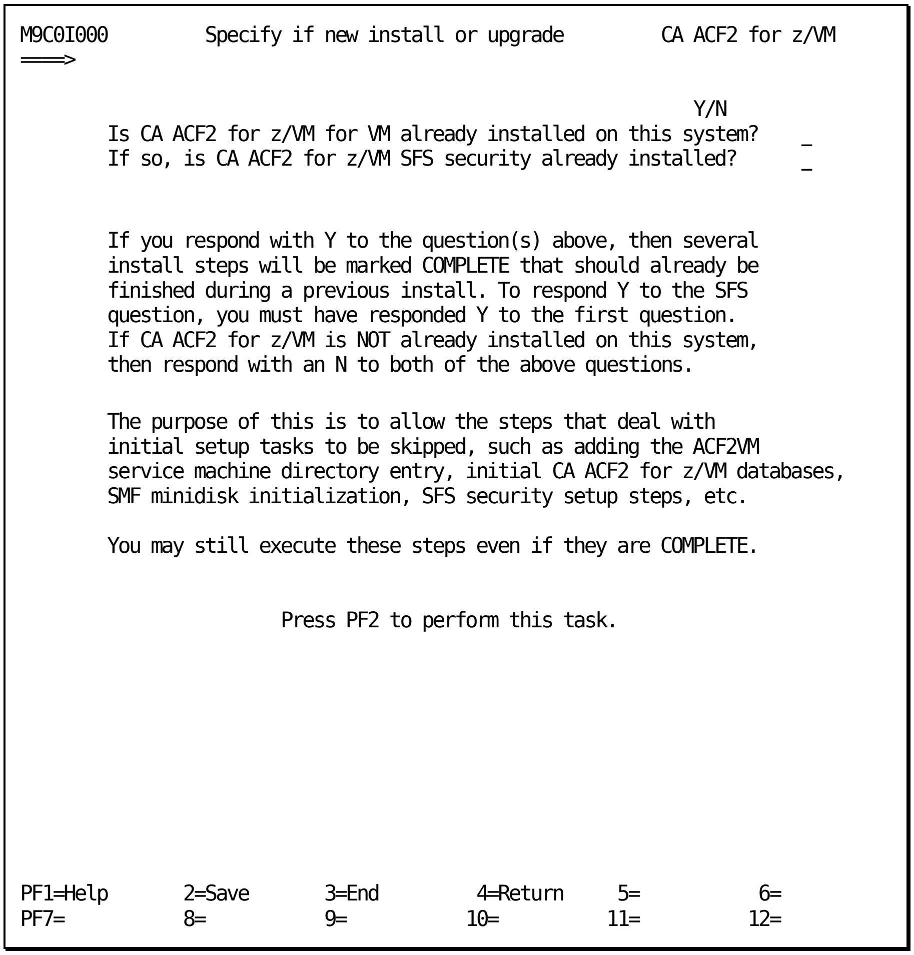
Task M9C0I000: Specify If New Install or Upgrade

This task indicates whether you are installing CA ACF2 for z/VM for the first time or this is an upgrade to an existing installation.

To complete this task:

1. Select panel M9C0I000 from the Task Selection Menu.

   The following panel appears:



```
M9C0I000        Specify if new install or upgrade        CA ACF2 for z/VM
===>


                                                     Y/N
        Is CA ACF2 for z/VM for VM already installed on this system?   _
        If so, is CA ACF2 for z/VM SFS security already installed?     _


        If you respond with Y to the question(s) above, then several
        install steps will be marked COMPLETE that should already be
        finished during a previous install. To respond Y to the SFS
        question, you must have responded Y to the first question.
        If CA ACF2 for z/VM is NOT already installed on this system,
        then respond with an N to both of the above questions.

        The purpose of this is to allow the steps that deal with
        initial setup tasks to be skipped, such as adding the ACF2VM
        service machine directory entry, initial CA ACF2 for z/VM databases,
        SMF minidisk initialization, SFS security setup steps, etc.

        You may still execute these steps even if they are COMPLETE.


                        Press PF2 to perform this task.








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

2. Specify N (No) if this is an initial installation of CA ACF2 for z/VM. Specifying Y (Yes) indicates this is an upgrade to an existing installation. The default is No.

3. Press PF2 to perform the task.

4. Press PF3 to return to the Task Selection Menu.

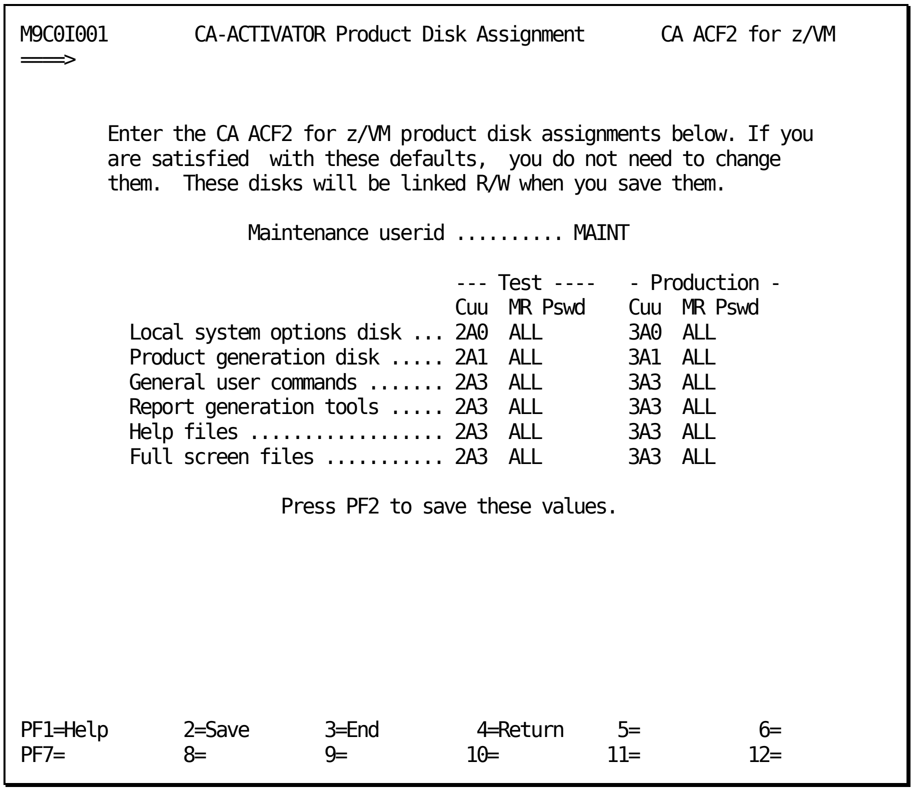
Task M9C0I001: CA-ACTIVATOR Product Disk Assignment

This task defines the minidisks CA ACF2 for z/VM uses on your system maintenance ID.

To complete this task:

1. Select panel M9C0I001 from the Task Selection Menu.

   The following panel appears:

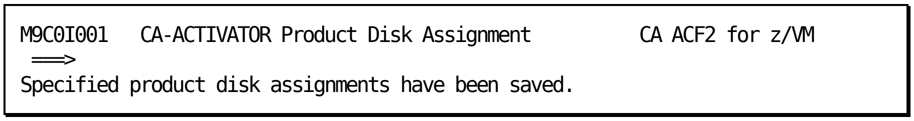


```
M9C0I001       CA-ACTIVATOR Product Disk Assignment       CA ACF2 for z/VM
===>


       Enter the CA ACF2 for z/VM product disk assignments below. If you
       are satisfied  with these defaults,  you do not need to change
       them.  These disks will be linked R/W when you save them.

                   Maintenance userid .......... MAINT

                                      --- Test ----   - Production -
                                      Cuu  MR Pswd    Cuu  MR Pswd
         Local system options disk ... 2A0  ALL        3A0  ALL
         Product generation disk ..... 2A1  ALL        3A1  ALL
         General user commands ....... 2A3  ALL        3A3  ALL
         Report generation tools ..... 2A3  ALL        3A3  ALL
         Help files .................. 2A3  ALL        3A3  ALL
         Full screen files ........... 2A3  ALL        3A3  ALL

                       Press PF2 to save these values.






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

   You can change the maintenance ID, the disk assignments, and the passwords for those disks. You can use the defaults if you do not want to make any changes.

2. Press PF2 to save these values.

   A message appears on the panel confirming that the disk assignments you specified are saved:



```
M9C0I001   CA-ACTIVATOR Product Disk Assignment          CA ACF2 for z/VM
 ===>
Specified product disk assignments have been saved.
```

3. Press PF3 to return to the Task Selection Menu.

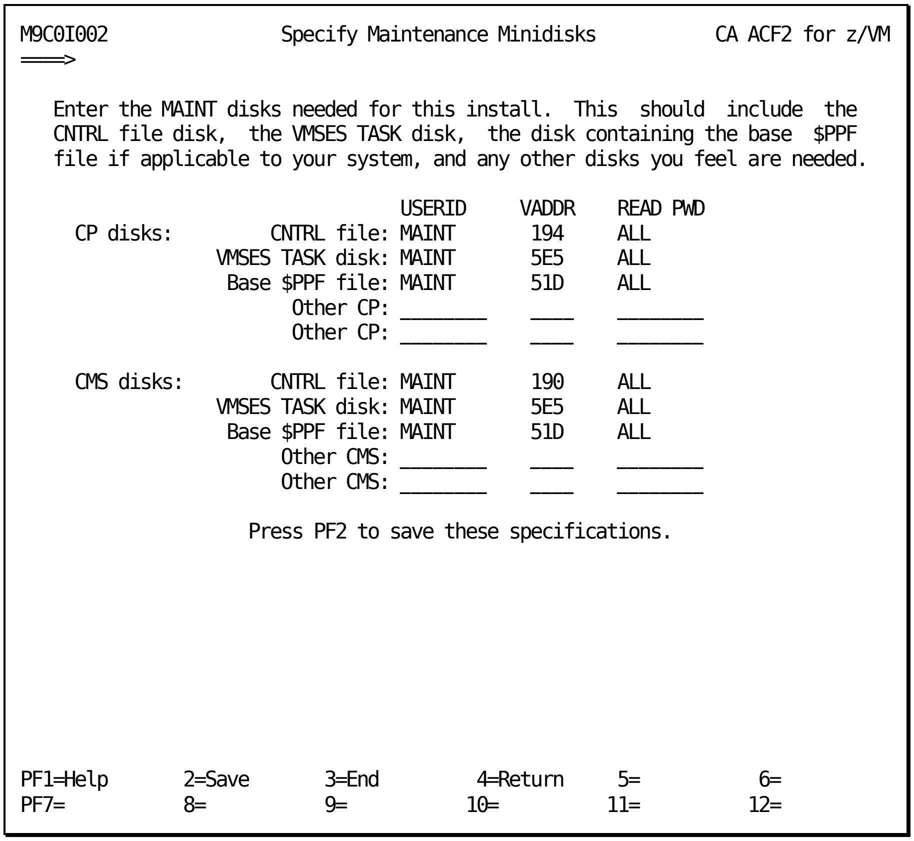
Task M9C0I002: Specify Maintenance Minidisks

This task lets you specify your VM maintenance ID and minidisks required for certain installation activities.

To complete this task:

1. Select panel M9C0I002 from the Task Selection Menu.

   The following panel appears, on which you can specify your maintenance ID and your CP and CMS disk locations.



```
M9C0I002                Specify Maintenance Minidisks           CA ACF2 for z/VM
===>

   Enter the MAINT disks needed for this install.  This  should  include  the
   CNTRL file disk,  the VMSES TASK disk,  the disk containing the base  $PPF
   file if applicable to your system, and any other disks you feel are needed.

                                   USERID     VADDR    READ PWD
     CP disks:          CNTRL file: MAINT       194     ALL
                    VMSES TASK disk: MAINT       5E5     ALL
                     Base $PPF file: MAINT       51D     ALL
                           Other CP: ________   ____    ________
                           Other CP: ________   ____    ________

     CMS disks:         CNTRL file: MAINT       190     ALL
                    VMSES TASK disk: MAINT       5E5     ALL
                     Base $PPF file: MAINT       51D     ALL
                          Other CMS: ________   ____    ________
                          Other CMS: ________   ____    ________

                      Press PF2 to save these specifications.







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

2. Enter the user ID, virtual address, and password of the CP and CMS disks that you want to access.

   Note: If you keep the IBM CNTRL files, $PPF files, and the SES task disk in SFS directories, you may leave these fields blank, but you must specifically ACCESS the required SFS directories in the CAIMAINT PROFILE EXEC.

3. Press PF2 to save the values.

   When processing is complete, a message appears on the panel confirming that the MAINT specifications you entered are saved.

4. Press PF3 to return to the Task Selection Menu.

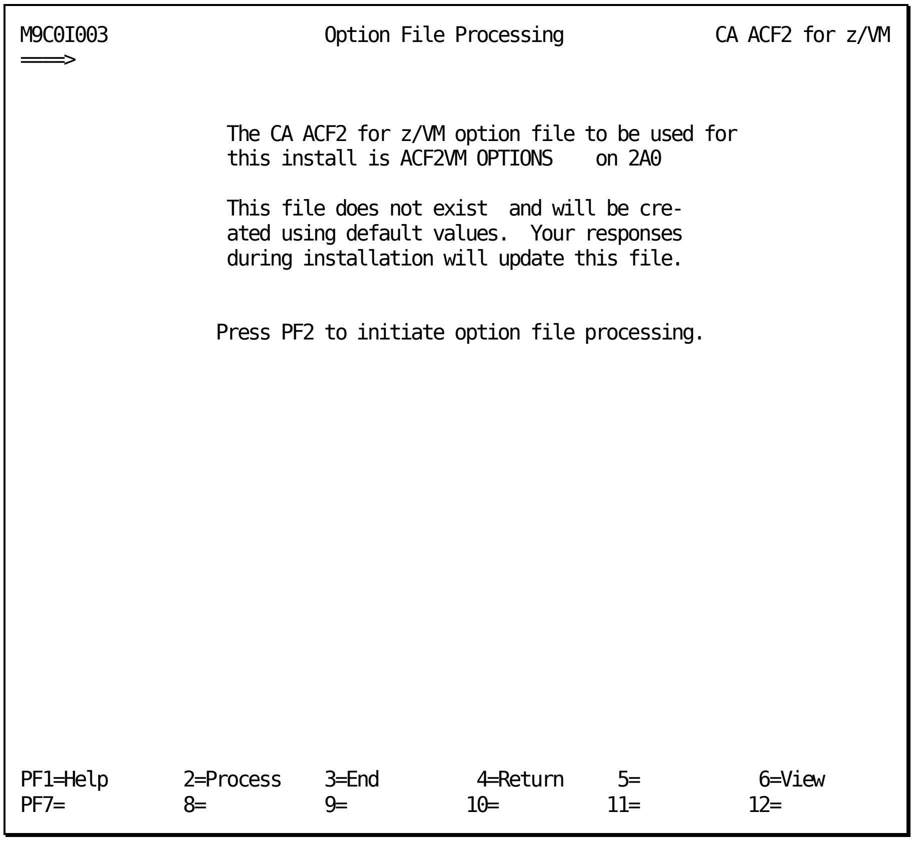
Task M9C0I003: Option File Processing

This task creates the ACF2VM OPTIONS file that contains information CA-ACTIVATOR needs to perform the remaining installation tasks.

To complete this task:

1. Select panel M9C0I003 from the Task Selection Menu.

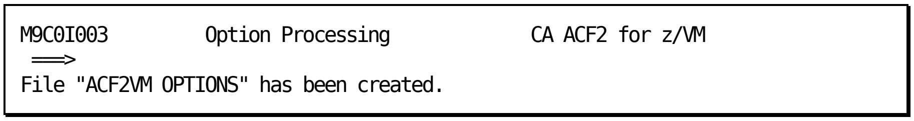
   The following panel appears:



```
M9C0I003                   Option File Processing              CA ACF2 for z/VM
===>


                   The CA ACF2 for z/VM option file to be used for
                   this install is ACF2VM OPTIONS    on 2A0

                   This file does not exist  and will be cre-
                   ated using default values.  Your responses
                   during installation will update this file.


                  Press PF2 to initiate option file processing.















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

2. Press PF2 to create ACF2VM OPTIONS on the specified disk.

   When processing is complete, a message appears on the panel confirming that the ACF2VM OPTIONS file is created:



```
M9C0I003        Option Processing             CA ACF2 for z/VM
 ===>
File "ACF2VM OPTIONS" has been created.
```

3. Press PF3 to return to the Task Selection Menu.

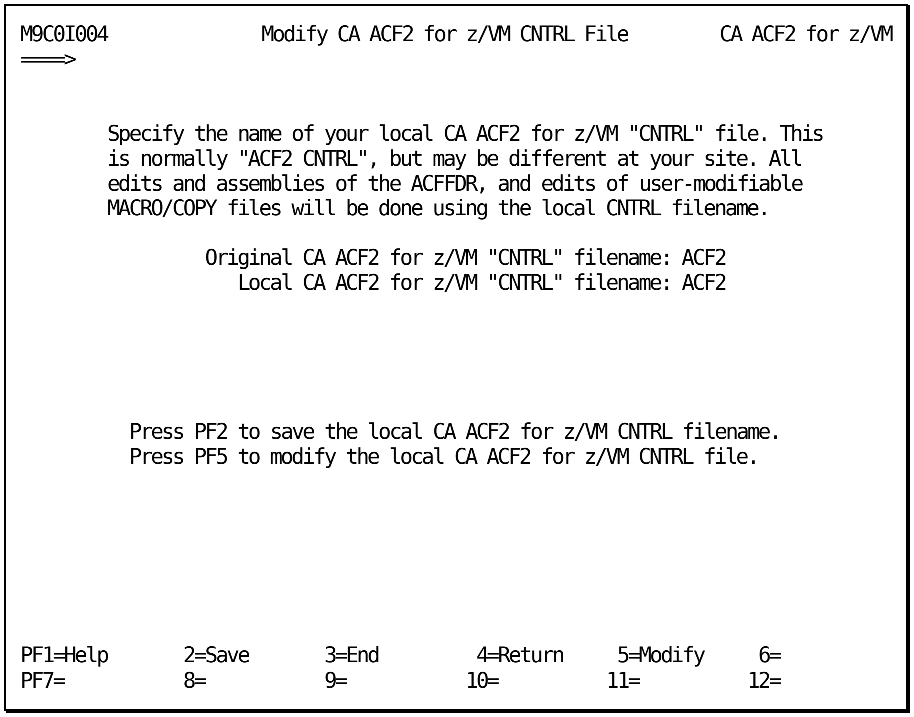
Task M9C0I004: Modify CA ACF2 for z/VM CNTRL File

This task lets you modify the CA ACF2 for z/VM CNTRL file that it uses to assemble the ACFFDR and update the user-tailorable macros. CA ACF2 for z/VM does not use this CNTRL file to assemble CP or CMS modules, nor to generate the CP or CMS nucleus.

To complete this task:

1. Select panel M9C0I004 from the Task Selection Menu.

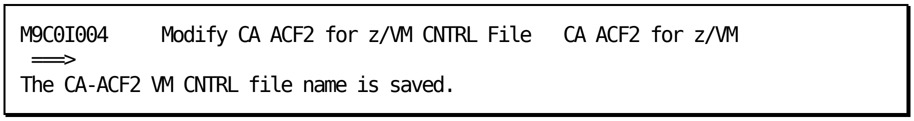
   The following panel appears:



```
M9C0I004            Modify CA ACF2 for z/VM CNTRL File        CA ACF2 for z/VM
===>


           Specify the name of your local CA ACF2 for z/VM "CNTRL" file. This
           is normally "ACF2 CNTRL", but may be different at your site. All
           edits and assemblies of the ACFFDR, and edits of user-modifiable
           MACRO/COPY files will be done using the local CNTRL filename.

                       Original CA ACF2 for z/VM "CNTRL" filename: ACF2
                          Local CA ACF2 for z/VM "CNTRL" filename: ACF2





             Press PF2 to save the local CA ACF2 for z/VM CNTRL filename.
             Press PF5 to modify the local CA ACF2 for z/VM CNTRL file.








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

2. Press PF2 to save the CA ACF2 for z/VM CNTRL filename or PF5 to update it.

   When processing is complete, a message appears on the panel confirming that the CNTRL file was saved:



```
M9C0I004     Modify CA ACF2 for z/VM CNTRL File    CA ACF2 for z/VM
 ===>
The CA-ACF2 VM CNTRL file name is saved.
```

3. Press PF3 to return to the Task Selection Menu.

Task M9C0I005: Specify CP Installation Option

This task lets you specify the current CP release at your site.

Follow these steps:

1. Select panel M9C0I005 from the Task Selection Menu.

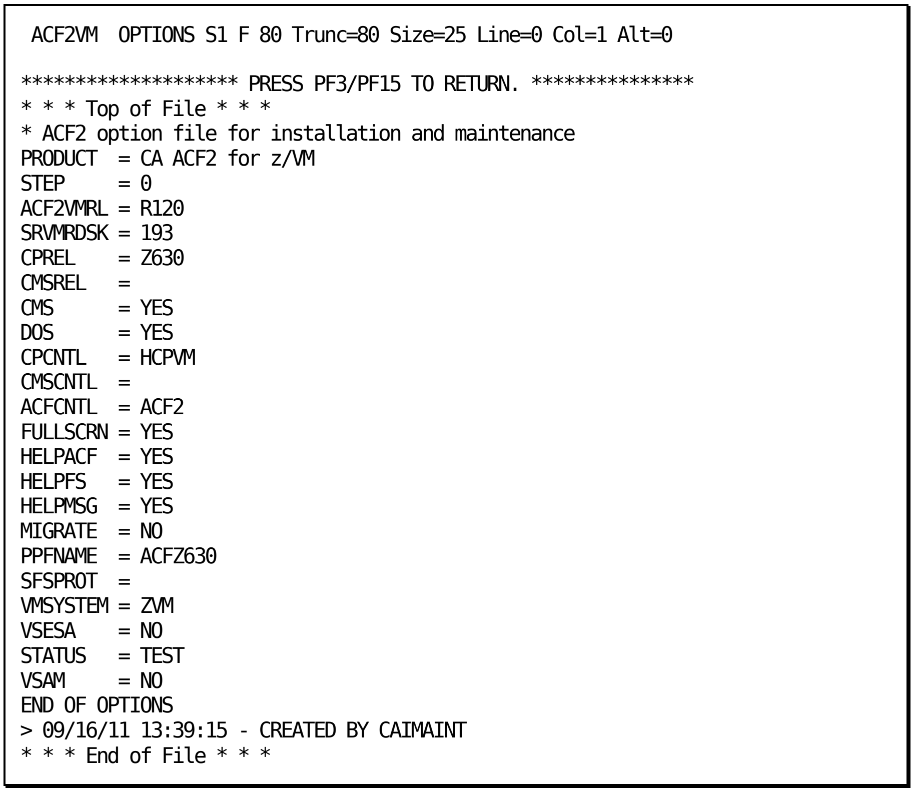
   The Specify CP Installation Option panel appears.

2. Specify your current CP release level and press PF2 to save your option.

   When processing is complete, a message appears on the panel confirming that the installation options you specified are saved.

3. Press PF6 to display the ACF2VM OPTIONS file. You cannot modify any of these options. If you need to change this file, make them outside CA-ACTIVATOR and save them.

   The following screen is a sample for Version 6 Release 3.0.



```
 ACF2VM  OPTIONS S1 F 80 Trunc=80 Size=25 Line=0 Col=1 Alt=0

******************** PRESS PF3/PF15 TO RETURN. ***************
* * * Top of File * * *
* ACF2 option file for installation and maintenance
PRODUCT  = CA ACF2 for z/VM
STEP     = 0
ACF2VMRL = R120
SRVMRDSK = 193
CPREL    = Z630
CMSREL   =
CMS      = YES
DOS      = YES
CPCNTL   = HCPVM
CMSCNTL  =
ACFCNTL  = ACF2
FULLSCRN = YES
HELPACF  = YES
HELPFS   = YES
HELPMSG  = YES
MIGRATE  = NO
PPFNAME  = ACFZ630
SFSPROT  =
VMSYSTEM = ZVM
VSESA    = NO
STATUS   = TEST
VSAM     = NO
END OF OPTIONS
> 09/16/11 13:39:15 - CREATED BY CAIMAINT
* * * End of File * * *
```

4. Press PF3 to return to the Task Selection Menu.

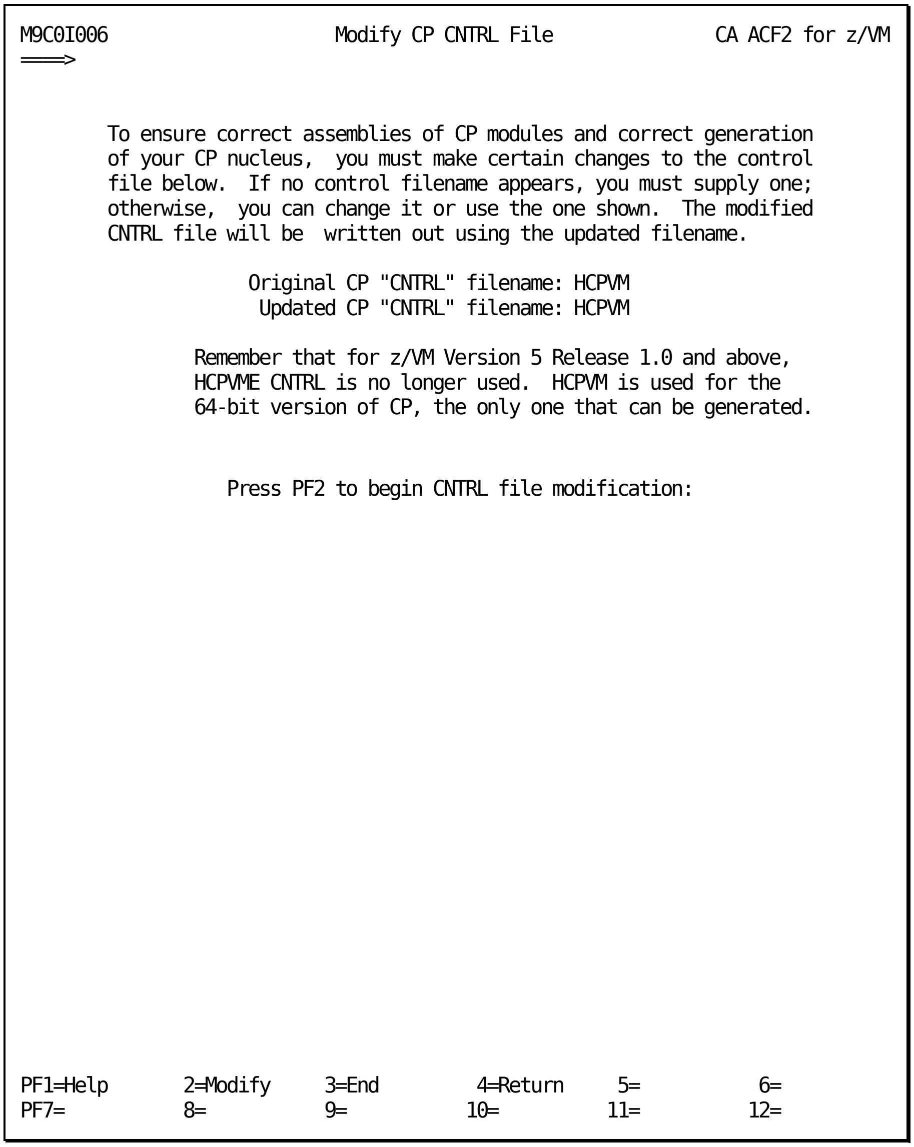
Task M9C0I006: Modify CP CNTRL File

This task lets you create the CP CNTRL file that CA ACF2 for z/VM uses to assemble the HCPAC0 module. This CNTRL file also generates the CP nucleus.

Follow these steps:

1. Select task M9C0I006 from the Task Selection Menu.

   The following panel appears:

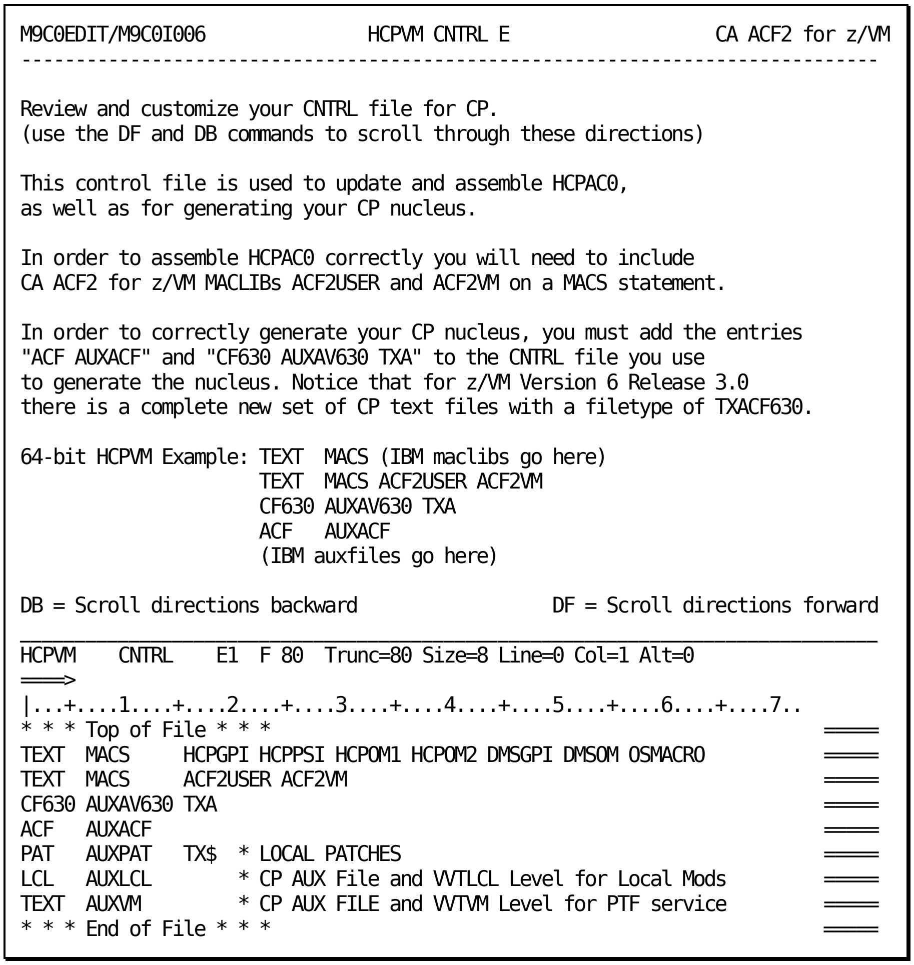


```
M9C0I006                    Modify CP CNTRL File                CA ACF2 for z/VM
===>


        To ensure correct assemblies of CP modules and correct generation
        of your CP nucleus,  you must make certain changes to the control
        file below.  If no control filename appears, you must supply one;
        otherwise,  you can change it or use the one shown.  The modified
        CNTRL file will be  written out using the updated filename.

                     Original CP "CNTRL" filename: HCPVM
                      Updated CP "CNTRL" filename: HCPVM

                Remember that for z/VM Version 5 Release 1.0 and above,
                HCPVME CNTRL is no longer used.  HCPVM is used for the
                64-bit version of CP, the only one that can be generated.


                   Press PF2 to begin CNTRL file modification:


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

- If you want to change the updated CNTRL filename, enter it on the panel. Be sure to use the same CNTRL filename in task M9C0I017.

2. Answer the questions with the appropriate value and press PF2 to apply the changes.

   The CNTRL file you specified appears:



```
M9C0EDIT/M9C0I006              HCPVM CNTRL E                 CA ACF2 for z/VM
------------------------------------------------------------------------------

Review and customize your CNTRL file for CP.
(use the DF and DB commands to scroll through these directions)

This control file is used to update and assemble HCPAC0,
as well as for generating your CP nucleus.

In order to assemble HCPAC0 correctly you will need to include
CA ACF2 for z/VM MACLIBs ACF2USER and ACF2VM on a MACS statement.

In order to correctly generate your CP nucleus, you must add the entries
"ACF AUXACF" and "CF630 AUXAV630 TXA" to the CNTRL file you use
to generate the nucleus. Notice that for z/VM Version 6 Release 3.0
there is a complete new set of CP text files with a filetype of TXACF630.

64-bit HCPVM Example: TEXT  MACS (IBM maclibs go here)
                      TEXT  MACS ACF2USER ACF2VM
                      CF630 AUXAV630 TXA
                      ACF   AUXACF
                      (IBM auxfiles go here)

DB = Scroll directions backward                  DF = Scroll directions forward
______________________________________________________________________________
HCPVM    CNTRL    E1  F 80  Trunc=80 Size=8 Line=0 Col=1 Alt=0
===>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
* * * Top of File * * *                                                  =====
TEXT  MACS     HCPGPI HCPPSI HCPOM1 HCPOM2 DMSGPI DMSOM OSMACRO          =====
TEXT  MACS     ACF2USER ACF2VM                                           =====
CF630 AUXAV630 TXA                                                       =====
ACF   AUXACF                                                             =====
PAT   AUXPAT   TX$  * LOCAL PATCHES                                      =====
LCL   AUXLCL        * CP AUX File and VVTLCL Level for Local Mods        =====
TEXT  AUXVM         * CP AUX FILE and VVTVM Level for PTF service        =====
* * * End of File * * *                                                  =====
```

Note: This screen is a sample for Version 6 Release 3.0.

- z/VM Version 5 Release 1.0 is 64-bit only. Because of the difference from the older z/VM releases, there is a full new set of CA ACF2 for z/VM CP text files with file type TXACF510, and a full set of AUXAV510 files. Therefore, use the CF510 AUXAV510 TXA statement in your HCPVM CNTRL file. You do not need the CF510 statement. Although z/VM Version 5 Release 1.0 is 64-bit, HCPVM CNTRL is used (not HCPVME) and TXA is used (not TXE) to select the CA ACF2 for z/VM TXACF510 text files.

- z/VM Version 5 Release 2.0 is 64-bit only. Because of the difference from the older z/VM releases, there is a full new set of CA ACF2 for z/VM CP text files with file type TXACF520, and a full set of AUXAV520 files. Therefore, use the CF520 AUXAV520 TXA statement in your HCPVM CNTRL file. You do not need the CF510 statement. Although z/VM Version 5 Release 2.0 is 64-bit, HCPVM CNTRL is used (not HCPVME) and TXA is used (not TXE) to select the CA ACF2 for z/VM TXACF520 text files.

- z/VM Version 5 Release 3.0 is 64-bit only. Because of the difference from the older z/VM releases, there is a full new set of CA ACF2 for z/VM CP text files with file type TXACF530, and a full set of AUXAV530 files. Therefore, use the CF530 AUXAV530 TXA statement in your HCPVM CNTRL file. You do not need the CF510 statement. Although z/VM Version 5 Release 3.0 is 64-bit, HCPVM CNTRL is used (not HCPVME) and TXA is used (not TXE) to select the CA ACF2 for z/VM TXACF530 text files.

- z/VM Version 5 Release 4.0 is 64-bit only. Because of the difference from the older z/VM releases, there is a full new set of CA ACF2 for z/VM CP text files with file type TXACF540 and a full set of AUXAV540 files. Therefore, use the CF610 AUXAV610 TXA statement in your HCPVM CNTRL file. You do not need the CF510 statement. Although z/VM Version 5 Release 4.0 is 64-bit, HCPVM CNTRL is used (not HCPVME) and TXA is used (not TXE) to select the CA ACF2 for z/VM TXACF540 text files.

- z/VM Version 6 Release 1.0 is 64-bit only. Because of the difference from the older z/VM releases, there is a full new set of CA ACF2 for z/VM CP text files with file type TXACF610, and a full set of AUXAV610 files. Therefore, use the CF610 AUXAV610 TXA statement in your HCPVM CNTRL file. You do not need the CF510 statement. Although z/VM Version 6 Release 1.0 is 64-bit, HCPVM CNTRL is used (not HCPVME) and TXA is used (not TXE) to select the CA ACF2 for z/VM TXACF610 text files.

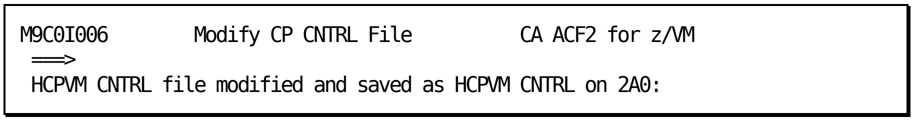
- z/VM Version 6 Release 2.0 is 64-bit only. Because of the difference from the older z/VM releases, there is a full new set of CA ACF2 for z/VM CP text files with file type TXACF620, and a full set of AUXAV620 files. Therefore, use the CF620 AUXAV620 TXA statement in your HCPVM CNTRL file. You do not need the CF510 statement. Although z/VM Version 6 Release 2.0 is 64-bit, HCPVM CNTRL is used (not HCPVME) and TXA is used (not TXE) to select the CA ACF2 for z/VM TXACF620 text files.

- z/VM Version 6 Release 3.0 is 64-bit only. Because of the difference from the older z/VM releases, there is a full new set of CA ACF2 for z/VM CP text files with file type TXACF630, and a full set of AUXAV630 files. Therefore, use the CF630 AUXAV630 TXA statement in your HCPVM CNTRL file. You do not need the CF510 statement. Although z/VM Version 6 Release 3.0 is 64-bit, HCPVM CNTRL is used (not HCPVME) and TXA is used (not TXE) to select the CA ACF2 for z/VM TXACF630 text files.

- Make the necessary changes to this file as the directions at the top of the panel indicate.

- Type FILE and press Enter to save your changes.

When processing is complete, a message appears on the panel confirming that you modified and saved the CNTRL file:



```
M9C0I006       Modify CP CNTRL File         CA ACF2 for z/VM
 ===>
 HCPVM CNTRL file modified and saved as HCPVM CNTRL on 2A0:
```

3. Press PF3 to return to the Task Selection Menu.

Task M9C0I007: Specify CMS Installation Option

You can specify the CMS release you are running at your site, and whether you want to install CMS file protection and CMS/DOS intercepts.

Follow these steps:

1. Select panel M9C0I007 from the Task Selection Menu.

   The Specify CMS Installation Options panel appears.

2. Specify **Y** if you want to install CMS file protection and CMS/DOS intercepts. Enter **N** if you do not want to install these features.

3. Press PF6 to display the ACF2VM OPTIONS file. You cannot modify any of these options. If you must change this file, return to the proper installation steps.

   When processing is complete, a message appears on the panel confirming that the installation options you specified are saved.

4. Press PF3 to return to the Task Selection Menu.

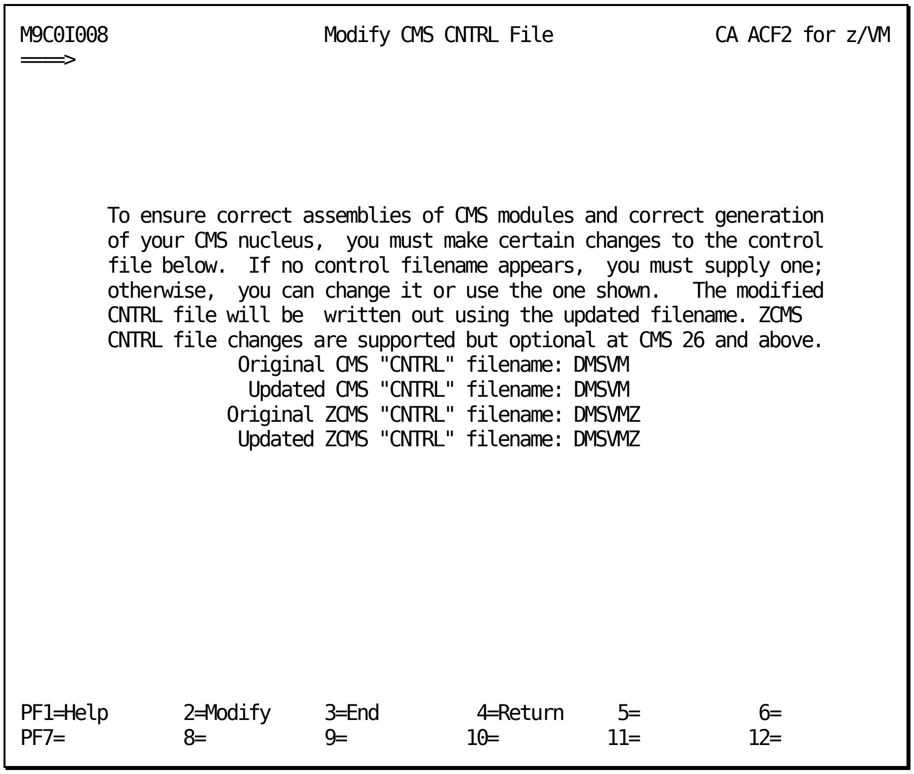
Task M9C0I008: Modify CMS CNTRL File

This task lets you create the CMS CNTRL file CA ACF2 for z/VM uses to assemble the CA ACF2 for z/VM source-intercepted CMS modules and to generate your CMS nucleus.

To complete this task:

1. Select task M9C0I008 from the Task Selection Menu.

   The following panel appears:

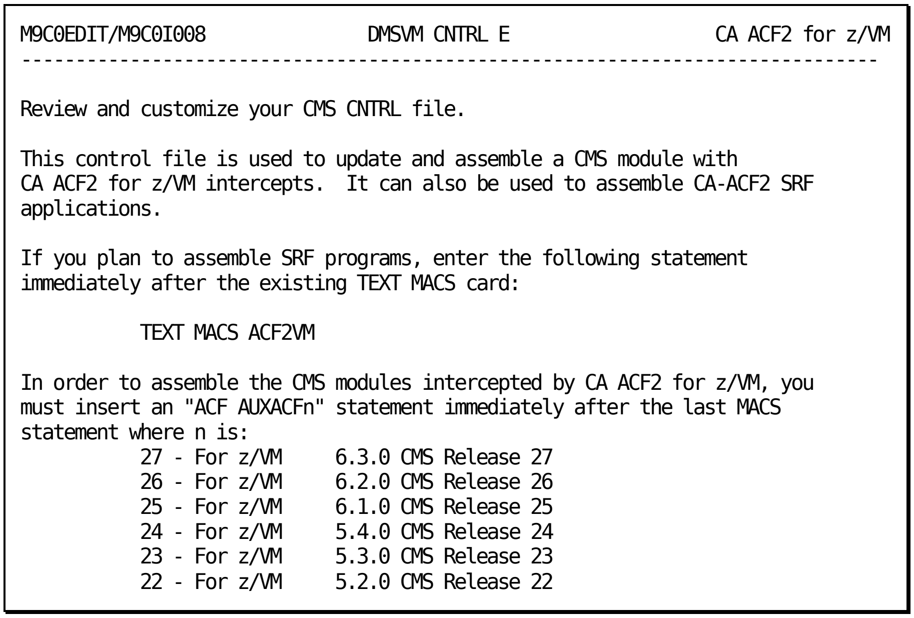


```
M9C0I008                    Modify CMS CNTRL File              CA ACF2 for z/VM
====>




        To ensure correct assemblies of CMS modules and correct generation
        of your CMS nucleus,  you must make certain changes to the control
        file below.  If no control filename appears,  you must supply one;
        otherwise,  you can change it or use the one shown.   The modified
        CNTRL file will be  written out using the updated filename. ZCMS
        CNTRL file changes are supported but optional at CMS 26 and above.
                    Original CMS "CNTRL" filename: DMSVM
                     Updated CMS "CNTRL" filename: DMSVM
                   Original ZCMS "CNTRL" filename: DMSVMZ
                    Updated ZCMS "CNTRL" filename: DMSVMZ








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

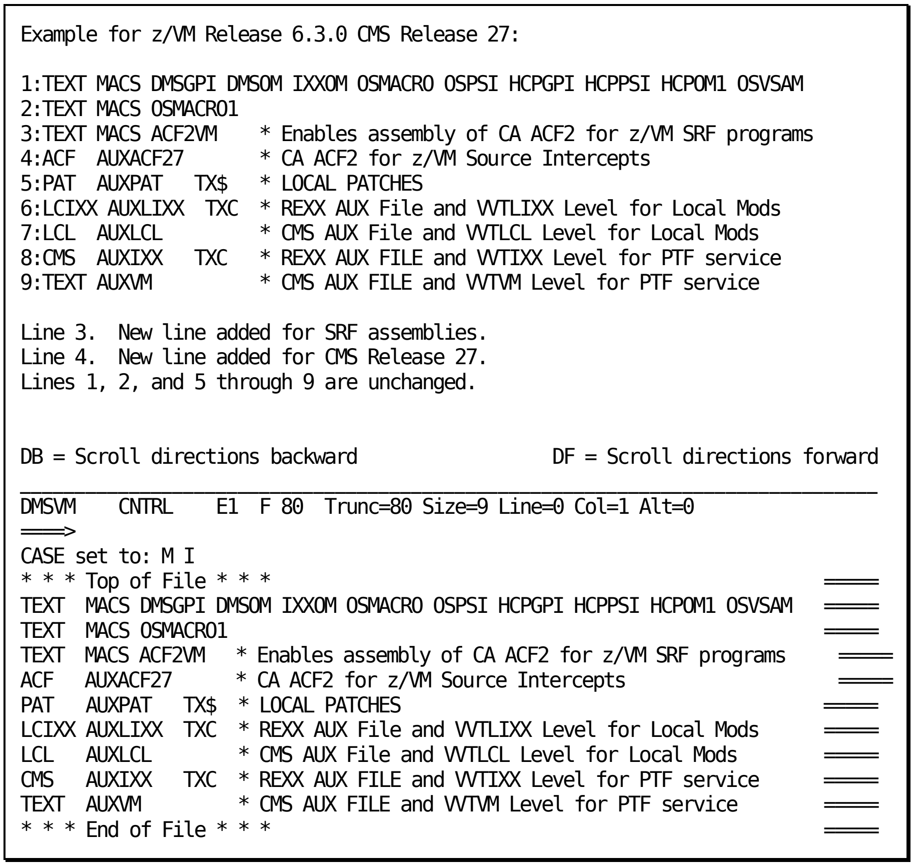
   If you want to change the updated CNTRL filename, enter it on the panel. Use the same CNTRL filename in task M9C0I017.

2. Answer the questions with the appropriate values and press PF2 to make the changes.

   The CNTRL file you specified appears:



```
M9C0EDIT/M9C0I008              DMSVM CNTRL E                 CA ACF2 for z/VM
-------------------------------------------------------------------------------

Review and customize your CMS CNTRL file.

This control file is used to update and assemble a CMS module with
CA ACF2 for z/VM intercepts.  It can also be used to assemble CA-ACF2 SRF
applications.

If you plan to assemble SRF programs, enter the following statement
immediately after the existing TEXT MACS card:

           TEXT MACS ACF2VM

In order to assemble the CMS modules intercepted by CA ACF2 for z/VM, you
must insert an "ACF AUXACFn" statement immediately after the last MACS
statement where n is:
           27 - For z/VM     6.3.0 CMS Release 27
           26 - For z/VM     6.2.0 CMS Release 26
           25 - For z/VM     6.1.0 CMS Release 25
           24 - For z/VM     5.4.0 CMS Release 24
           23 - For z/VM     5.3.0 CMS Release 23
           22 - For z/VM     5.2.0 CMS Release 22
```

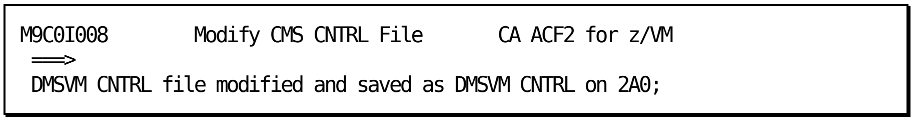


```
Example for z/VM Release 6.3.0 CMS Release 27:

1:TEXT MACS DMSGPI DMSOM IXXOM OSMACRO OSPSI HCPGPI HCPPSI HCPOM1 OSVSAM
2:TEXT MACS OSMACRO1
3:TEXT MACS ACF2VM    * Enables assembly of CA ACF2 for z/VM SRF programs
4:ACF  AUXACF27       * CA ACF2 for z/VM Source Intercepts
5:PAT  AUXPAT   TX$   * LOCAL PATCHES
6:LCIXX AUXLIXX  TXC  * REXX AUX File and VVTLIXX Level for Local Mods
7:LCL  AUXLCL         * CMS AUX File and VVTLCL Level for Local Mods
8:CMS  AUXIXX   TXC   * REXX AUX FILE and VVTIXX Level for PTF service
9:TEXT AUXVM          * CMS AUX FILE and VVTVM Level for PTF service

Line 3.  New line added for SRF assemblies.
Line 4.  New line added for CMS Release 27.
Lines 1, 2, and 5 through 9 are unchanged.


DB = Scroll directions backward                DF = Scroll directions forward
________________________________________________________________________________
DMSVM    CNTRL    E1  F 80  Trunc=80 Size=9 Line=0 Col=1 Alt=0
====>
CASE set to: M I
* * * Top of File * * *                                                    =====
TEXT  MACS DMSGPI DMSOM IXXOM OSMACRO OSPSI HCPGPI HCPPSI HCPOM1 OSVSAM     =====
TEXT  MACS OSMACRO1                                                        =====
TEXT  MACS ACF2VM   * Enables assembly of CA ACF2 for z/VM SRF programs     =====
ACF   AUXACF27      * CA ACF2 for z/VM Source Intercepts                    =====
PAT   AUXPAT   TX$  * LOCAL PATCHES                                        =====
LCIXX AUXLIXX  TXC  * REXX AUX File and VVTLIXX Level for Local Mods       =====
LCL   AUXLCL        * CMS AUX File and VVTLCL Level for Local Mods         =====
CMS   AUXIXX   TXC  * REXX AUX FILE and VVTIXX Level for PTF service       =====
TEXT  AUXVM         * CMS AUX FILE and VVTVM Level for PTF service         =====
* * * End of File * * *                                                    =====
```

3. Make the necessary changes to this file as the directions at the top of the panel indicate.

4. Type FILE and press Enter to save your changes.

   When processing is complete, a message appears on the panel confirming that you modified and saved the CNTRL file:



```
M9C0I008       Modify CMS CNTRL File       CA ACF2 for z/VM
 ===>
 DMSVM CNTRL file modified and saved as DMSVM CNTRL on 2A0;
```

5. Press PF3 to return to the Task Selection Menu.

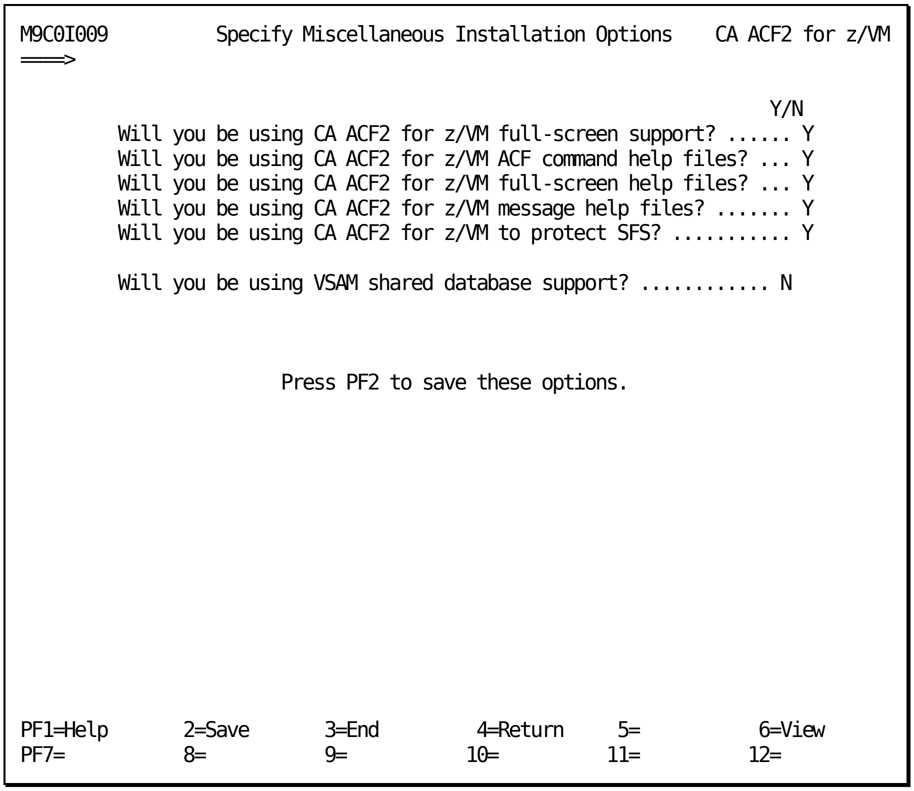
Task M9C0I009: Specify Miscellaneous Installation Options

This task lets you specify various installation options.

To complete this task:

1. Select task M9C0I009 from the Task Selection Menu.

   The following panel appears:



```
M9C0I009         Specify Miscellaneous Installation Options    CA ACF2 for z/VM
===>

                                                           Y/N
         Will you be using CA ACF2 for z/VM full-screen support? ...... Y
         Will you be using CA ACF2 for z/VM ACF command help files? ... Y
         Will you be using CA ACF2 for z/VM full-screen help files? ... Y
         Will you be using CA ACF2 for z/VM message help files? ....... Y
         Will you be using CA ACF2 for z/VM to protect SFS? ........... Y

         Will you be using VSAM shared database support? ........... N



                        Press PF2 to save these options.














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

2. Answer the questions with Y if you are installing the specified support or N if you do not want to install these featuresWill you be using CA-ACF2 full-screen support?

   If you specify Y (the default), CA-ACTIVATOR copies full-screen support files from the CA ACF2 for z/VM minidisks to a user-specified minidisk. Allow any user who can use the CA ACF2 for z/VM full-screen functions access to this disk. You must convert the full-screen files into REXX format before using them or they will not function. If you ever intend to use the full-screen interface, do not bypass this task.

   To copy these files, you must have read/write access to the minidisk that contains the ACFFSCMD ACFCOPY and ACFPANEL ACFCOPY files. CA-ACTIVATOR uses these files to determine which files to copy. You can review and customize these files before executing this task. You do not normally need to customize theWill you be using CA-ACF2 help files for the ACF command?

   If you specify Y (the default), CA-ACTIVATOR copies command files from the CA ACF2 for z/VM minidisks to a user-specified minidisk. Allow any user who can issue ACF commands, such as ACF, ACFCOMP, and ACFDCMP, access to this disk. Common choices for target disks are 319 P-disk and the 193 Y-disk.

   CA-ACTIVATOR also copies the CA ACF2 for z/VM command files to the command minidisk and replaces any files of the same name. The file named ACFCMNDS ACFCOPY determines which files are copied. You can review this file and customize it before executing this task. You do not normally need to customize them.

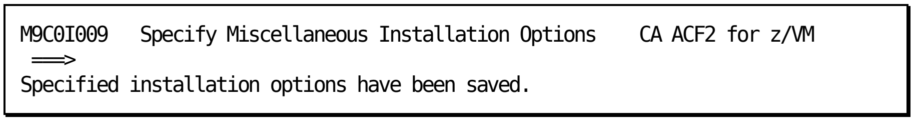
   To copy these files, you must have read/write access to the minidisk that contains the ACFCMNDS ACFCOPY file.Will you be using CA-ACF2 full-screen help files?

   If you specify Y (the default), CA-ACTIVATOR copies the files specified in ACFFSHLP ACFCOPY file. Allow all users who may need to use the full-screen help files access to this minidisk.Will you be using CA-ACF2 message help files?

   If you specify Y (the default), CA-ACTIVATOR copies the files specified in the MSGHELP ACFCOPY file. Allow all users who may need to use the message help files access to this minidisk.

3. Press PF2 to execute the task.

   When processing is complete, a message appears on the panel confirming that the installation options you specified are saved:



```
M9C0I009   Specify Miscellaneous Installation Options    CA ACF2 for z/VM
 ===>
Specified installation options have been saved.
```

4. Press PF3 to return to the Task Selection Menu.
5. Will you be using CA ACF2 for z/VM to protect SFS?

   Specify Y if you will be protecting SFS directories and files with CA ACF2 for z/VM. If you specify N, CA-ACTIVATOR will allow you to bypass the SFS security setup steps.

6. Will you be using VSAM shared database support?

   Specify Y if you will be sharing CA ACF2 for z/VM VSAM databases with another VM system running CA ACF2 for z/VM Security for VM.

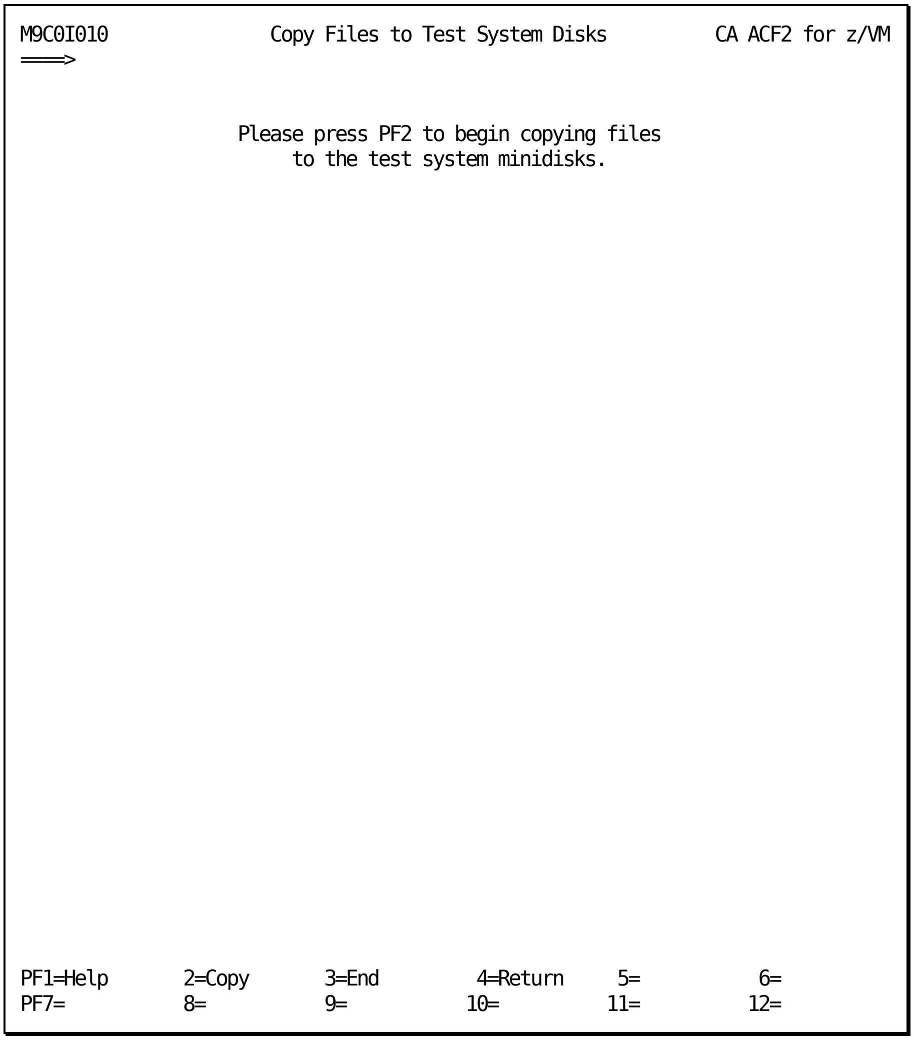
Task M9C0I010: Copy Files to Test System Disks

This task copies the files you selected in task M9C0I009 to the test system minidisks.

To complete this task:

1. Select panel M9C0I010 from the Task Selection Menu.

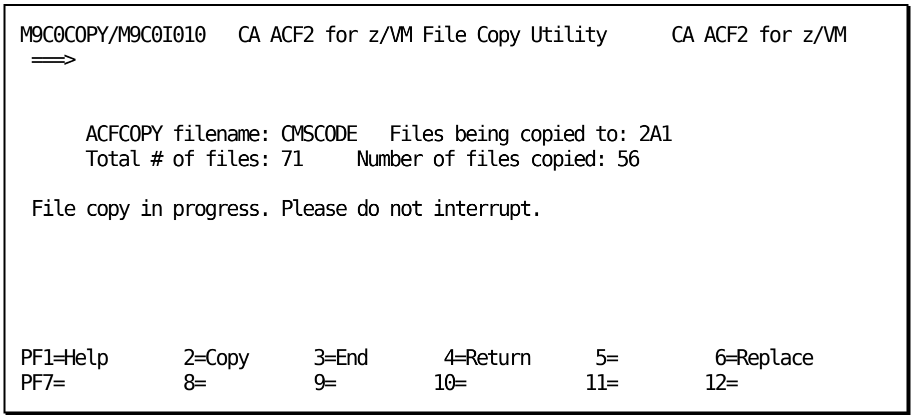
   The following panel appears:



```
M9C0I010              Copy Files to Test System Disks          CA ACF2 for z/VM
===>


                   Please press PF2 to begin copying files
                        to the test system minidisks.






















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

2. Press PF2 to copy the files.

   The following panel displays the information for each set of files that you are copying. For example, if you specified Y for the CA ACF2 for z/VM full-screen file support and CA ACF2 for z/VM help files for the ACF command, this panel displays the information for the full-screen file support, then the information for the help files.



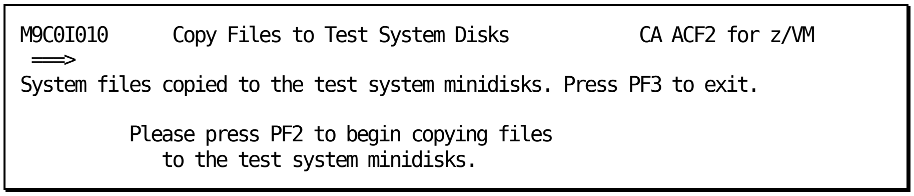
```
M9C0COPY/M9C0I010   CA ACF2 for z/VM File Copy Utility      CA ACF2 for z/VM
 ===>


      ACFCOPY filename: CMSCODE   Files being copied to: 2A1
      Total # of files: 71     Number of files copied: 56

 File copy in progress. Please do not interrupt.





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

   This panel tells you the following information:

   - What control file is used to control the copy
   - What disk the files are being copied to
   - How many files will be copied
   - How many files were copied so far.

   The value for Number of files copied: changes as you watch the panel.

   When all the files are copied (all the support you selected in task M9C0I009), the following panel appears:



```
M9C0I010      Copy Files to Test System Disks            CA ACF2 for z/VM
 ===>
System files copied to the test system minidisks. Press PF3 to exit.

          Please press PF2 to begin copying files
             to the test system minidisks.
```

3. Press PF3 to return to the Task Selection Menu.

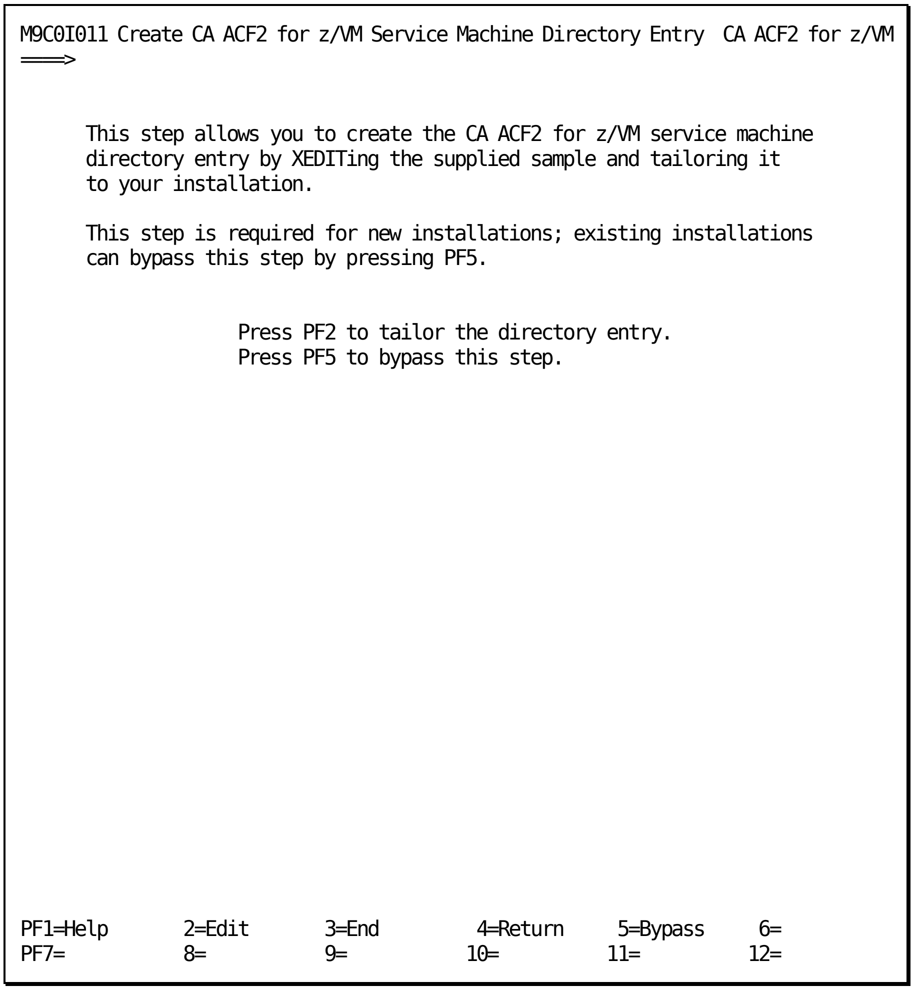
Task M9C0I011: Create CA ACF2 for z/VM Service Machine Directory Entry

This task defines the directory entry for the CA ACF2 for z/VM service machine. It is required for new users. If you have already defined your service machine, you can press PF5 to bypass this task. It will be marked COMPLETE on the Test System Generation - Task Selection panel (CACT-2121).

To complete this task:

1. Select the task M9C0I011 from the Task Selection Menu.

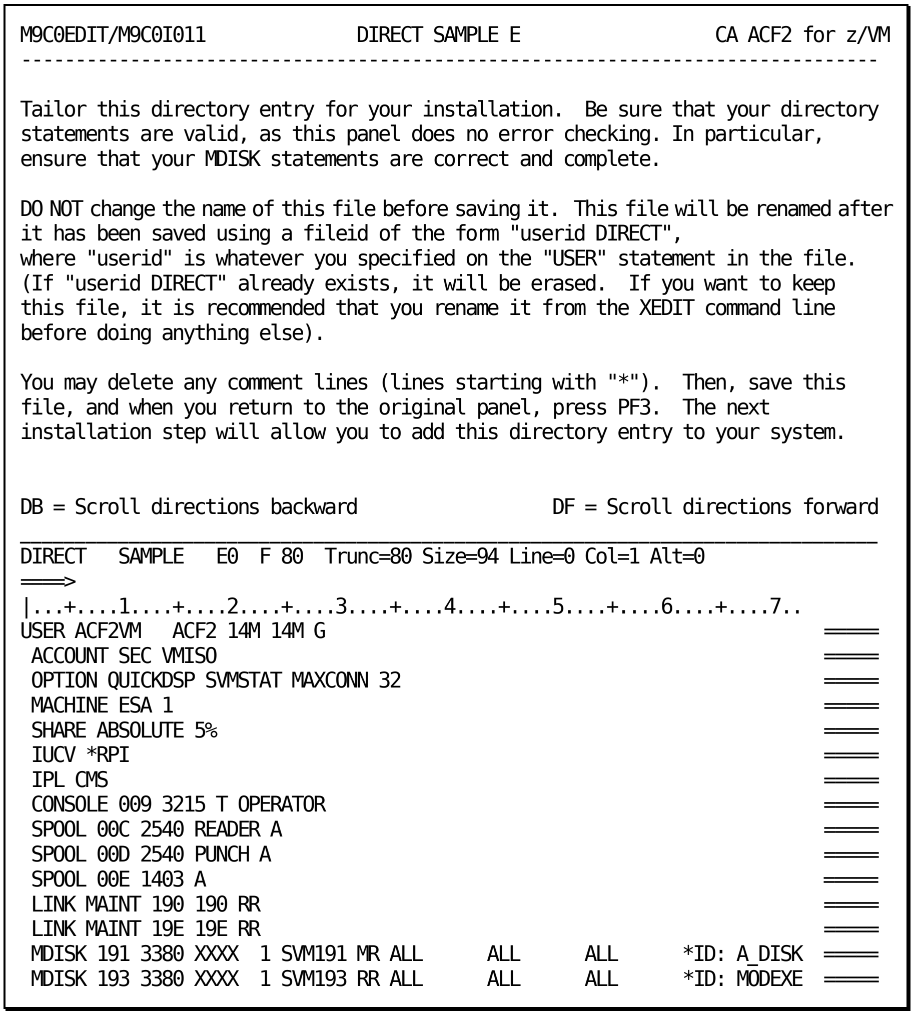
   The following panel appears:



```
M9C0I011 Create CA ACF2 for z/VM Service Machine Directory Entry  CA ACF2 for z/VM
===>


      This step allows you to create the CA ACF2 for z/VM service machine
      directory entry by XEDITing the supplied sample and tailoring it
      to your installation.

      This step is required for new installations; existing installations
      can bypass this step by pressing PF5.


                    Press PF2 to tailor the directory entry.
                    Press PF5 to bypass this step.


















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

2. Press PF2 to tailor the service machine directory entry.

   The following appears:



```
M9C0EDIT/M9C0I011             DIRECT SAMPLE E                 CA ACF2 for z/VM
-------------------------------------------------------------------------------

Tailor this directory entry for your installation.  Be sure that your directory
statements are valid, as this panel does no error checking. In particular,
ensure that your MDISK statements are correct and complete.

DO NOT change the name of this file before saving it.  This file will be renamed after
it has been saved using a fileid of the form "userid DIRECT",
where "userid" is whatever you specified on the "USER" statement in the file.
(If "userid DIRECT" already exists, it will be erased.  If you want to keep
this file, it is recommended that you rename it from the XEDIT command line
before doing anything else).

You may delete any comment lines (lines starting with "*").  Then, save this
file, and when you return to the original panel, press PF3.  The next
installation step will allow you to add this directory entry to your system.


DB = Scroll directions backward                  DF = Scroll directions forward
_______________________________________________________________________________
DIRECT   SAMPLE   E0  F 80  Trunc=80 Size=94 Line=0 Col=1 Alt=0
====>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
USER ACF2VM   ACF2 14M 14M G                                               =====
 ACCOUNT SEC VMISO                                                         =====
 OPTION QUICKDSP SVMSTAT MAXCONN 32                                        =====
 MACHINE ESA 1                                                             =====
 SHARE ABSOLUTE 5%                                                         =====
 IUCV *RPI                                                                 =====
 IPL CMS                                                                   =====
 CONSOLE 009 3215 T OPERATOR                                               =====
 SPOOL 00C 2540 READER A                                                   =====
 SPOOL 00D 2540 PUNCH A                                                    =====
 SPOOL 00E 1403 A                                                          =====
 LINK MAINT 190 190 RR                                                     =====
 LINK MAINT 19E 19E RR                                                     =====
 MDISK 191 3380 XXXX  1 SVM191 MR ALL      ALL      ALL      *ID: A_DISK   =====
 MDISK 193 3380 XXXX  1 SVM193 RR ALL      ALL      ALL      *ID: MODEXE   =====
```

3. Type FILE and press Enter to save your changes.

4. Press PF3 to return to the Task Selection Menu.

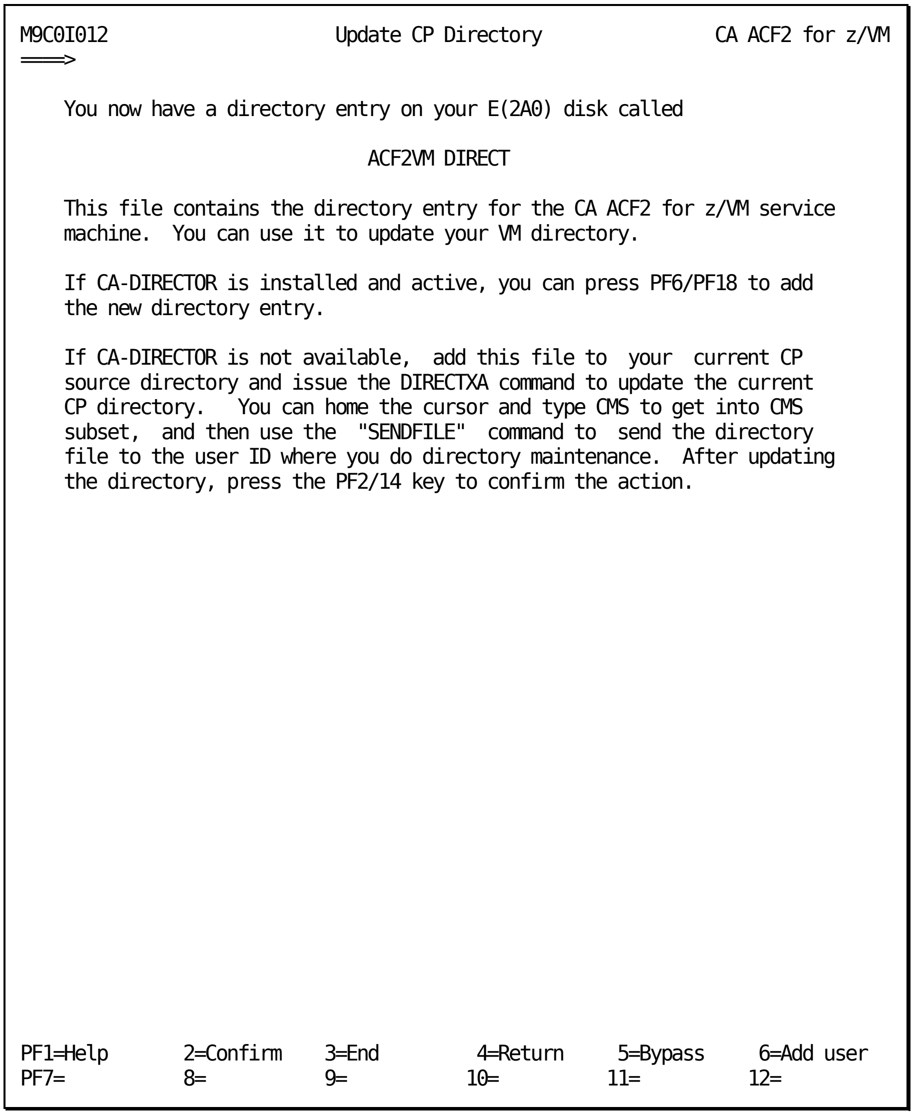
Task M9C0I012: Update CP Directory

This task updates the CP directory with the CA ACF2 for z/VM service machine ID definition.

To complete this task:

1. Select the task M9C0I012 from the Task Selection Menu.

   The following panel appears:



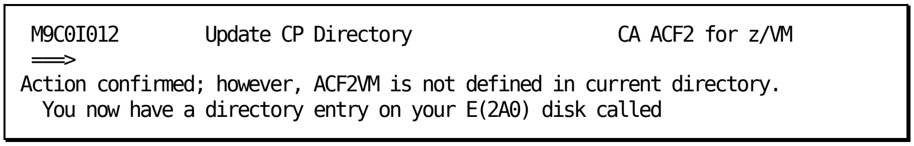
```
M9C0I012                    Update CP Directory                CA ACF2 for z/VM
===>

    You now have a directory entry on your E(2A0) disk called

                               ACF2VM DIRECT

    This file contains the directory entry for the CA ACF2 for z/VM service
    machine.  You can use it to update your VM directory.

    If CA-DIRECTOR is installed and active, you can press PF6/PF18 to add
    the new directory entry.

    If CA-DIRECTOR is not available,  add this file to  your  current CP
    source directory and issue the DIRECTXA command to update the current
    CP directory.   You can home the cursor and type CMS to get into CMS
    subset,  and then use the  "SENDFILE"  command to  send the directory
    file to the user ID where you do directory maintenance.  After updating
    the directory, press the PF2/14 key to confirm the action.















PF1=Help       2=Confirm    3=End         4=Return     5=Bypass     6=Add user
PF7=           8=           9=           10=          11=          12=
```

2. If CA-Director is available, you can press PF6 to automatically add the user ID. Otherwise, you must add the directory entry manually.

   To do this, log onto the user ID where you maintain the directory. Link to the MAINT 2A0 disk and add the ACF2VM DIRECT file to the directory. Then issue a DIRECTXA command to bring the updated directory online.

   Note: The ACF2VM DIRECT file is saved with a filemode type of 0. If you link to MAINT 2A0 read-only, this file might not be visible.

3. Press PF2 to confirm that you added the directory entry for the user ID you indicated.



```
 M9C0I012       Update CP Directory                  CA ACF2 for z/VM
 ===>
Action confirmed; however, ACF2VM is not defined in current directory.
  You now have a directory entry on your E(2A0) disk called
```

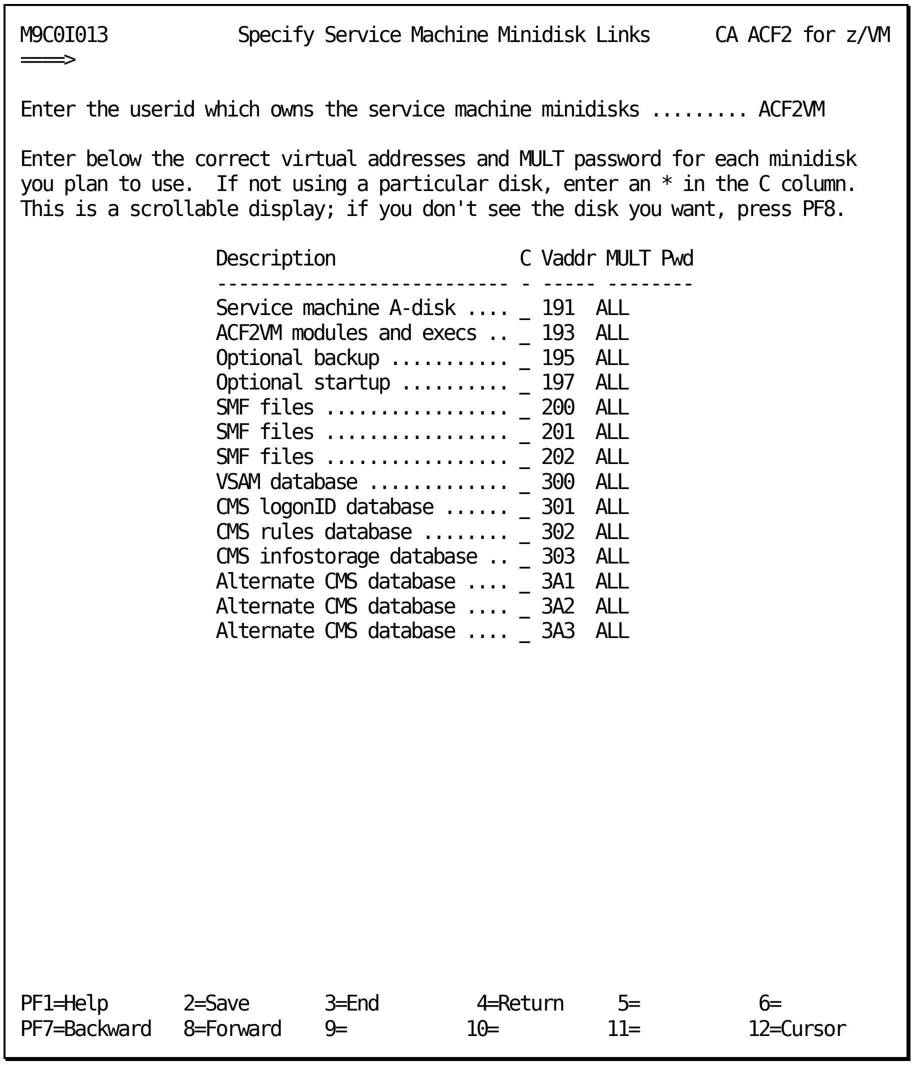
4. Press PF3 to return to the Task Selection Menu.

Task M9C0I013: Specify Service Machine Minidisk Links

This task requires you to access the service machine minidisks. To complete this task:

1. Select task M9C0I013 from the Task Selection Menu.

   The following panel appears:



```
M9C0I013          Specify Service Machine Minidisk Links      CA ACF2 for z/VM
===>

Enter the userid which owns the service machine minidisks ......... ACF2VM

Enter below the correct virtual addresses and MULT password for each minidisk
you plan to use.  If not using a particular disk, enter an * in the C column.
This is a scrollable display; if you don't see the disk you want, press PF8.

                  Description                C Vaddr MULT Pwd
                  -------------------------- - ----- --------
                  Service machine A-disk .... _ 191  ALL
                  ACF2VM modules and execs .. _ 193  ALL
                  Optional backup ........... _ 195  ALL
                  Optional startup .......... _ 197  ALL
                  SMF files ................. _ 200  ALL
                  SMF files ................. _ 201  ALL
                  SMF files ................. _ 202  ALL
                  VSAM database ............. _ 300  ALL
                  CMS logonID database ...... _ 301  ALL
                  CMS rules database ........ _ 302  ALL
                  CMS infostorage database .. _ 303  ALL
                  Alternate CMS database .... _ 3A1  ALL
                  Alternate CMS database .... _ 3A2  ALL
                  Alternate CMS database .... _ 3A3  ALL













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

2. Enter the virtual address and password for each minidisk you want to access.

3. Press PF2 to save your information.

4. Press PF3 to return to the Task Selection Menu.

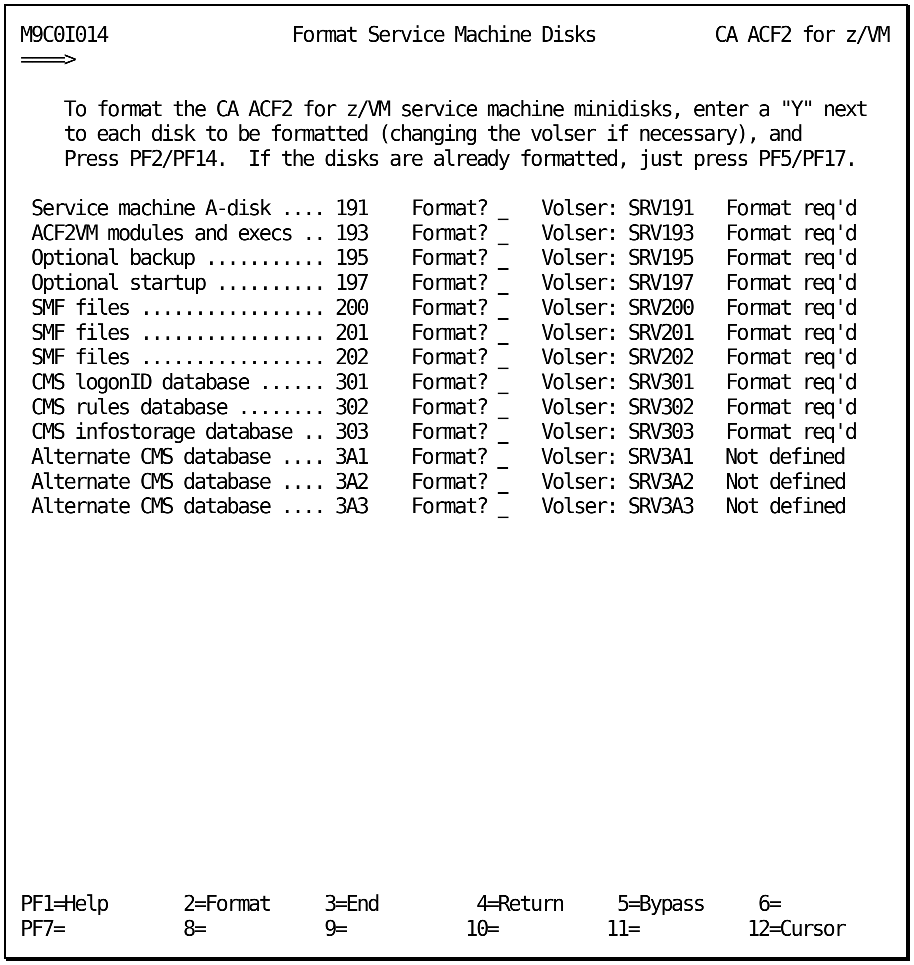
Task M9C0I014: Format the Service Machine Minidisks

This task lets you format the CA ACF2 for z/VM service machine disks. If the service machine disks are already formatted, press PF5 to bypass this task. If you want to manually format these disks, skip to the Formatting Service Machine Disks Manually section..

To complete this task:

1. Select task M9C0I014 from the Task Selection Menu.

   The following panel appears:

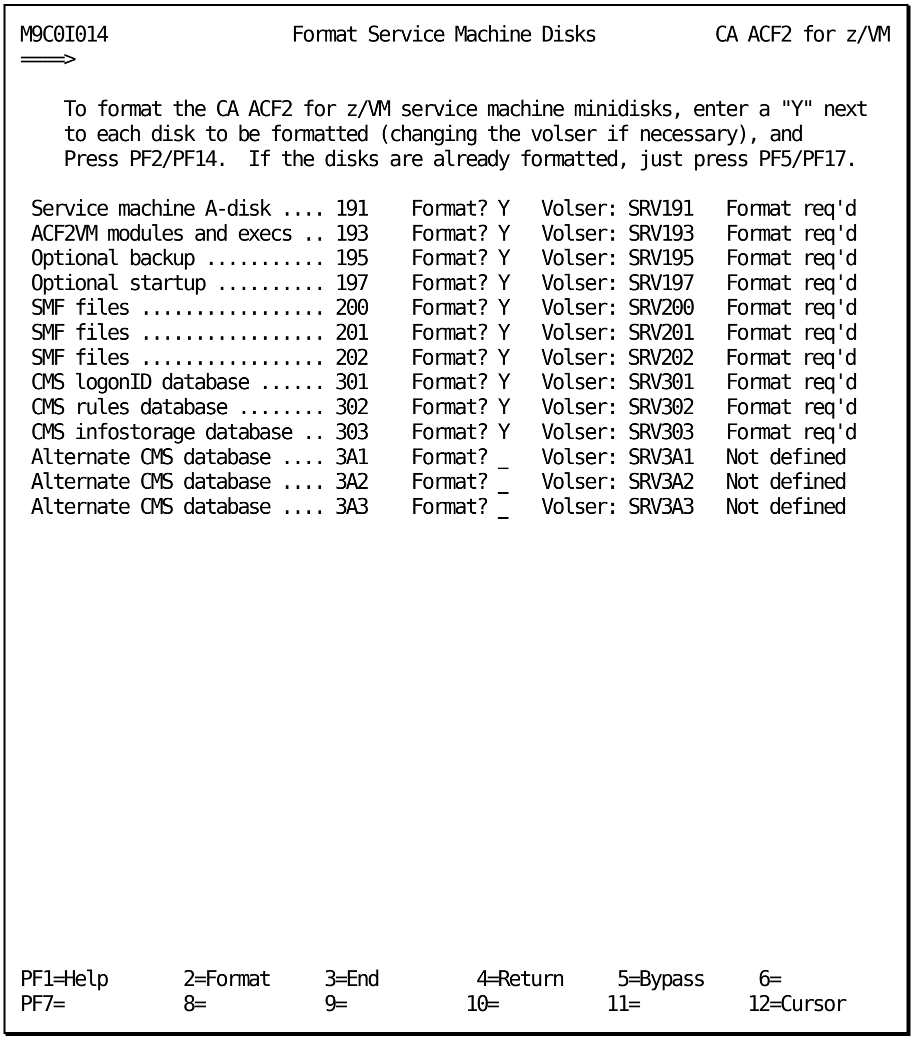


```
M9C0I014                 Format Service Machine Disks          CA ACF2 for z/VM
====>

   To format the CA ACF2 for z/VM service machine minidisks, enter a "Y" next
   to each disk to be formatted (changing the volser if necessary), and
   Press PF2/PF14.  If the disks are already formatted, just press PF5/PF17.

 Service machine A-disk .... 191    Format? _    Volser: SRV191   Format req'd
 ACF2VM modules and execs .. 193    Format? _    Volser: SRV193   Format req'd
 Optional backup ........... 195    Format? _    Volser: SRV195   Format req'd
 Optional startup .......... 197    Format? _    Volser: SRV197   Format req'd
 SMF files ................. 200    Format? _    Volser: SRV200   Format req'd
 SMF files ................. 201    Format? _    Volser: SRV201   Format req'd
 SMF files ................. 202    Format? _    Volser: SRV202   Format req'd
 CMS logonID database ...... 301    Format? _    Volser: SRV301   Format req'd
 CMS rules database ........ 302    Format? _    Volser: SRV302   Format req'd
 CMS infostorage database .. 303    Format? _    Volser: SRV303   Format req'd
 Alternate CMS database .... 3A1    Format? _    Volser: SRV3A1   Not defined
 Alternate CMS database .... 3A2    Format? _    Volser: SRV3A2   Not defined
 Alternate CMS database .... 3A3    Format? _    Volser: SRV3A3   Not defined

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

   Some disks can appear as undetermined. This means you cannot obtain a write link to the indicated disk.

You may see the DMSACP112S Z(051) device error message before the panel displays. This message is not critical. It just means that the minidisks are not formated. The above panel tries to access all of the server minidisks to determine whether they were formatted so you do not lose any data. You can ignore this message and any subsequent DMSACP112S device error messages.

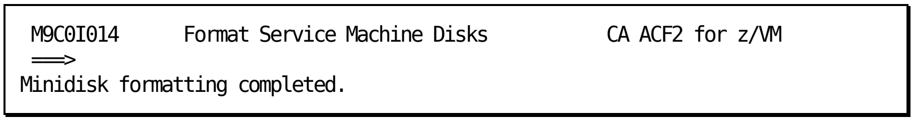
2. To format these disks, enter Y next to the Format? prompt for those disks you want to format.



```
M9C0I014                Format Service Machine Disks          CA ACF2 for z/VM
===>

    To format the CA ACF2 for z/VM service machine minidisks, enter a "Y" next
    to each disk to be formatted (changing the volser if necessary), and
    Press PF2/PF14.  If the disks are already formatted, just press PF5/PF17.

 Service machine A-disk .... 191    Format? Y   Volser: SRV191   Format req'd
 ACF2VM modules and execs .. 193    Format? Y   Volser: SRV193   Format req'd
 Optional backup ........... 195    Format? Y   Volser: SRV195   Format req'd
 Optional startup .......... 197    Format? Y   Volser: SRV197   Format req'd
 SMF files ................. 200    Format? Y   Volser: SRV200   Format req'd
 SMF files ................. 201    Format? Y   Volser: SRV201   Format req'd
 SMF files ................. 202    Format? Y   Volser: SRV202   Format req'd
 CMS logonID database ...... 301    Format? Y   Volser: SRV301   Format req'd
 CMS rules database ........ 302    Format? Y   Volser: SRV302   Format req'd
 CMS infostorage database .. 303    Format? Y   Volser: SRV303   Format req'd
 Alternate CMS database .... 3A1    Format? _   Volser: SRV3A1   Not defined
 Alternate CMS database .... 3A2    Format? _   Volser: SRV3A2   Not defined
 Alternate CMS database .... 3A3    Format? _   Volser: SRV3A3   Not defined










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

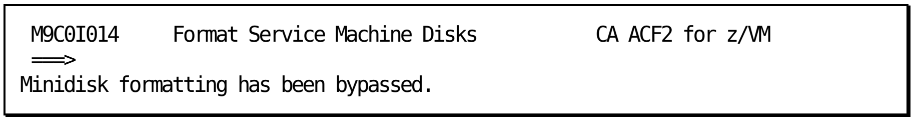
3. Press PF2 to start the formatting process.

   When processing is complete, a message appears on the panel confirming that the disks are formatted:



```
 M9C0I014     Format Service Machine Disks          CA ACF2 for z/VM
 ===>
Minidisk formatting completed.
```

4. If you press PF5 to bypass formatting of the service machine minidisks, a message appears on the panel confirming that you are bypassing this step:



```
 M9C0I014     Format Service Machine Disks          CA ACF2 for z/VM
 ===>
Minidisk formatting has been bypassed.
```

5. Press PF3 to return to the Task Selection Menu.

Formatting Service Machine Disks Manually

To format the service machine disks manually, move the cursor to the command line. Type CMS to enter the CMS subset. Then follow the steps below:

1. Type the following command and press Enter.

   ```
   LINK serverid cuu cuu M
   ```

   serverid

   The user ID of the CA ACF2 for z/VM service machine

   cuu

   The disk to format.

2. To check if the disk is already formatted, type the following command and press Enter.

   ```
   ACCESS cuu Z
   ```

   If the access works, the disk is already formatted. Compare the output of the Q V cuu and Q DISK Z commands. The disk size reported should be the same. If they are, the disk is correctly formatted. You can issue the FILELIST command for this disk to be sure the disk contains the correct files. If the disk is already formatted, skip to step 4 below.

3. To format the disk, type the following command and press Enter.

   ```
   FORMAT cuu Z
   ```

   You see the standard CMS FORMAT prompts.

4. Reply 1 or YES to the first prompt and reply with an appropriate volser to the second prompt. The disk will then be formatted.

5. Type the following command and press Enter.

   ```
   RELEASE Z (DET
   ```

   This releases the formatted disk and makes file mode Z available for the next disk you will process

6. If you need to format additional disks, return to step 1 above and proceed as before.

7. When you have formatted all the disks you need, type RETURN and press Enter to return to panel M9C0I014.

The status of this panel is still OPEN (or possibly INCOMPLETE). Press PF5 (BYPASS) to indicate that you bypassed panel formatting of these minidisks. The step will be marked complete, and you can continue with the next installation task.

Task M9C0I015: Create/Edit ACFCP CAXALOAD File

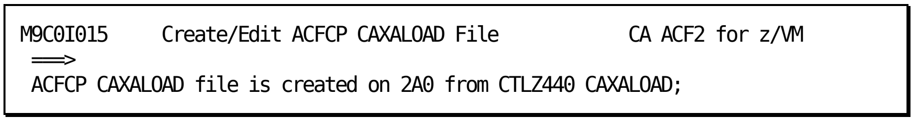
This task lets you create or edit the ACFCP CAXALOAD that is appropriate for your system.

1. Select task M9C0I015 from the Task Selection Menu.

   The following panel appears:

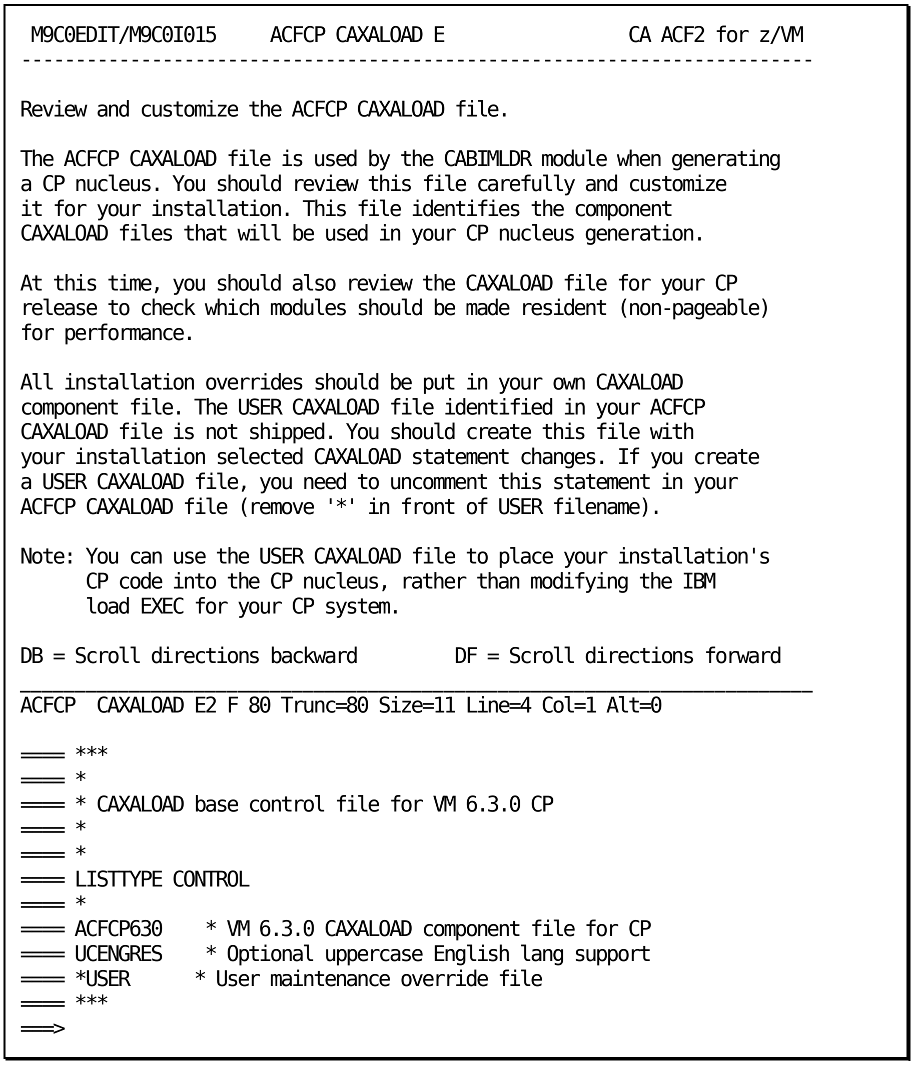
2. Enter Y to create the ACFCP CAXALOAD file or N to bypass this task.

   - If you enter N, you return to the Task Selection Menu.
   - If you enter Y, press PF2. When processing is complete, a message appears on the panel confirming that the CAXALOAD file is created:



```
M9C0I015     Create/Edit ACFCP CAXALOAD File           CA ACF2 for z/VM
 ===>
 ACFCP CAXALOAD file is created on 2A0 from CTLZ440 CAXALOAD;
```

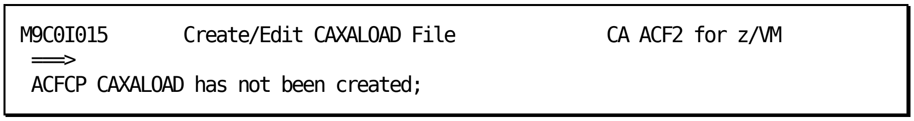
If the ACFCP CAXALOAD file already exists, press PF6 to edit this file. The following panel displays that lets you edit this file:



```
 M9C0EDIT/M9C0I015     ACFCP CAXALOAD E                CA ACF2 for z/VM
------------------------------------------------------------------------

Review and customize the ACFCP CAXALOAD file.

The ACFCP CAXALOAD file is used by the CABIMLDR module when generating
a CP nucleus. You should review this file carefully and customize
it for your installation. This file identifies the component
CAXALOAD files that will be used in your CP nucleus generation.

At this time, you should also review the CAXALOAD file for your CP
release to check which modules should be made resident (non-pageable)
for performance.

All installation overrides should be put in your own CAXALOAD
component file. The USER CAXALOAD file identified in your ACFCP
CAXALOAD file is not shipped. You should create this file with
your installation selected CAXALOAD statement changes. If you create
a USER CAXALOAD file, you need to uncomment this statement in your
ACFCP CAXALOAD file (remove '*' in front of USER filename).

Note: You can use the USER CAXALOAD file to place your installation's
      CP code into the CP nucleus, rather than modifying the IBM
      load EXEC for your CP system.

DB = Scroll directions backward        DF = Scroll directions forward
________________________________________________________________________
ACFCP  CAXALOAD E2 F 80 Trunc=80 Size=11 Line=4 Col=1 Alt=0

==== ***
==== *
==== * CAXALOAD base control file for VM 6.3.0 CP
==== *
==== *
==== LISTTYPE CONTROL
==== *
==== ACFCP630    * VM 6.3.0 CAXALOAD component file for CP
==== UCENGRES    * Optional uppercase English lang support
==== *USER      * User maintenance override file
==== ***
===>
```

3. Press PF2 to save any changes you made.

   If you attempt to edit the ACFCP CAXALOAD file before you create it, the following message appears on the panel.



```
M9C0I015      Create/Edit CAXALOAD File               CA ACF2 for z/VM
 ===>
 ACFCP CAXALOAD has not been created;
```

4. Press PF3 to return to the Task Selection Menu.

Task M9C0I017: Edit CA ACF2 for z/VM $PPF File

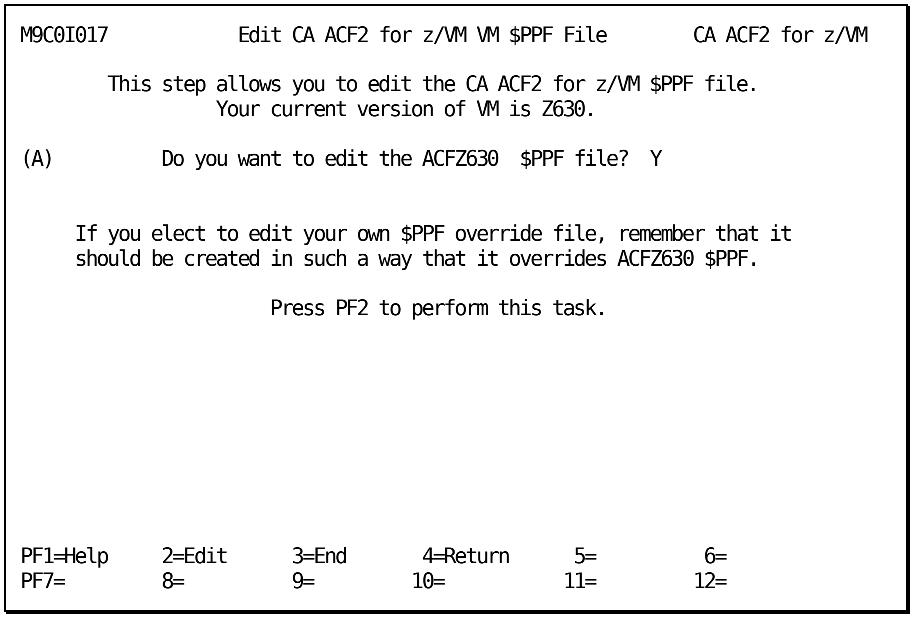
This task lets you modify the VM $PPF file.

Important!

- The CA ACF2 for z/VM sample $PPF file overrides the CP and CMS components in the IBM $PPF file. If you normally use the CPSFS and CMSSFS components, modify the CA ACF2 for z/VM $PPF file to specify CPSFS (instead of CP) and CMSSFS (instead of CMS) on the $PPF component override statements.
- To use component names other than the default names (CP, CPTEST, CMS and CMSTEST), you must manually compile the $PPF file by issuing a VMFPPF command.
- If you changed the name of the CP CNTRL file in task M9C0I006 or the name of the CMS CNTRL file in task M9C0I008, change the $PPF file to reflect the names of the CNTRL files.

1. Select task M9C0I017 from the Task Selection Menu.

   The following panel appears:

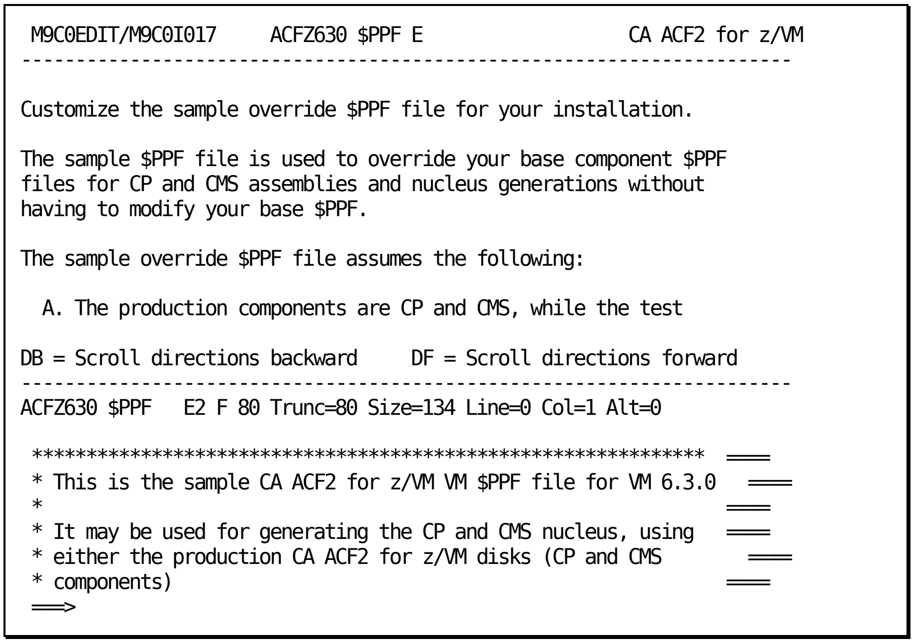


```
M9C0I017          Edit CA ACF2 for z/VM VM $PPF File       CA ACF2 for z/VM

           This step allows you to edit the CA ACF2 for z/VM $PPF file.
                       Your current version of VM is Z630.

(A)           Do you want to edit the ACFZ630  $PPF file?  Y


       If you elect to edit your own $PPF override file, remember that it
       should be created in such a way that it overrides ACFZ630 $PPF.

                             Press PF2 to perform this task.








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

2. (Optional) Override the $PPF file name by typing over the ACFZ630 entry and then entering Y at the end of the line.

3. Press PF2 to edit the $PPF file.

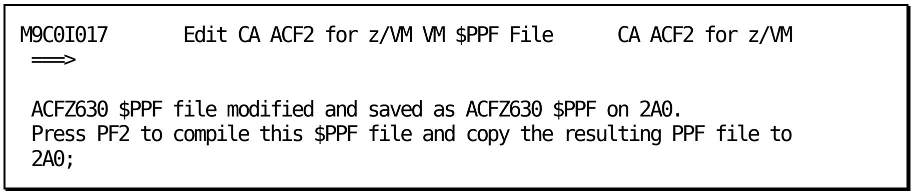
   You see the following panel:



```
 M9C0EDIT/M9C0I017     ACFZ630 $PPF E                  CA ACF2 for z/VM
---------------------------------------------------------------------

Customize the sample override $PPF file for your installation.

The sample $PPF file is used to override your base component $PPF
files for CP and CMS assemblies and nucleus generations without
having to modify your base $PPF.

The sample override $PPF file assumes the following:

  A. The production components are CP and CMS, while the test

DB = Scroll directions backward     DF = Scroll directions forward
---------------------------------------------------------------------
ACFZ630 $PPF   E2 F 80 Trunc=80 Size=134 Line=0 Col=1 Alt=0

 ************************************************************  ====
 * This is the sample CA ACF2 for z/VM VM $PPF file for VM 6.3.0   ====
 *                                                              ====
 * It may be used for generating the CP and CMS nucleus, using  ====
 * either the production CA ACF2 for z/VM disks (CP and CMS       ====
 * components)                                                  ====
 ===>
```

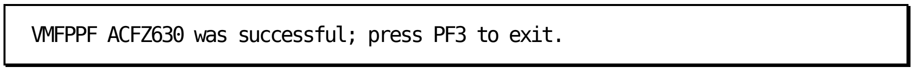
4. Make modifications as necessary to the $PPF file, then enter the FILE command to save your changes.

   When processing is complete, a message appears on the panel, confirming that you modified and saved the $PPF file:



```
M9C0I017      Edit CA ACF2 for z/VM VM $PPF File      CA ACF2 for z/VM
 ===>

 ACFZ630 $PPF file modified and saved as ACFZ630 $PPF on 2A0.
 Press PF2 to compile this $PPF file and copy the resulting PPF file to
 2A0;
```

5. Press PF2 to compile the PPF file into a final PPF file that is used for system generation. You see the following message:



```
 VMFPPF ACFZ630 was successful; press PF3 to exit.
```

6. Press PF3 to return to the Task Selection Menu if the VMFPPF processing terminated without errors. If the VMFPPF terminated with errors, repeat the editing and compiling until the VMFPPF processing terminates successfully.

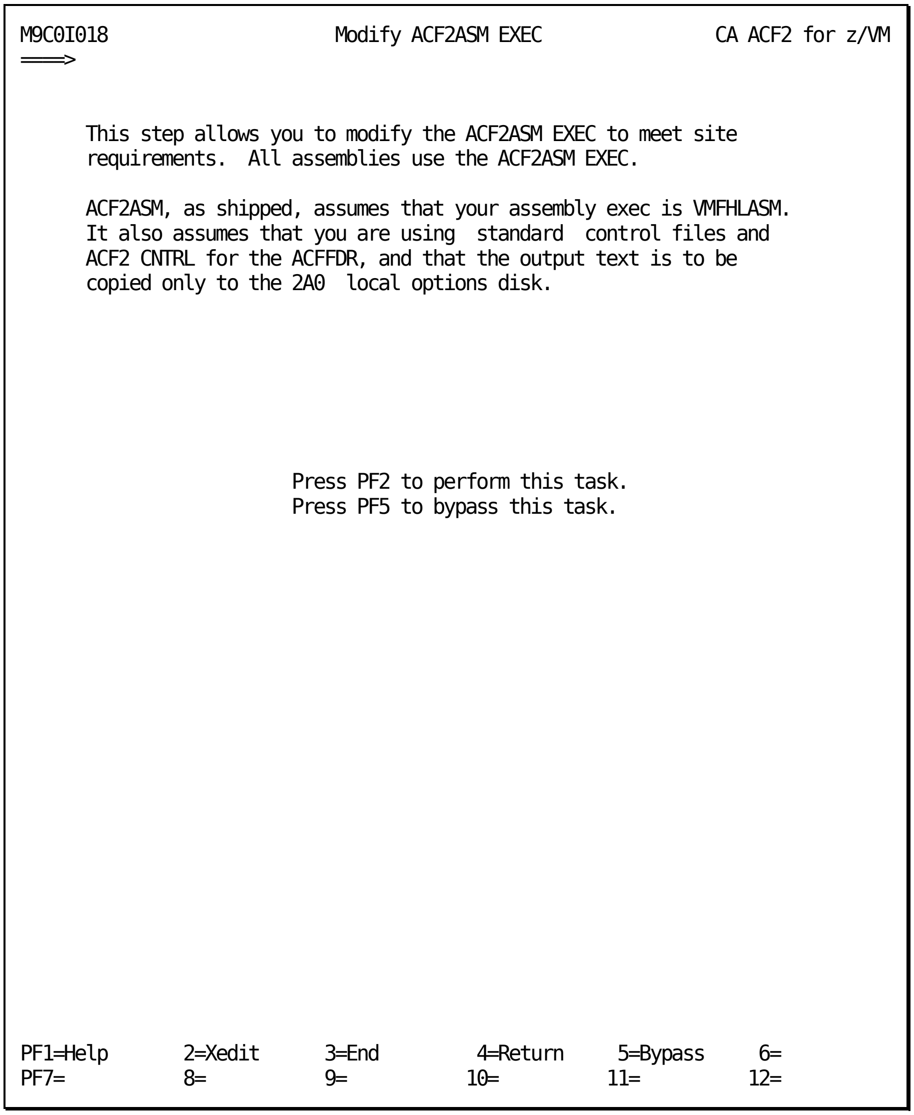
Task M9C0I018: Modify ACF2ASM EXEC

This task lets you modify your ACF2ASM EXEC file.

To complete this task:

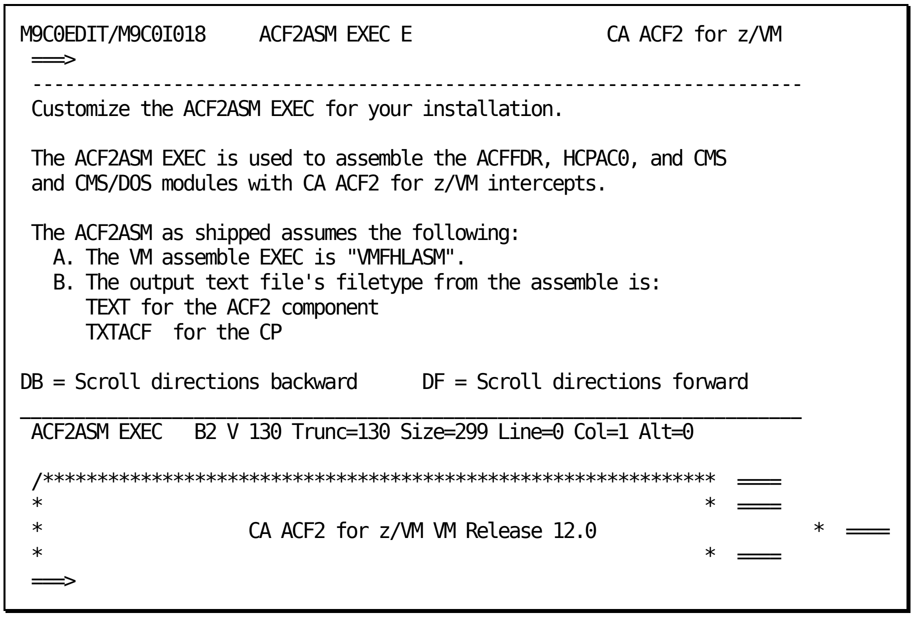
1. Select task M9C0I018 from the Task Selection Menu.
2. Press PF2 to edit the ACF2ASM EXEC.

   The following panel appears:



```
M9C0I018                      Modify ACF2ASM EXEC                CA ACF2 for z/VM
====>


      This step allows you to modify the ACF2ASM EXEC to meet site
      requirements.  All assemblies use the ACF2ASM EXEC.

      ACF2ASM, as shipped, assumes that your assembly exec is VMFHLASM.
      It also assumes that you are using  standard  control files and
      ACF2 CNTRL for the ACFFDR, and that the output text is to be
      copied only to the 2A0  local options disk.






                         Press PF2 to perform this task.
                         Press PF5 to bypass this task.




















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

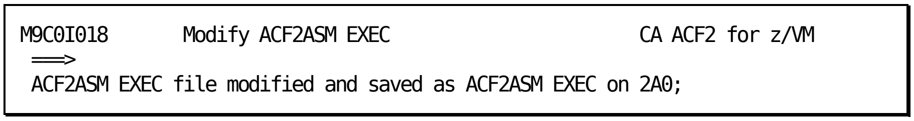
3. Press PF5 to bypass this task or press PF2 to display the ACF2ASM EXEC:



```
M9C0EDIT/M9C0I018     ACF2ASM EXEC E                CA ACF2 for z/VM
 ===>
 ----------------------------------------------------------------------
 Customize the ACF2ASM EXEC for your installation.

 The ACF2ASM EXEC is used to assemble the ACFFDR, HCPAC0, and CMS
 and CMS/DOS modules with CA ACF2 for z/VM intercepts.

 The ACF2ASM as shipped assumes the following:
   A. The VM assemble EXEC is "VMFHLASM".
   B. The output text file's filetype from the assemble is:
      TEXT for the ACF2 component
      TXTACF  for the CP

DB = Scroll directions backward      DF = Scroll directions forward
_______________________________________________________________________
 ACF2ASM EXEC   B2 V 130 Trunc=130 Size=299 Line=0 Col=1 Alt=0

 /**************************************************************  ====
 *                                                             *  ====
 *                  CA ACF2 for z/VM VM Release 12.0                      *  ====
 *                                                             *  ====
 ===>
```

4. After you modify this exec, press PF2 to save your changes.

   When processing is complete, a message appears on the panel confirming that you modified and saved the ACF2ASM EXEC:



```
M9C0I018      Modify ACF2ASM EXEC                        CA ACF2 for z/VM
 ===>
 ACF2ASM EXEC file modified and saved as ACF2ASM EXEC on 2A0;
```

5. Press PF3 to return to the Task Selection Menu.

Task M9C0I019: Modify the Service Machine PROFILE EXEC

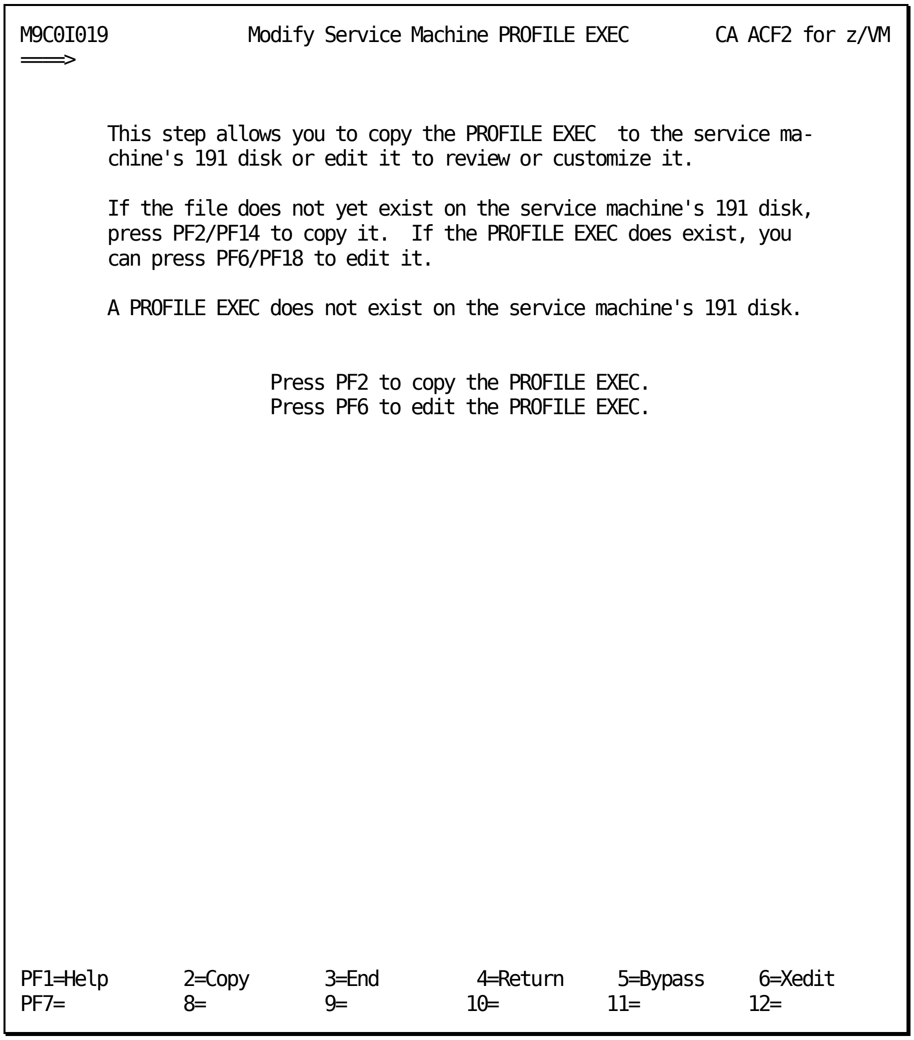
This task lets you modify the PROFILE EXEC for the service machine.

To complete this task:

1. Select task M9C0I019 from the Task Selection Menu.

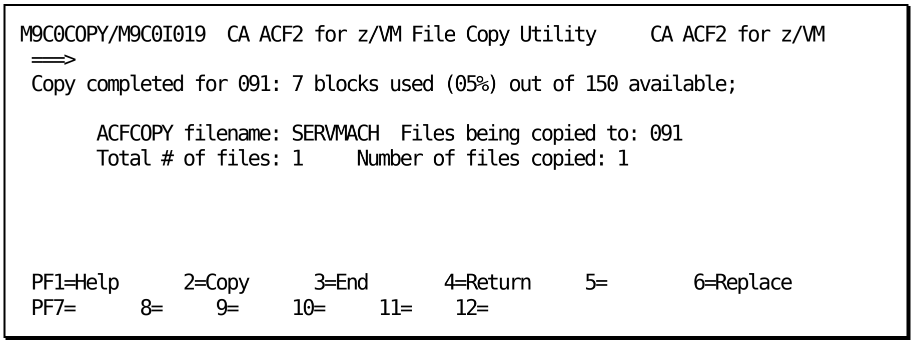
   The following panel appears:

2. Press PF2 to copy the PROFILE EXEC or PF6 to edit it.



```
M9C0I019           Modify Service Machine PROFILE EXEC       CA ACF2 for z/VM
===>


        This step allows you to copy the PROFILE EXEC  to the service ma-
        chine's 191 disk or edit it to review or customize it.

        If the file does not yet exist on the service machine's 191 disk,
        press PF2/PF14 to copy it.  If the PROFILE EXEC does exist, you
        can press PF6/PF18 to edit it.

        A PROFILE EXEC does not exist on the service machine's 191 disk.


                       Press PF2 to copy the PROFILE EXEC.
                       Press PF6 to edit the PROFILE EXEC.

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

If you elect to copy the exec, the following panel appears.



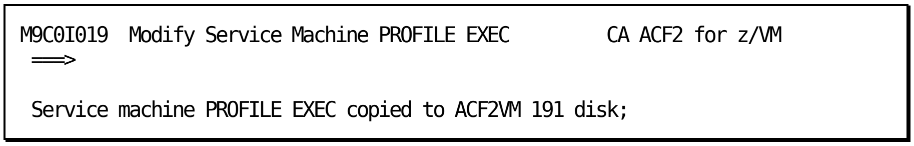
```
M9C0COPY/M9C0I019  CA ACF2 for z/VM File Copy Utility     CA ACF2 for z/VM
 ===>
 Copy completed for 091: 7 blocks used (05%) out of 150 available;

       ACFCOPY filename: SERVMACH  Files being copied to: 091
       Total # of files: 1     Number of files copied: 1




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

Note: The above message appears even if you bypassed copying the PROFILE EXEC.

This panel tells you the following information:

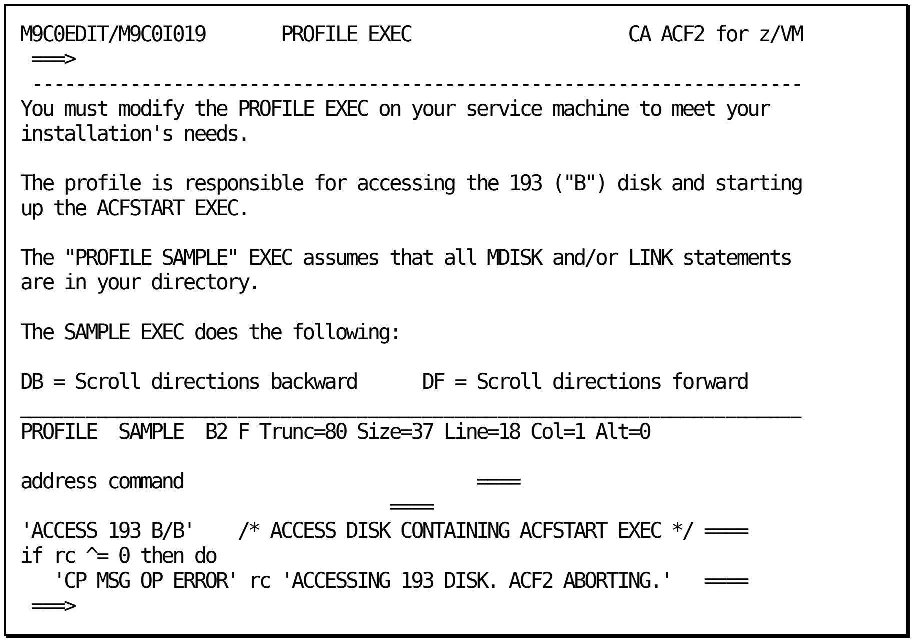
- What control file is used to control the copy
- What disk the files are being copied to
- How many files will be copied
- How many files were copied so far.
- The value for Number of files copied: changes as you watch the panel.

When processing is complete, a message appears on the panel confirming that you copied the PROFILE EXEC to the ACF2VM 191 disk.



```
M9C0I019  Modify Service Machine PROFILE EXEC        CA ACF2 for z/VM
 ===>

 Service machine PROFILE EXEC copied to ACF2VM 191 disk;
```

If you elect to edit the PROFILE EXEC, it displays over the bottom half of the panel:



```
M9C0EDIT/M9C0I019       PROFILE EXEC                  CA ACF2 for z/VM
 ===>
 ----------------------------------------------------------------------
You must modify the PROFILE EXEC on your service machine to meet your
installation's needs.

The profile is responsible for accessing the 193 ("B") disk and starting
up the ACFSTART EXEC.

The "PROFILE SAMPLE" EXEC assumes that all MDISK and/or LINK statements
are in your directory.

The SAMPLE EXEC does the following:

DB = Scroll directions backward      DF = Scroll directions forward
________________________________________________________________________
PROFILE  SAMPLE  B2 F Trunc=80 Size=37 Line=18 Col=1 Alt=0

address command                                ====
                                       ====
'ACCESS 193 B/B'    /* ACCESS DISK CONTAINING ACFSTART EXEC */ ====
if rc ^= 0 then do
   'CP MSG OP ERROR' rc 'ACCESSING 193 DISK. ACF2 ABORTING.'   ====
 ===>
```

After you make your changes, press PF2 to save them.

1. Press PF3 to return to the Task Selection Menu.

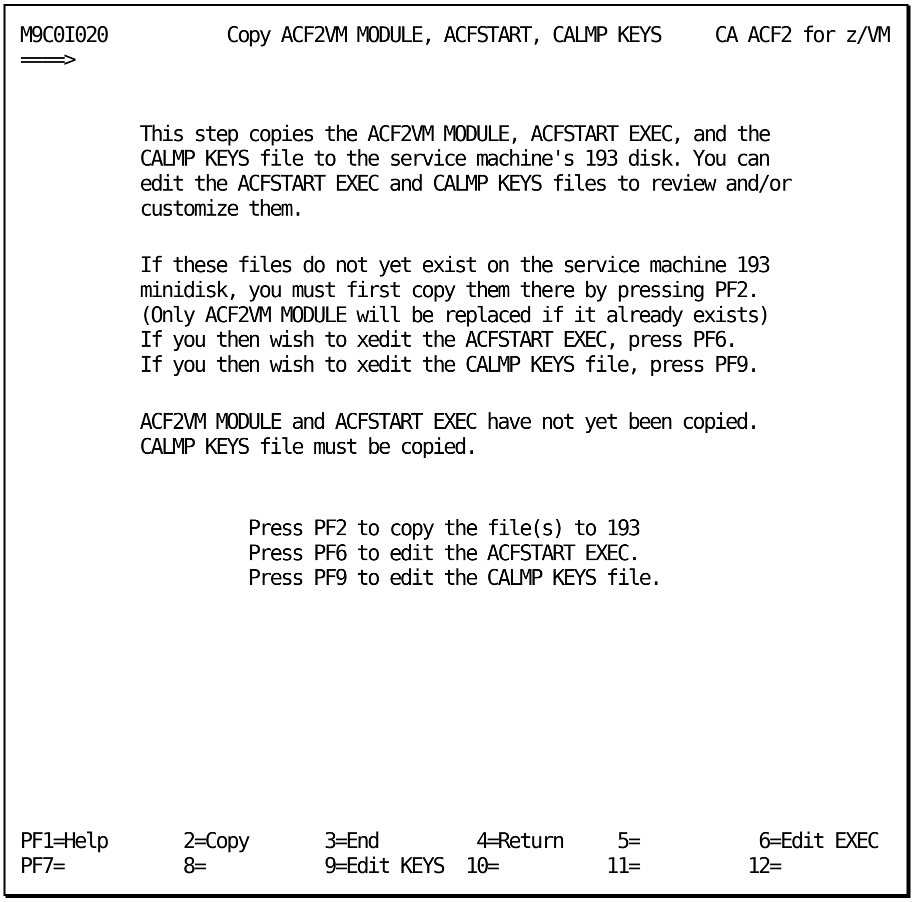
Task M9C0I020: Copy ACF2VM MODULE, ACFSTART EXEC, and CALMP KEYS

This task lets you copy the ACF2VM MODULE, ACFSTART EXEC, and CALMP KEYS to your service machine's disk.

To complete this task:

1. Select task M9C0I020 from the Task Selection Menu.

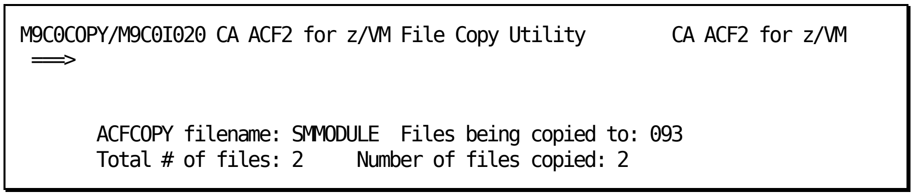
   The following panel appears:



```
M9C0I020          Copy ACF2VM MODULE, ACFSTART, CALMP KEYS     CA ACF2 for z/VM
===>


           This step copies the ACF2VM MODULE, ACFSTART EXEC, and the
           CALMP KEYS file to the service machine's 193 disk. You can
           edit the ACFSTART EXEC and CALMP KEYS files to review and/or
           customize them.

           If these files do not yet exist on the service machine 193
           minidisk, you must first copy them there by pressing PF2.
           (Only ACF2VM MODULE will be replaced if it already exists)
           If you then wish to xedit the ACFSTART EXEC, press PF6.
           If you then wish to xedit the CALMP KEYS file, press PF9.

           ACF2VM MODULE and ACFSTART EXEC have not yet been copied.
           CALMP KEYS file must be copied.


                     Press PF2 to copy the file(s) to 193
                     Press PF6 to edit the ACFSTART EXEC.
                     Press PF9 to edit the CALMP KEYS file.









PF1=Help       2=Copy       3=End         4=Return     5=            6=Edit EXEC
PF7=           8=           9=Edit KEYS  10=          11=           12=
```

2. Press PF2 to copy the files or PF6 to edit the ACFSTART EXEC.

   If you elect to copy the files, the following panel displays.



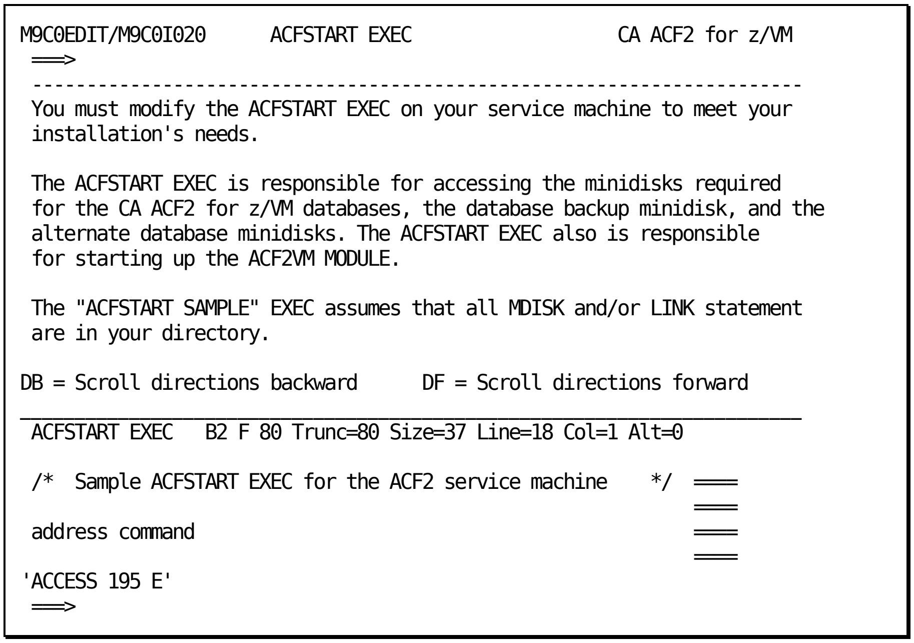
```
M9C0COPY/M9C0I020 CA ACF2 for z/VM File Copy Utility       CA ACF2 for z/VM
 ===>


      ACFCOPY filename: SMMODULE  Files being copied to: 093
      Total # of files: 2     Number of files copied: 2
```

   Note the following:

   - Because the files are copied very quickly, you may not see this status display panel.
   - The ACF2VM MODULE will always be copied, replacing any existing ACF2VM MODULE after saving it as filetype MODOLD. If the ACFSTART EXEC or CALMP KEYS files already exist on the target disk, they will not be copied.

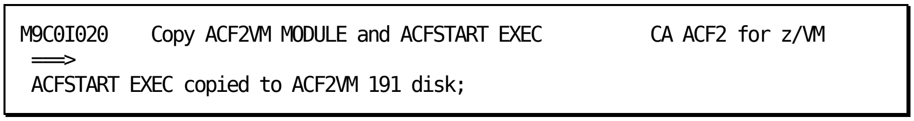
3. After the files are copied, press PF6 to edit the ACFSTART EXEC.

   It displays over the bottom half of the panel:



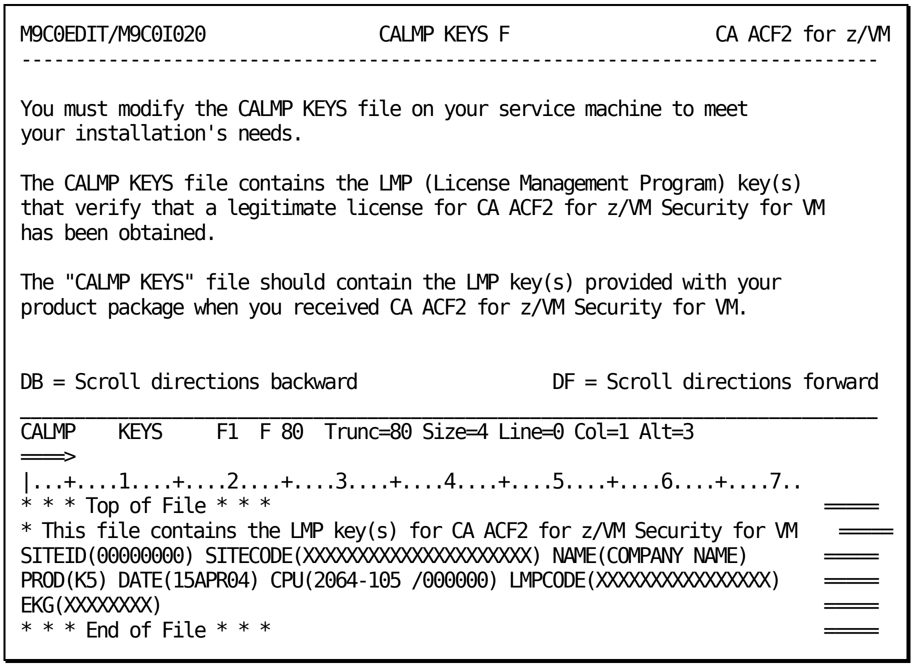
```
M9C0EDIT/M9C0I020      ACFSTART EXEC                  CA ACF2 for z/VM
 ===>
 ----------------------------------------------------------------------
 You must modify the ACFSTART EXEC on your service machine to meet your
 installation's needs.

 The ACFSTART EXEC is responsible for accessing the minidisks required
 for the CA ACF2 for z/VM databases, the database backup minidisk, and the
 alternate database minidisks. The ACFSTART EXEC also is responsible
 for starting up the ACF2VM MODULE.

 The "ACFSTART SAMPLE" EXEC assumes that all MDISK and/or LINK statement
 are in your directory.

DB = Scroll directions backward      DF = Scroll directions forward
________________________________________________________________________
 ACFSTART EXEC   B2 F 80 Trunc=80 Size=37 Line=18 Col=1 Alt=0

 /*  Sample ACFSTART EXEC for the ACF2 service machine    */  ====
                                                             ====
 address command                                             ====
                                                             ====
'ACCESS 195 E'
 ===>
```

   When processing is complete, a message appears on the panel confirming that you changed and saved the PROFILE EXEC.



```
M9C0I020    Copy ACF2VM MODULE and ACFSTART EXEC          CA ACF2 for z/VM
 ===>
 ACFSTART EXEC copied to ACF2VM 191 disk;
```

4. Press PF9 to edit CALMP KEYS file.



```
M9C0EDIT/M9C0I020                CALMP KEYS F                 CA ACF2 for z/VM
--------------------------------------------------------------------------------

You must modify the CALMP KEYS file on your service machine to meet
your installation's needs.

The CALMP KEYS file contains the LMP (License Management Program) key(s)
that verify that a legitimate license for CA ACF2 for z/VM Security for VM
has been obtained.

The "CALMP KEYS" file should contain the LMP key(s) provided with your
product package when you received CA ACF2 for z/VM Security for VM.


DB = Scroll directions backward                  DF = Scroll directions forward
________________________________________________________________________________
CALMP    KEYS     F1  F 80  Trunc=80 Size=4 Line=0 Col=1 Alt=3
===>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
* * * Top of File * * *                                                    =====
* This file contains the LMP key(s) for CA ACF2 for z/VM Security for VM     =====
SITEID(00000000) SITECODE(XXXXXXXXXXXXXXXXXXXXXX) NAME(COMPANY NAME)       =====
PROD(K5) DATE(15APR04) CPU(2064-105 /000000) LMPCODE(XXXXXXXXXXXXXXXXX)    =====
EKG(XXXXXXXX)                                                              =====
* * * End of File * * *                                                    =====
```

5. Press PF3 to return to the Task Selection Menu.

Task M9C0I021: Modify MLAVM, USERLID, and USERXLID COPY

This task builds the ACF2USER MACLIB. The ACF2USER MACLIB contains several MACRO and COPY files, including the MLAVM MACRO and the USERLID and USERXLID COPY files. The MLAVM MACRO defines the layout of the standard VM minilid (MLID). The USERLID COPY and USERXLID COPY files contain any user-defined fields that are included in the CA ACF2 for z/VM logonid record. The LIDREC MACRO copies these two COPY files. They become part of the LIDREC DSECT that maps the logonid record. USERLID is for fields in the first user section of the logonid record. USERXLID is for fields in the second user section of the logonid record.

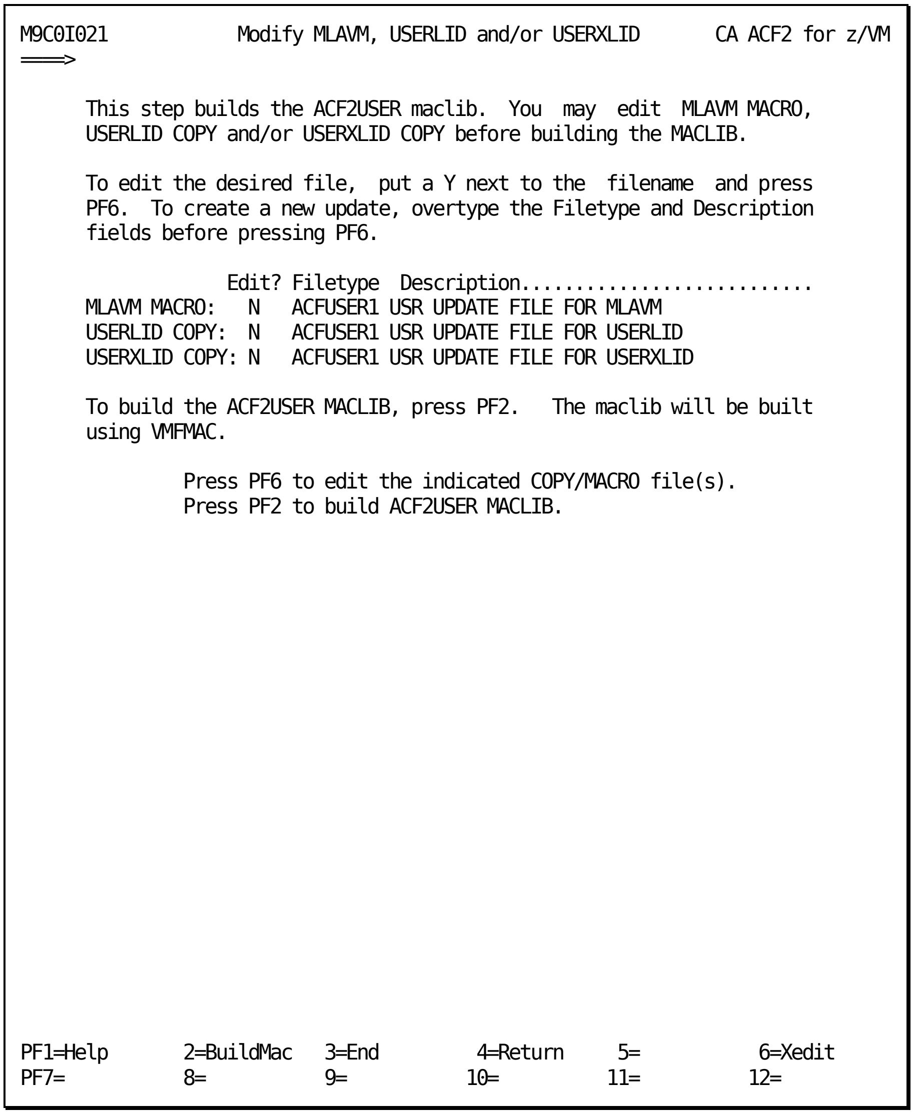
If you are adding @CFDE entries in the ACFFDR for user-defined fields, you must add the corresponding symbolic label to either USERLID or USERXLID and then rebuild the ACF2USER MACLIB. Once it is built, you must assemble the ACFFDR (done in a later step). This step, M9C0I021, modifies these files and rebuilds the maclib.

See "Field Definition Record" for more information on the ACFFDR, which uses these macros.

To complete this task:

1. Select task M9C0I021 from the Task Selection Menu.

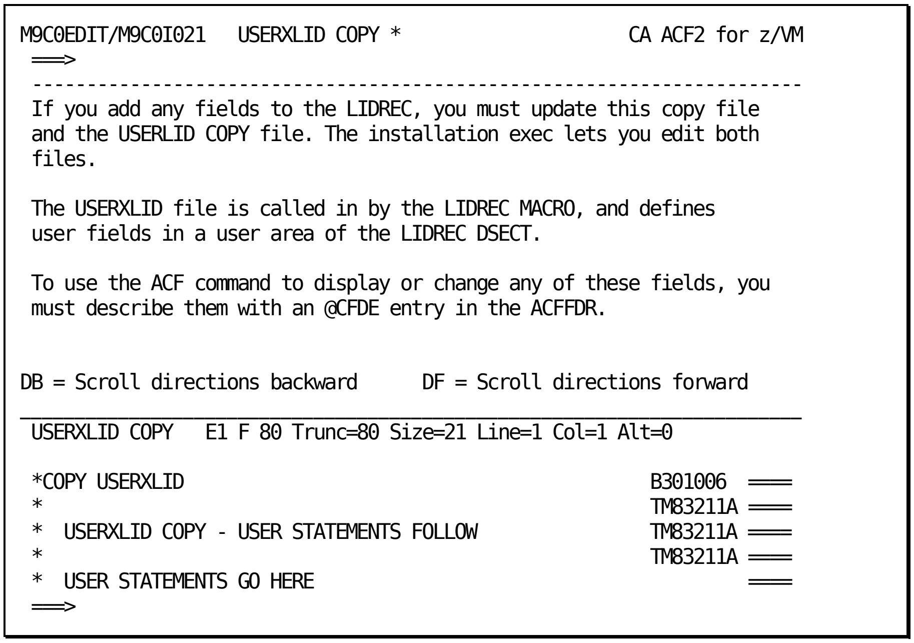
   The following panel appears:



```
M9C0I021           Modify MLAVM, USERLID and/or USERXLID       CA ACF2 for z/VM
===>

         This step builds the ACF2USER maclib.  You  may  edit  MLAVM MACRO,
         USERLID COPY and/or USERXLID COPY before building the MACLIB.

         To edit the desired file,  put a Y next to the  filename  and press
         PF6.  To create a new update, overtype the Filetype and Description
         fields before pressing PF6.

                     Edit? Filetype  Description...........................
         MLAVM MACRO:   N   ACFUSER1 USR UPDATE FILE FOR MLAVM
         USERLID COPY:  N   ACFUSER1 USR UPDATE FILE FOR USERLID
         USERXLID COPY: N   ACFUSER1 USR UPDATE FILE FOR USERXLID

         To build the ACF2USER MACLIB, press PF2.   The maclib will be built
         using VMFMAC.

                       Press PF6 to edit the indicated COPY/MACRO file(s).
                       Press PF2 to build ACF2USER MACLIB.

















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

2. Type Y to indicate you want to modify the MLAVM macro, the USERLID COPY file, and the USERXLID COPY file.

3. Press PF6 to edit the files you selected. The first file you selected displays over the bottom of the panel. In the example below, the file being modified is the USERXLID COPY file:

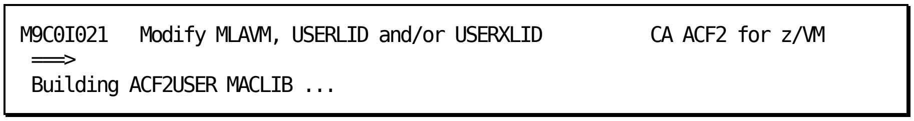


```
M9C0EDIT/M9C0I021   USERXLID COPY *                    CA ACF2 for z/VM
 ===>
 ---------------------------------------------------------------------
 If you add any fields to the LIDREC, you must update this copy file
 and the USERLID COPY file. The installation exec lets you edit both
 files.

 The USERXLID file is called in by the LIDREC MACRO, and defines
 user fields in a user area of the LIDREC DSECT.

 To use the ACF command to display or change any of these fields, you
 must describe them with an @CFDE entry in the ACFFDR.


DB = Scroll directions backward      DF = Scroll directions forward
______________________________________________________________________
 USERXLID COPY   E1 F 80 Trunc=80 Size=21 Line=1 Col=1 Alt=0

 *COPY USERXLID                                        B301006  ====
 *                                                     TM83211A ====
 *  USERXLID COPY - USER STATEMENTS FOLLOW             TM83211A ====
 *                                                     TM83211A ====
 *  USER STATEMENTS GO HERE                                     ====
 ===>
```

4. After you make your changes, press PF2 to save them.

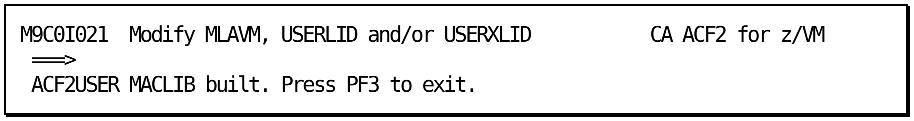
5. Press PF2 to create the ACF2USER MACLIB.

   A message appears on the panel confirming that CA-ACTIVATOR is building the ACF2USER MACLIB:



```
M9C0I021   Modify MLAVM, USERLID and/or USERXLID          CA ACF2 for z/VM
 ===>
 Building ACF2USER MACLIB ...
```

   When processing is complete, a message appears on the panel confirming that the ACF2USER MACLIB is built:



```
M9C0I021  Modify MLAVM, USERLID and/or USERXLID           CA ACF2 for z/VM
 ===>
 ACF2USER MACLIB built. Press PF3 to exit.
```

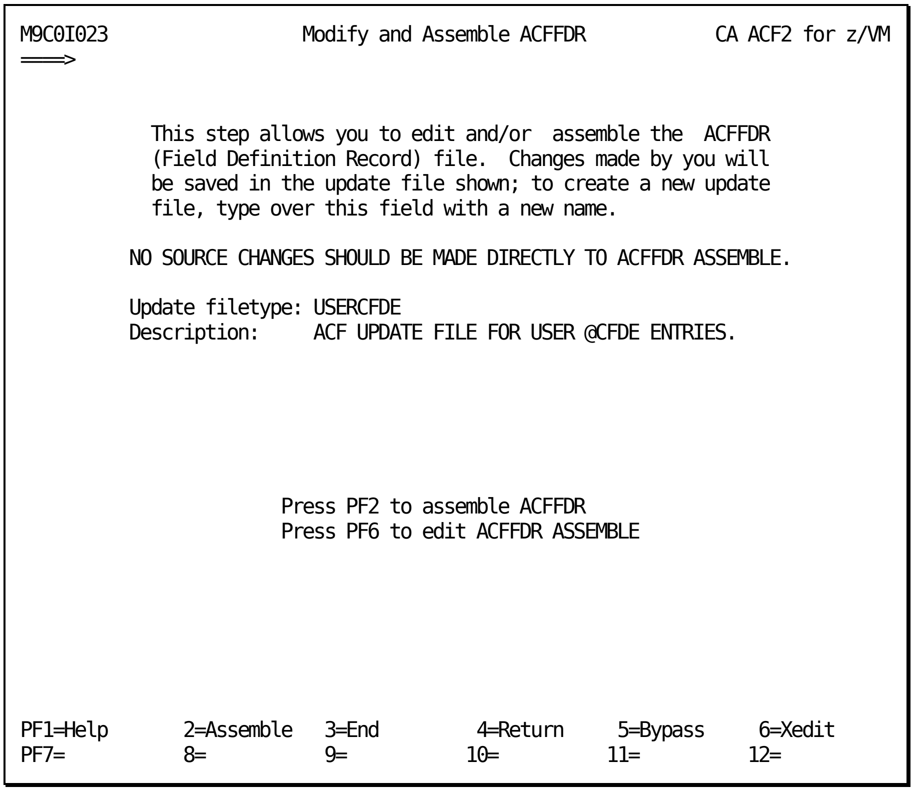
6. Press PF3 to return to the Task Selection Menu.

Task M9C0I023: Modify and Assemble ACFFDR

This task lets you modify and assemble the ACFFDR. See "Field Definition Record" for more information on the ACFFDR. To complete this task:

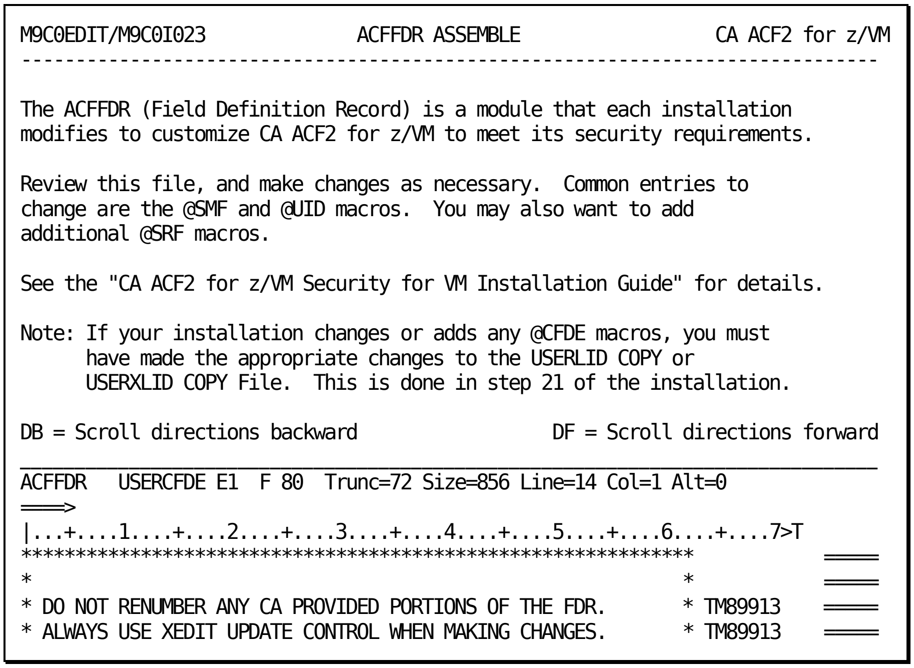
1. Select task M9C0I023 from the Task Selection Menu.

   The following panel appears:



```
M9C0I023                  Modify and Assemble ACFFDR           CA ACF2 for z/VM
===>


            This step allows you to edit and/or  assemble the  ACFFDR
            (Field Definition Record) file.  Changes made by you will
            be saved in the update file shown; to create a new update
            file, type over this field with a new name.

          NO SOURCE CHANGES SHOULD BE MADE DIRECTLY TO ACFFDR ASSEMBLE.

          Update filetype: USERCFDE
          Description:     ACF UPDATE FILE FOR USER @CFDE ENTRIES.







                        Press PF2 to assemble ACFFDR
                        Press PF6 to edit ACFFDR ASSEMBLE








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

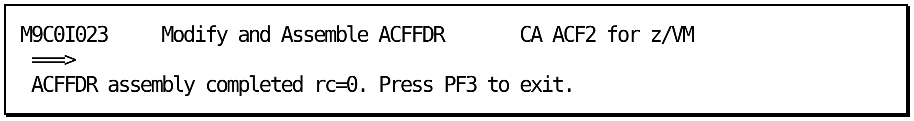
2. Press PF6 to update the ACFFDR ASSEMBLE file. You should **not** edit the source file directly. The ACFFDR ASSEMBLE file is overlayed on the bottom of the panel:



```
M9C0EDIT/M9C0I023            ACFFDR ASSEMBLE                  CA ACF2 for z/VM
------------------------------------------------------------------------------

The ACFFDR (Field Definition Record) is a module that each installation
modifies to customize CA ACF2 for z/VM to meet its security requirements.

Review this file, and make changes as necessary.  Common entries to
change are the @SMF and @UID macros.  You may also want to add
additional @SRF macros.

See the "CA ACF2 for z/VM Security for VM Installation Guide" for details.

Note: If your installation changes or adds any @CFDE macros, you must
      have made the appropriate changes to the USERLID COPY or
      USERXLID COPY File.  This is done in step 21 of the installation.

DB = Scroll directions backward                  DF = Scroll directions forward
______________________________________________________________________________
ACFFDR   USERCFDE E1  F 80  Trunc=72 Size=856 Line=14 Col=1 Alt=0
====>
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7>T
**************************************************************          =====
*                                                           *           =====
* DO NOT RENUMBER ANY CA PROVIDED PORTIONS OF THE FDR.      * TM89913   =====
* ALWAYS USE XEDIT UPDATE CONTROL WHEN MAKING CHANGES.      * TM89913   =====
```

3. If you are converting from a previous CA ACF2 for z/VM release, you must refit your updates to the ACFFDR because the FDR has been restructured.

4. If you press PF2 to assemble the ACFFDR, a message appears on the panel confirming the assembly:



```
M9C0I023     Modify and Assemble ACFFDR      CA ACF2 for z/VM
 ====>
 ACFFDR assembly completed rc=0. Press PF3 to exit.
```

5. If you receive a return code other than 0, correct the error and reexecute M9C0I023.

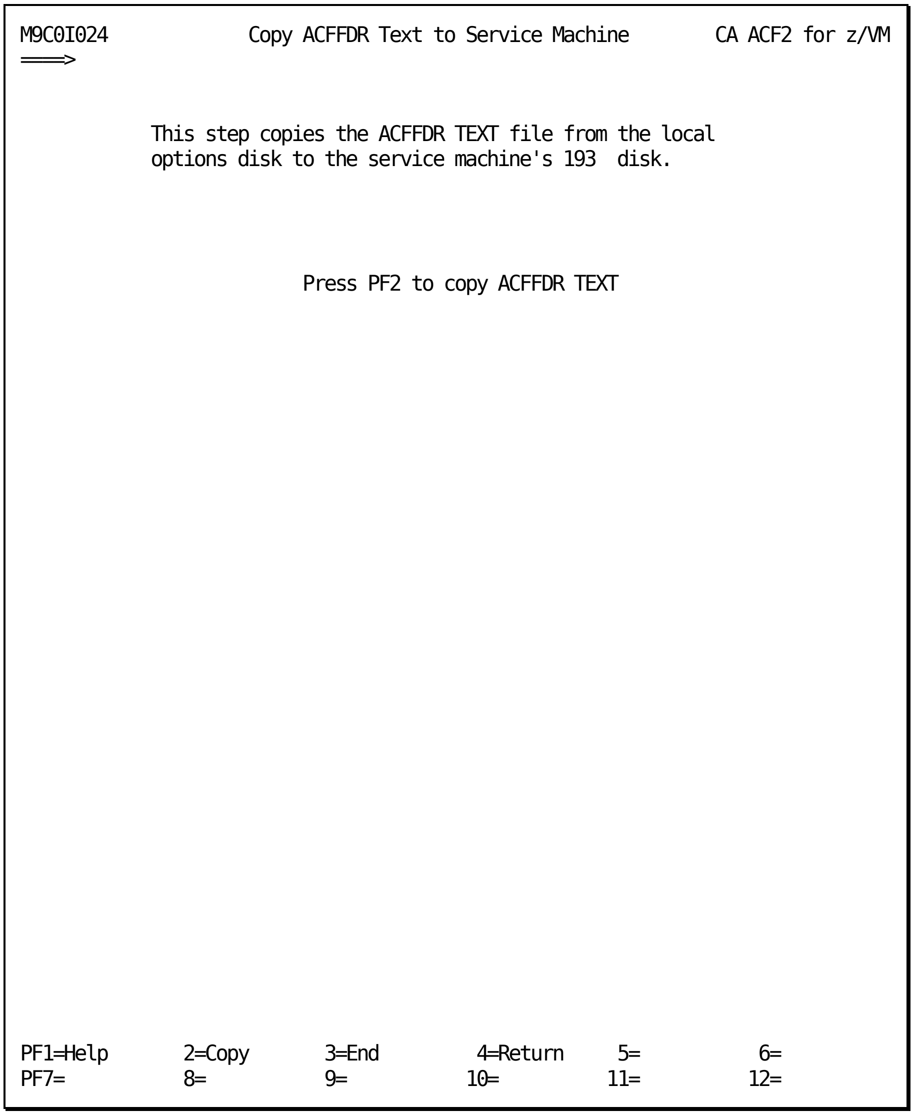
6. Press PF3 to return to the Task Selection Menu.

Task M9C0I024: Copy ACFFDR TEXT to Service Machine

This task copies the ACFFDR TEXT file to your service machine. To complete this task:

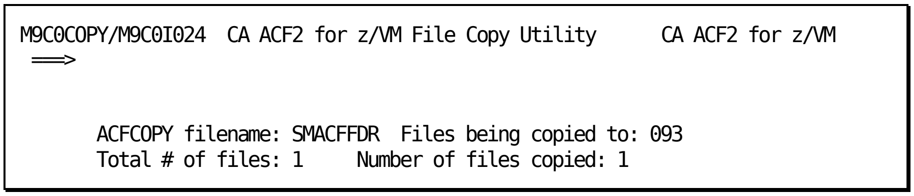
1. Select task M9C0I024 from the Task Selection Menu.

   The following panel appears:



```
M9C0I024            Copy ACFFDR Text to Service Machine       CA ACF2 for z/VM
===>


            This step copies the ACFFDR TEXT file from the local
            options disk to the service machine's 193  disk.



                          Press PF2 to copy ACFFDR TEXT





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

2. Press PF2 to copy the exec. The following panel appears. **Note**: Because the ACFFDR TEXT file is copied very quickly, you may not see this status display panel.

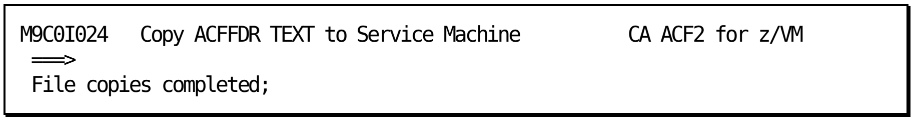


```
M9C0COPY/M9C0I024  CA ACF2 for z/VM File Copy Utility      CA ACF2 for z/VM
 ===>


       ACFCOPY filename: SMACFFDR  Files being copied to: 093
       Total # of files: 1     Number of files copied: 1
```

This panel tells you the following information:

- What control file is used to control the copy
- What disk the files are being copied to
- How many files will be copied
- How many files were copied so far.
- The value for Number of files copied: changes as you watch the panel.

When processing is complete, a message appears on the panel confirming that all the files are copied.



```
M9C0I024   Copy ACFFDR TEXT to Service Machine          CA ACF2 for z/VM
 ===>
 File copies completed;
```

3. Press PF3 to return to the Task Selection Menu.

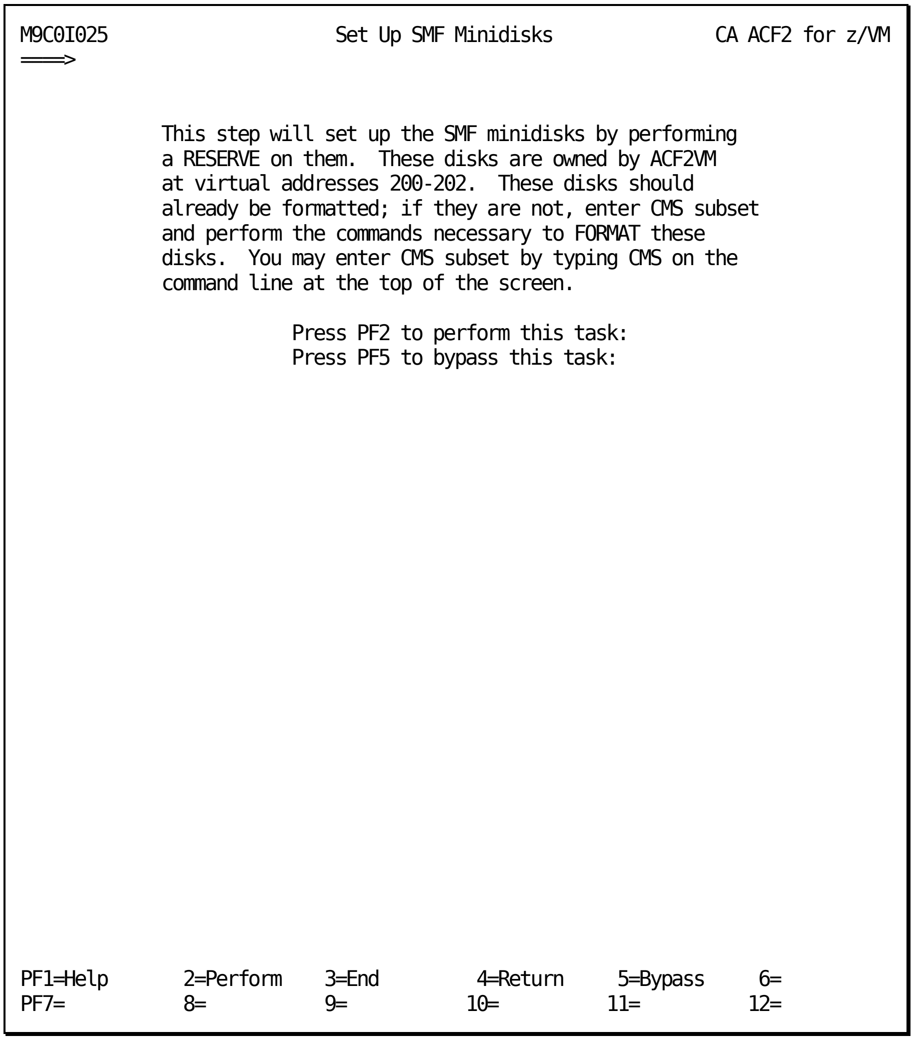
Task M9C0I025: Set Up SMF Minidisks

This task prepares minidisks for SMF recording. The @SMF macro in the ACFFDR specifies which SMF minidisks CA ACF2 for z/VM uses for SMF recording. Task M9C0I011 defined these minidisks and task M9C0I014 formatted them. If you are not using SMF minidisks, just press PF5 to bypass this task.

To complete this task:

1. Select task M9C0I025 from the Task Selection Menu.

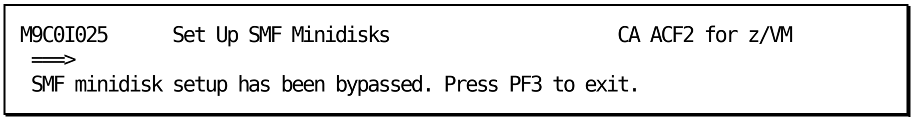
   The following panel appears:



```
M9C0I025                      Set Up SMF Minidisks                CA ACF2 for z/VM
===>


             This step will set up the SMF minidisks by performing
             a RESERVE on them.  These disks are owned by ACF2VM
             at virtual addresses 200-202.  These disks should
             already be formatted; if they are not, enter CMS subset
             and perform the commands necessary to FORMAT these
             disks.  You may enter CMS subset by typing CMS on the
             command line at the top of the screen.

                         Press PF2 to perform this task:
                         Press PF5 to bypass this task:


























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

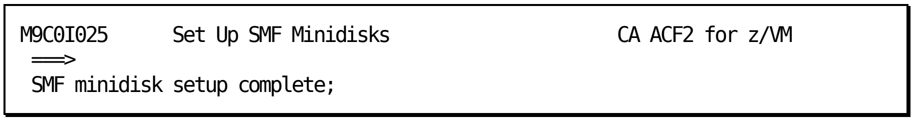
2. Press PF5 to bypass setting up the SMF minidisks or press PF2 to set up the SMF minidisks.

   If you press PF5, a message appears on the panel confirming that you are bypassing this step:



```
M9C0I025      Set Up SMF Minidisks                     CA ACF2 for z/VM
 ===>
 SMF minidisk setup has been bypassed. Press PF3 to exit.
```

   If you press PF2, a message appears on the above panel when processing is complete confirming that the SMF minidisks are set:



```
M9C0I025      Set Up SMF Minidisks                     CA ACF2 for z/VM
 ===>
 SMF minidisk setup complete;
```

3. Press PF3 to return to the Task Selection Menu.

Task M9C0I026: Set Up CMS Databases

This task sets up the CA ACF2 for z/VM CMS databases. The primary @DDSN macro in the ACFFDR specifies one or more CMS databases. You must set up the service machine CMS databases. See the *Administrator Guide* for information about the post backup service machine.

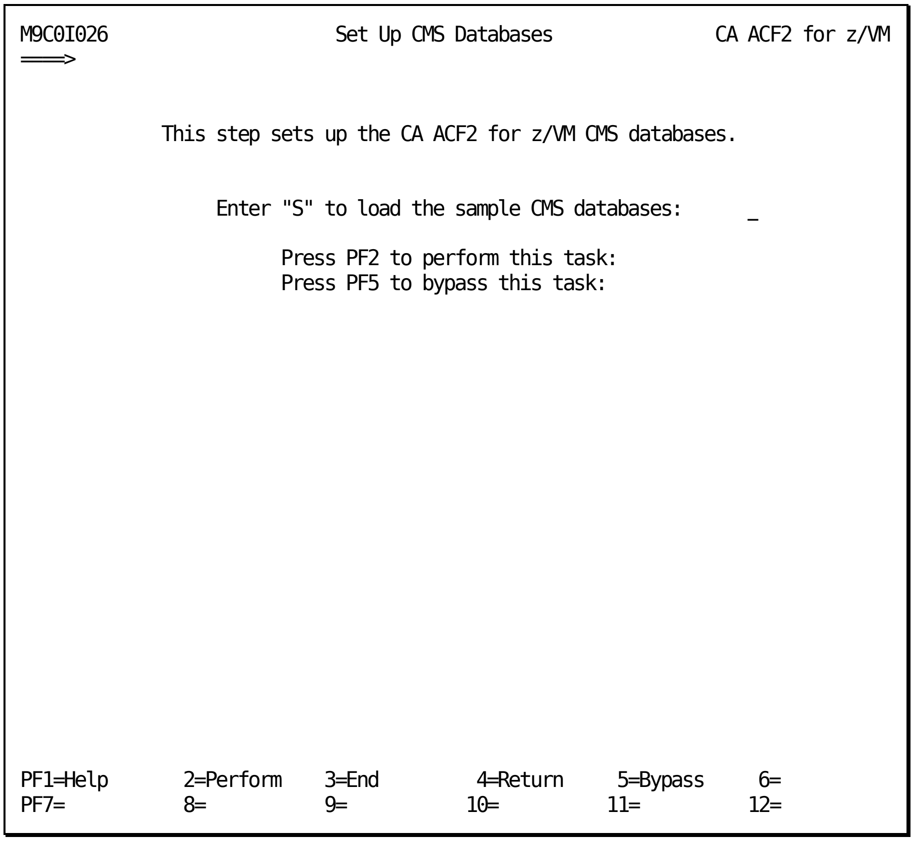
Each database requires an entire minidisk. Task M9C0I011 defined these minidisks and task M9C0I014 formatted them.

If you are a new CA ACF2 for z/VM site, load the supplied sample databases to set up your CMS databases. If you already have CMS databases, bypass this task.

To complete this task:

1. Select task M9C0I026 from the Task Selection Menu.

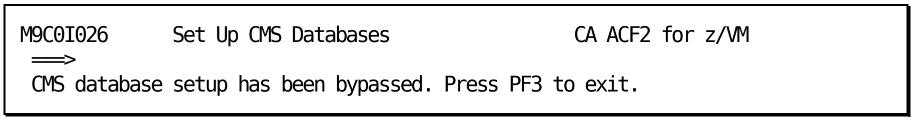
   The following panel appears:



```
M9C0I026                     Set Up CMS Databases                 CA ACF2 for z/VM
===>


             This step sets up the CA ACF2 for z/VM CMS databases.


                  Enter "S" to load the sample CMS databases:     _

                        Press PF2 to perform this task:
                        Press PF5 to bypass this task:














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

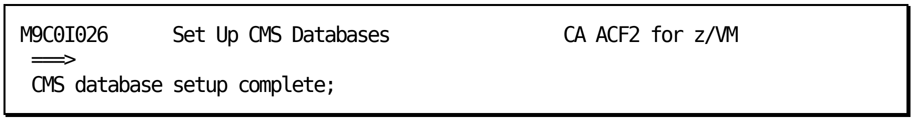
2. Press PF5 to bypass setting up the CMS databases or PF2 to set up the CA ACF2 for z/VM CMS databases.

   If you press PF5, a message appears on the panel confirming that you are bypassing this step:



```
M9C0I026      Set Up CMS Databases                 CA ACF2 for z/VM
 ===>
 CMS database setup has been bypassed. Press PF3 to exit.
```

   If you press PF2, a message appears on the above panel when processing is complete confirming that the CMS database is set:



```
M9C0I026      Set Up CMS Databases                CA ACF2 for z/VM
 ===>
 CMS database setup complete;
```

3. Press PF3 to return to the Task Selection Menu.

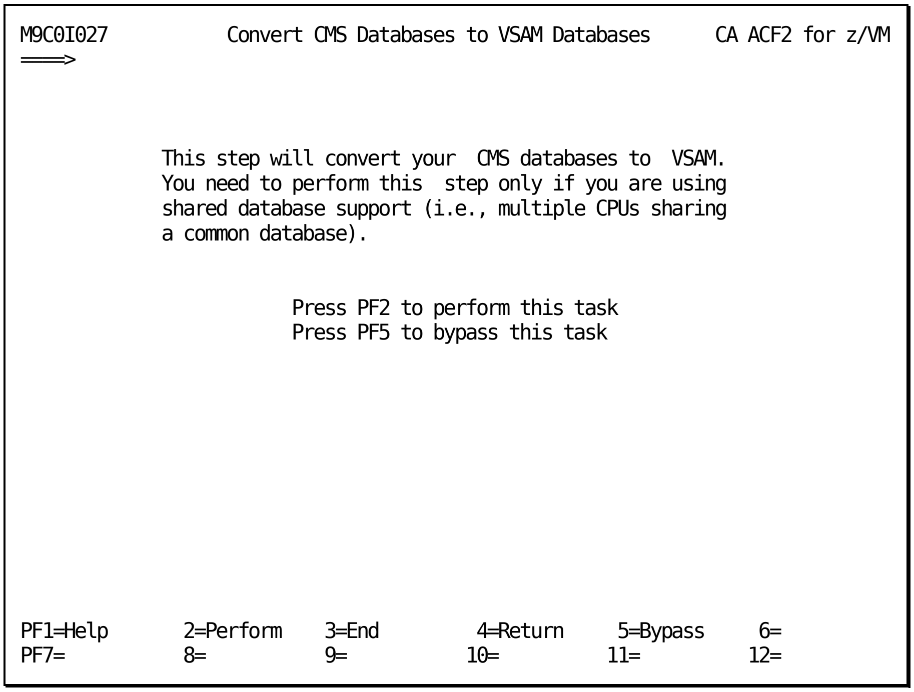
Task M9C0I027: Convert CMS Databases to VSAM Databases

This task converts CMS databases to VSAM databases. If your site is not sharing databases, press PF5 to bypass this task.

To complete this task:

1. Select task M9C0I027 from the Task Selection Menu.

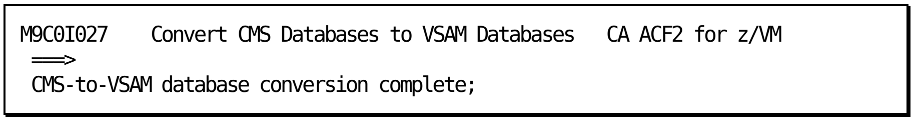
   The following panel appears:



```
M9C0I027         Convert CMS Databases to VSAM Databases      CA ACF2 for z/VM
===>



              This step will convert your  CMS databases to  VSAM.
              You need to perform this  step only if you are using
              shared database support (i.e., multiple CPUs sharing
              a common database).



                          Press PF2 to perform this task
                          Press PF5 to bypass this task














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

2. Press PF5 to bypass the CMS-to-VSAM database conversion, or
3. Press PF2 to run the conversion.

   If you press PF2 to run the conversion, when processing is complete, a message appears on the above panel confirming that the conversion is complete:



```
M9C0I027    Convert CMS Databases to VSAM Databases    CA ACF2 for z/VM
 ===>
 CMS-to-VSAM database conversion complete;
```

4. Press PF3 to return to the Task Selection Menu.

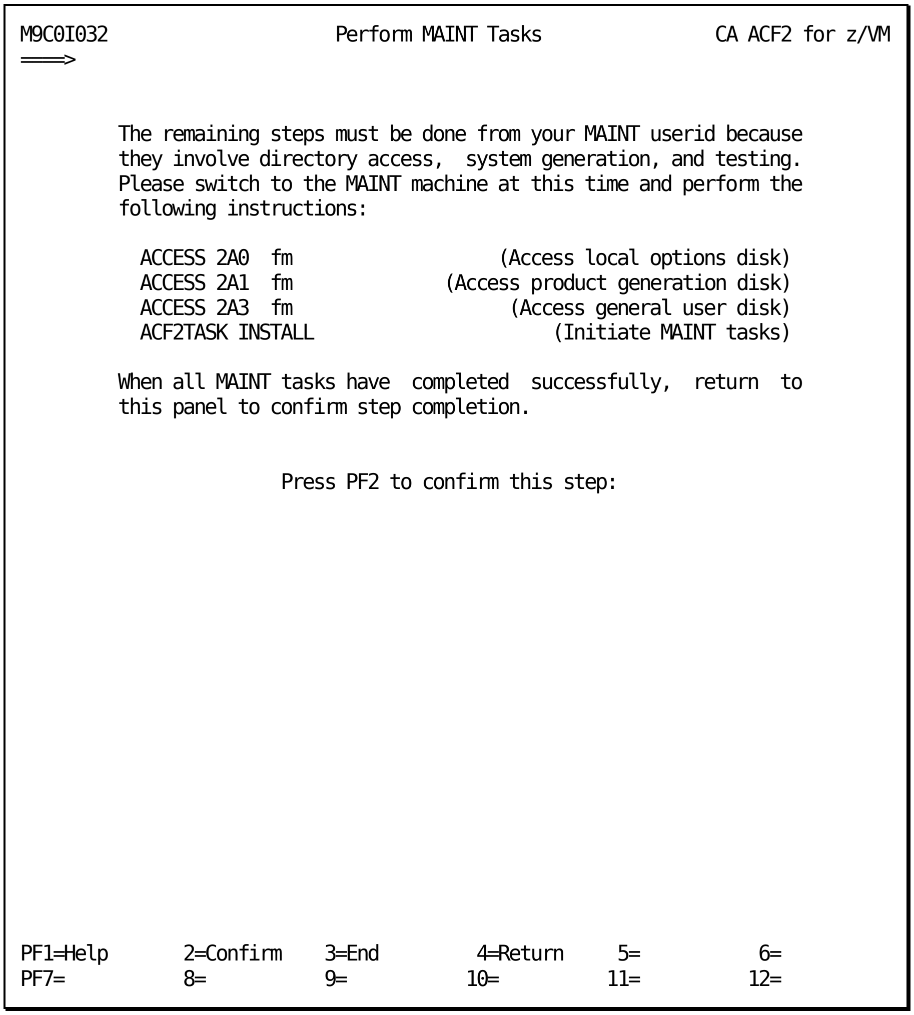
Task M9C0I032: Perform MAINT Tasks

You have completed all the steps that you can execute under CA-ACTIVATOR. You must perform the remaining installation tasks from the MAINT user ID. This task instructs you to switch to the MAINT service machine and initiate the ACF2TASK EXEC.

To complete this task:

1. Select task M9C0I032 from the Task Selection Menu.

   The following panel appears:



```
M9C0I032                      Perform MAINT Tasks                CA ACF2 for z/VM
====>


         The remaining steps must be done from your MAINT userid because
         they involve directory access,  system generation, and testing.
         Please switch to the MAINT machine at this time and perform the
         following instructions:

           ACCESS 2A0  fm                      (Access local options disk)
           ACCESS 2A1  fm                 (Access product generation disk)
           ACCESS 2A3  fm                       (Access general user disk)
           ACF2TASK INSTALL                         (Initiate MAINT tasks)

         When all MAINT tasks have  completed  successfully,  return  to
         this panel to confirm step completion.


                        Press PF2 to confirm this step:


















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

2. Log onto the VM maintenance ID. Follow the instructions on this panel.

   The ACF2TASK EXEC lets you perform the following steps. We provide detailed information for each step on the following pages.

Step	Description
1	Merges the VM directory into the Logonid database.
2	Modify options in VMXAOPTS and assemble HCPAC0 (or create/modify the ACF2VM CONFIG file options).
3	Sets up the STARTUP OPTIONS file.
4	Generates the CP nucleus, IPLs, and updates the VMO records.
5	Performs preliminaries for CMS installation.
6	Adds intercepts to CMS and CMS/DOS modules.
7	Generates the CMS nucleus with CA ACF2 for z/VM intercepts.
8	Generates module files for various CMS commands.
9	Finishes the installation.

The syntax of the ACF2TASK EXEC follows:

```
           [ INSTALL      ]
ACF2TASK   [ RESTART [nn] ]
           [ STEP   [nn]  ]
           [ ?            ]
```

INSTALL

Starts the ACF2TASK EXEC. You normally use this parameter once for a CA ACF2 for z/VM release.

RESTART

Restarts the exec. You are prompted for the step where you want to restart the installation. You can specify a step you already completed or the step after the one you last completed.

STEP

Installs one step. You can use this parameter for maintenance or to perform a step you previously completed.

?

Displays the ACF2TASK command syntax and provides help information.

nn

Indicates the step you want to restart with or the step you want to perform.

To execute the ACF2TASK utility, enter the following command:

```
ACF2TASK INSTALL
```

The 2A0 disk must be linked and accessed R/W. The 2A1 and 2A3 disks must be linked and accessed R/O.

You see the following information:

```
acf2task install

Do you want information on how to use this exec?
Reply "NO" or "YES". The default is "NO".
yes
Below is information on using the ACF2TASK EXEC and associated
CA ACF2 for z/VM execs.

There are many prompts that list all the valid replies. Defaults
are listed first. You can enter a null line to select defaults.
Most prompts have a default value.

All CA ACF2 for z/VM execs let you to reply "END" or "EXIT" to
immediately return to CMS with no further action.

All CA ACF2 for z/VM execs issue short prompts. For additional information
about the required reply, enter "?". The exec then issues more
detailed information and reissues the prompt.

You can reply "HELP" to most prompts to see a description of how
to use the exec.

You can reply "CMND" to most prompts. If "CMND" is not allowed,
you can enter "CMS". You can then enter one or more CMS or CP
commands. Enter a null line to reissue the prompt.

This exec issues a prompt at the beginning of each step that
states that the step is starting. Reply "GO" or enter a null
line to proceed with the specified step.

Reply "COMPLETE" to the STEP STARTING prompt to indicate you have
completed all of the functions that this step performs. This exec
can not check what you have done. It assumes that you did
everything correctly. Reply "COMPLETE" when the normal functions
of this exec do not meet your unique requirements, or if the step
requires you to perform tasks outside the scope of this exec and
you have completed those tasks.

If you reply "?" to the STEP STARTING prompt, this exec lists
information about the functions this step performs.


You can now repeat this information.

Do you want information on how to use this exec?
Reply "NO" or "YES". The default is "NO".
no
```

You can exit the procedure at the end of any step and restart it later. When you restart, ACF2TASK tells you which steps were successfully completed and asks if you want to start at the next step or a previous step. To restart the installation procedure, enter the following command:

```
ACF2TASK RESTART
```

When you have successfully completed all ACF2TASK steps, return to the CAIMAINT ID and press PF2 in M9C0I032 to confirm the step is complete.

A message appears on M9C0I032 confirming the action:

```
M9C0I032       Perform MAINT Tasks          CA ACF2 for z/VM
 ===>
 Action confirmed;
```

The remaining steps must be done from your MAINT userid because

3. Press PF3 to return to the Task Selection Menu.

Maintenance ID Installation Steps

The ACF2TASK EXEC performs installation steps for the MAINT user ID. Refer to Task M9C0I032 for information on the syntax of this exec.

The ACF2TASK EXEC prompts you to perform the following steps:

Step 1: Merge VM Directory into the Logonid Database

The ACFLIDGN utility is a conversion aid that generates a logonid record for each user in the VM directory. We recommend you use this procedure because users without logonid records cannot log onto a system with CA ACF2 for z/VM installed. Be aware, however, that incorrect input can lead to CA ACF2 for z/VM granting every logonid in your system SECURITY or some other CA ACF2 for z/VM privilege. Another possible outcome of incorrect input is that all logonids could have only the JOB and VM privileges. We strongly recommend you use the default values when you run the ACFLIDGN utility.

Sites that are using VSAM shared CA ACF2 for z/VM databases must use the ACF2VSAM utility. See the *Report and Utilities Guide* for information about this utility.

```
Step 1 is starting.
Merge the VM directory into the Logonid database.

Enter SKIP to bypass merging the CMS directory.
Enter GO to continue or ? for more information.
```

If you respond to this question with ?, you see the following information:

```
?
Step 1 merges the current VM directory into the Logonid database.

If you are using your current databases as they are, reply COMPLETE
to this prompt.

New users should reply GO.

This step performs the following functions:

A. Lets you generate logonid records from the VM directory. If you
   reply NO, go to step G. If you reply YES, you need read/write
   access to the logonid database minidisk.

B. Prompts for the directory filename, filetype, and filemode.
C. Asks if the directory is in DirMaint cluster/dirmpart format. If
   it is, prompts you for the filemode of the minidisk where the
   cluster/dirmpart files reside.

D. Prompts you for the logonid name you are using as a model.

E. Asks if you want to replace the user IDs already present in the
   Logonid database.

F. Asks if you want to clear the ZERO=YES field during the merge. If
   you reply YES, the sensitive data is not copied.

G. Lets you add or replace a specific logonid record.

Replies and their meanings are:

 GO  (or a null line) to continue with the installation.
 COMPLETE indicates you have completed all of the required functions
    this step performs. CA ACF2 for z/VM assumes that you have completed
    everything properly and continues to the next step.
 HELP displays information on using this exec and valid replies for
    all prompts.

Wait for a repeat of the prompt, then enter your reply.
Step 1 is starting.
Merge the VM directory into the Logonid database.

Enter SKIP to bypass merging the CMS directory.
Enter GO to continue or ? for more information.
```

If you respond to this question with GO, you see the following information:

```
go
ACFLIDGN EXEC preliminaries:

This exec lets you modify your logonid database using your VM directory
and the distributed CA ACF2 for z/VM model logonids. CA distributes a
sample database that includes model logonid records for various types of
users, including general users, auditors, and account managers. If you
are a new user, these sample files were previously copied to your service
machine database disk.

This exec executes in two phases. The first phase processes the VM
directory. It creates a logonid record from each user directory entry,
preserving the original user ID and password. It obtains additional
logonid record data from a model logonid you specify. NOLOG users
are bypassed.

The second phase prompts you for any logonids that should be added or
reconstructed using a different model. This provides a simple method to
establish security officers, auditors, and leaders. All users will be
required to change their passwords at the next logon.
Press ENTER to continue.

CA ACF2 for z/VM needs read/write access to the minidisk that contains your
logonid database. If this disk is already accessed read/write,
enter GO and CA ACF2 for z/VM continues. If this disk is not accessed
read/write, enter a null line, and CA ACF2 for z/VM puts you into CMS
mode. Enter the commands necessary to access this disk in
read/write mode.
Enter GO to continue or a null line to go into CMS.
```

The prompts that CA ACF2 for z/VM displays depend on your responses to previous questions.

```
Step 1 is complete.
```

Supplied CA ACF2 for z/VM Logonid Records

The starter Logonid database contains predefined logonid records. Use these supplied logonids only as models. Change the passwords for all the supplied logonids immediately to maintain security. You should also cancel or suspend them as soon as you establish appropriate local logonid records to prevent possible unauthorized uses of them. The chart below lists the predefined logonids and their passwords.

Logonid	Password	Privileges	Comments
ACCOUNT	ACCOUNT	ACCOUNT	Can create logonid records for other users.
ACFUSER	ACFUSER	SECURITY	Can store access rules, insert logonid records, and display all system parameters.

Logonid	Password	Privileges	Comments
ACF2VM	ACF2VM	AUDIT	A sample used by the CA-ACF2 service machine, can list all other logonid records and decompile access rules.
AUDIT	AUDIT	AUDIT	Can list all other logonid records and decompile all access rules.
AUTOLOG1	AUTOLOG1	USER	none
GENUSER	GENUSER	JOB	Model for general users.
MAINT	MAINT	SECURITY	Stores access rules and inserts additional logonid records.
OPERATOR	OPERATOR	USER	none
SECURITY	SECURITY	SECURITY	Writes access rules for all files.

Step 2: Modify VMXAOPTS

The VMXAOPTS macro in HCPAC0 and the ACFFDR specify a number of important options. We supply default values for these options. Review and modify these values as appropriate. For more information about the VMXAOPTS macro, see the *Systems Programmer Guide*.

You can also use the ACF2VM CONFIG file to supply these options. If you respond “CONFIG” when prompted, Step 2 allows you to create/modify the ACF2VM CONFIG. If you chose this option, you should remove any old HCPAC0 text file from a previous release that you have on your 2A0 minidisk.

Before you start this step, run VMFSETUP to access the disks required to generate your CP system. For example:

- For the test system, enter VMFSETUP ACFZ630 CPTEST
- For the production system, enter VMFSETUP ACFZ630 CP

```
Step 2 is starting.
Modify VMXAOPTS and assemble HCPAC0.
Enter GO to continue or ? for more information.
Enter “CONFIG” to edit or create the ACF2VM CONFIG file if you
      are using the ACF2VM CONFIG file support to supply
      VMXAOPTS options.
```

If you respond to this question with ?, you see the following information:

```
Step 2 modifies VMXAOPTS and assembles HCPAC0.

The VMXAOPTS macro specifies certain information about the
site environment that cannot be obtained from the CA ACF2 for z/VM
ACFFDR. The assembly of HCPAC0 includes this macro.

Enter "CONFIG" to edit or create the ACF2VM CONFIG file if you
are using the ACF2VM CONFIG file support to supply VMXAOPTS
options.

Reply "?" to the prompts within this step for more information.

This step performs the following functions: (for VMXAOPTS)

A. Prompts you to change VMXAOPTS.  If you reply "NO", go to
   step B.  If you reply "YES", CA ACF2 for z/VM issues the XEDIT
   command with the CTL option for HCPAC0 ASSEMBLE and displays
   the VMXAOPTS macro. You can then edit HCPAC0. When done,
   enter the xedit FILE subcommand and this exec regains
   control.

B. Prompts you to assemble HCPAC0. If you reply "YES", CA ACF2 for z/VM
   issues the ACF2ASM HCPAC0 CP G command. The minidisk with
   CP maclibs must be accessible.

This step performs the following functions: (for ACF2VM CONFIG)

A. If you enter "CONFIG", you are prompted to edit or create
   the ACF2VM CONFIG file.  If the file does not exist, you are
   given the option to create one starting with a sample config
   file.

Valid replies and their meanings are:

  "GO" (or a null line)  to continue with the installation.
  "CONFIG"  indicates that you want to edit or create the
       ACF2VM CONFIG file to specify VMXAOPTS options.
  "COMPLETE"  indicates you have completed all of the required
       functions this step performs. CA ACF2 for z/VM assumes that
       you have completed everything properly and continues to
       the next step.
  "HELP"  displays information on using this exec and
       valid replies for all prompts.

Wait for a repeat of the prompt, then enter your reply.
Press ENTER when you are finished reading:
```

If you respond to this question with GO, you see the following information:

```
go
Do you want to edit HCPAC0 (to modify VMXAOPTS)?.
Reply YES or NO. The default is YES.
```

We recommend that you add CAIMAINT to the FORCEID list.

After you edit your options, the following prompt appears:

```
Do you want to assemble HCPAC0 ?

Be sure that the minidisks with your CP maclibs are accessed.
If they are not, reply "CMS" and then enter the commands to
access them before replying as described below.

Reply "YES" or "NO".
Reply "EDIT" to re-edit HCPAC0.

The default is "YES".

Warning:  If you reply "NO", the HCPAC0 module may be incorrect.
```

The prompts that CA ACF2 for z/VM displays depend on your responses to previous questions.

If you reply “CONFIG” to the Step 2 prompt, you see the following:

```
Enter the fileid (filename, filetype, and filemode) of
your ACF2VM CONFIG file.  If the file does not exist yet,
you will be given the option to create the initial file
from the ACF2SAMP CONFIG file.  The ACF2VM CONFIG file
needs to reside on the same parm disk as the SYSTEM CONFIG
file.

You may substitute an equal sign ("=") to take the default
value for any part.  For example, "= = K" would mean use
the default filename and filetype with a filemode of "K".

The current fileid value is: ACF2VM CONFIG K
You may just press ENTER to use the current fileid
(Reply "CMS" to enter CMS commands)
```

If the file ID you enter does not exist, Step 2 allows you to create your ACF2VM CONFIG file from a sample:

```
Config file ACF2VM CONFIG K not found.
Do you want to create it from the ACF2SAMP CONFIG file?

Reply "YES" to create file: ACF2VM CONFIG K
Reply "NO" to enter a new fileid
Reply "CMS" to enter CMS commands
yes
```

You also require an imbed statement in your system config. If you do not have an imbed statement, the following warning appears:

```
Warning: The SYSTEM CONFIG does not appear to contain
an IMBED statement to include the ACF2VM CONFIG
file.  You need to have the following statement:
IMBED ACF2VM CONFIG
in the SYSTEM CONFIG file.  Do you want to xedit the
the SYSTEM CONFIG file?

Reply "YES" to xedit SYSTEM CONFIG K
Reply "SKIP" to skip this check and continue.
yes

Step 2 is complete.
```

Step 3: Set Up STARTUP OPTIONS File

The STARTUP OPTIONS file reloads the CA ACF2 for z/VM startup options. CA ACF2 for z/VM uses this information to do an automatic restart without prompting the operator. To use this facility, you must code the IPROMPT, RPROMPT, and PRDISK values in the VMXAOPTS macro in HCPAC0. See the *System Programmer's Guide* for details. The ACF2TASK EXEC skips if you specified YES for the RPROMPT operand of the VMXAOPTS macro of HCPAC0. If this is the case on your system, press Enter to skip this step.

Otherwise, CA ACF2 for z/VM displays the following prompts:

```
Step 3 is starting.
Set up the CA ACF2 for z/VM STARTUP OPTIONS file.
Enter "GO" to continue or "?" for more information.
Enter "SKIP" to skip this step.
Enter "CONFIG" to run this step if you are using the
       ACF2VM CONFIG file to supply VMXAOPTS options.
```

If you respond to this question with ?, you see the following information:

```
?
This step prepares a minidisk for the CA ACF2 for z/VM STARTUP OPTIONS file.

The PRDISK parameter of the VMXAOPTS macro in module HCPAC0
specifies the virtual address of the minidisk used for the CA ACF2 for z/VM
STARTUP OPTIONS file. This minidisk is required if you specify
the RPROMPT VMXAOPTS parameters as "NO". You must define this
minidisk in your VM directory before this step can complete.

If the ACF2VM CONFIG file is being used, the PRDISK option may
be specified with the following configuration statement:
  ACF2_STARTUP_OPTIONS nnnn
The RPROMPT option is specified with the following:
  ACF2_RESTART_PROMPT  NO

You must use the CMS FORMAT command to format the minidisk
in 4K blocks before it can be reserved for the CA ACF2 for z/VM STARTUP
OPTIONS file.

This step performs the following functions:

 A. Prompts you for the filemode of the minidisk used for the
    CA ACF2 for z/VM STARTUP OPTIONS file.

 B. Ensures the minidisk is formatted in 4K blocks.

 C. If the CA ACF2 for z/VM STARTUP OPTIONS minidisk is not reserved, it
    issues the CMS RESERVE command to reserve it.

 D. Initializes the STARTUP OPTIONS file.

Valid replies and their meanings are:

  "GO" (or a null line)  to continue with the installation.
  "CONFIG"  indicates that you want to continue with the
       installation similar to responding "GO", but that
       you are using the ACF2VM CONFIG file for VMXAOPTS
       options.
  "COMPLETE"  indicates you have completed all of the required
       functions this step performs. CA ACF2 for z/VM assumes that
       you have completed everything properly and continues to
       the next step.
  "HELP"  displays information on using this exec and
       valid replies for all prompts.

Wait for a repeat of the prompt, then enter your reply.
Press ENTER when you are finished reading:
```

If you respond to this question with GO, you see the following information.

```
go
Step 3 sets up the CA ACF2 for z/VM STARTUP OPTIONS file.
```

The prompts that CA ACF2 for z/VM displays depend on your responses to previous questions.

```
Step 3 is complete.
```

Step 4: Generate the CP Nucleus, IPL, and Update VMO Records

This step describes the procedures for generating a CP nucleus. If you are installing a new release of CA ACF2 for z/VM, be sure to reassemble your CA ACF2 for z/VM CP exits before generating your nucleus.

```
Step 4 is starting.
Generate the CP nucleus, perform the first IPL with CA ACF2 for z/VM, and update VMO
records.
Do this under a second-level CP.

Note: The CMS that you will use under this IPL can not have
any intercepts from a previous CA ACF2 for z/VM release.

For more detailed information, reply "?".  You should also
refer to Chapter 4 of the "CA ACF2 for z/VM Security for VM
Installation Guide" for more information.

Reply "?" , "EXIT" or "COMPLETE".

Reply "COMPLETE" after you have completed the required tasks.
```

If you respond to this question with ?, you see the following information:

```
Step 4 tells you to generate the CP nucleus,  do the
first IPL of the CP nucleus with CA ACF2 for z/VM installed, and update
VMO records.

This step tests the correct installation of CA ACF2 for z/VM CP code.

If a previous release of CA ACF2 for z/VM is protecting your CMS, you
cannot IPL CP at this time (you cannot run CMS). You must remove
all of the CA ACF2 for z/VM CMS intercepts from that CMS and generate a
a non-CA-ACF2 protected CMS, or skip this IPL (reply "COMPLETE")
and generate a new CMS protected by this CA ACF2 for z/VM release.

Complete the following tasks before continuing with this exec.

1. You can copy CP code needed to generate CP to a minidisk
   other than the CA ACF2 for z/VM source disk.
   This task is entirely optional and most sites do not need to
   do it. If you choose to do this, enter "ACF2COPY CPCODE fm",
   where fm is the filemode of the target disk.  ACF2COPY use the
   CPCODE ACFCOPY file to determine which files are copied.
   You can review and customize this file before running the exec.

2. Generate the CP nucleus using the CAXABLD EXEC.
   If you did not perform task 1, be sure that the CA ACF2 for z/VM source
   and local options disks are accessed during this generation.

3. Examine the CP nucleus map created by task 2.  Look at the
   bottom of the map for unresolved forward references. There
   should be no new undefined references. Be sure that all of
   the CA ACF2 for z/VM modules are included.

4. A. IPL your system with the NOAUTO CA ACF2 for z/VM option. Issue the
      "ACFSERVE ENABLE NOAUTO UPDATES" command to start the
      service machine to allow updates. Note that only the user that
      issues this command can actually use the ACF command to
      update the CA ACF2 for z/VM databases in NOAUTO mode.

   B. If this is the first time you are installing CA ACF2 for z/VM
      or migrating from a release earlier than 3.1, review all of
      the VMO record options to determine the settings that you
      should use.

   C. Perform additional checks for the following field values.
      The default settings have a potentially negative impact
      on your system:
         OPTS ACCTVLD(FULL)
         OPTS MODE(ABORT)
         OPTS IUCVVLD
         OPTS VMCFVLD
         OPTS DSPVLD
         CMDLIM MODE(ABORT)
         DIAGLIM MODE(ABORT)

   D. Make the necessary changes to the VMO records. Refer to
      the "CA ACF2 for z/VM Security for VM Installation Guide"
      for details.

   E. Ensure that any access rules, command models, and command
      limiting rules required for testing based on the selected
      VMO setting are in place before continuing.
```

```
  F. Refer to the section titled "Final Installation Tasks" in
     the "CA ACF2 for z/VM Security for VM Installation Guide"

  G. After you finish the preceeding steps, issue the "ACFSERVE
     DISABLE NOAUTO UPDATES" command to shutdown the service
     machine.

  H. IPL the system and test the CP nucleus.

5. If you see no problem, continue with the installation.

Refer to the "CA ACF2 for z/VM Security for VM Installation Guide"
for more details.

When you finish the above tasks, issue the "ACF2TASK RESTART" command
and restart with this step (step 4). Then reply "COMPLETE"
to this step's prompt.

Press ENTER when you are finished reading:
```

This is a manual step you must perform outside of the exec. When the following manual procedures instruct you to invoke the CAXABLD EXEC, the intention for most sites is that the CAXABLD command replaces the VMFBLD command in your existing system generation procedures.

1. Ensure that you modified the system PPF file.

   - The file is named ACFZnnn $PPF, where nnn is the version and release of VM. The CA ACF2 for z/VM local options and source disks are your LOCALACF disks.

2. The National Language feature lets you alter CA ACF2 for z/VM messages.

   - You can define selected messages to highlight at the destination terminal
   - You can set an audible alarm bit for selected messages
   - You can send messages to selected users (such as security administrators) to control the routing of messages. You may want to do this if you are running PROP (Programmable Operator).

   You can update the copy book containing the ACFCPMST macros (referenced in HCPAF0), rebuild the ACF2USER MACLIB, reassemble HCPAF0, and generate a CP nucleus to alter CA ACF2 for z/VM messages at any time. As a time saving measure, modify CA ACF2 for z/VM messages now. For details, see the “Installation Options”.

3. Enter the following command to use VMFSETUP to access the disks required to generate your CP nucleus:

   For the test system, enter

   ```
   VMFSETUP ppfname CPTEST
   ```

   For the production system, enter

   ```
   VMFSETUP ppfname CP
   ```

   ppfname

   Is your PPF file, usually ACFZnnn, where nnn is the version and release of z/VM.

4. Use the CAXABLD EXEC to invoke your normal system generation command. For example:

   For the test system, if you use:

   ```
   VMFBLD PPF ppfname CPTEST CPLOAD (ALL
   ```

   You should now enter:

   ```
   CAXABLD ACFCP VMFBLD PPF ACFZnnn CPTEST CPLOAD (ALL
   ```

   where nnn is the version and release of z/VM

   For the production system, if you use:

   ```
   VMFBLD PPF ppfname CP CPLOAD (ALL
   ```

   You should now enter:

   ```
   CAXABLD ACFCP VMFBLD PPF ACFZnnn CP CPLOAD (ALL
   ```

   where nnn is the version and release of z/VM

   Note: Your installation may have a CP Nucleus building EXEC that contains the VMFBLD command that actually builds the CP nucleus. In this case, your installation's EXEC should be changed to use the CAXABLD command instead of the VMFBLD command.

5. Review the CP nucleus load map for unresolved references.

   If you did not select the upper case language support, you will see two undefined references: HCPAL0 and HCPAL0MT. These are normal. When you comment out or delete the LANG= parameter of VMXAOPTS, you should also delete the UCENG definition in CAXALOAD. This eliminates the undefined reference to HCPAL0.

   Review the CAXALOAD Summary Log for CP that contains information reflecting the new references the CAXALOAD control file changed. This log is in the CAXALOAD SUMMARY A1 file. Rename this file so future nucleus generations do not erase the old file to create another with the same name.

   Follow whatever normal procedures your installation uses to move the newly generated CP nucleus MODULE file to its normal IPL location.

   If you are refreshing CA ACF2 for z/VM with a new genlevel, skip to **Step 14.**

6. IPL your system with the NOAUTO CA-ACF2 option. Issue the following command to start the service machine to allow updates:

   ```
   ACFSERVE ENABLE NOAUTO UPDATES
   ```

   Note: Only the user that issues this command can actually use the ACF command to update the CA ACF2 for z/VM databases in NOAUTO mode.

7. If this is the first time you are installing CA-ACF2 or migrating from a release earlier than 3.2, review all of the VMO record options to determine the setting that you should use.

8. If you are installing on a Portable Operating Systems Interface for Computing Environments (POSIX) system, be sure you:

   - Define user profile records
   - Define group profile records
   - Define a primary logon group for each user
   - Define resource supplemental group records
   - Set the appropriate OPTS, RESCLASS, and RESTYPE VMO records.

9. Perform additional checks for the following field values. The default settings have a potentially negative impact on your system:

   ```
   OPTS ACCTVLD(FULL)
   OPTS DSPVLD
   OPTS IUCVVLD
   OPTS MODE(ABORT)
   OPTS VMCFVLD
   CMDLIM MODE(ABORT)
   DIAGLIM MODE(ABORT)
   ```

10. Make the necessary changes to the VMO records. If you do not create VMO records or make changes, CA ACF2 for z/VM creates VMO records using the default values.

11. Ensure that any access rules, command models, and command limiting rules required for testing based on the selected VMO setting are in place before continuing.

12. You can make additional changes using the ACF command.

13. After you finish the preceding steps, issue the following command to shutdown the service machine:

    ```
    ACFSERVE DISABLE NOAUTO UPDATES
    ```

14. Test the new CP nucleus at this point to verify that CA ACF2 for z/VM calls CA ACF2 for z/VM front-end modules and that the service machine works properly. Before IPL, we recommend you specify the VM system operator as the secondary console for AUTOLOG1 and the CA-ACF2 service machine for the duration of the CA ACF2 for z/VM installation. This lets the system programmer observe any unusual conditions and, if necessary, reply to error messages. For example, include the following statement in the VM directory for the CA-ACF2 service machine.

    ```
    CONSOLE 009 3215 T OPERATOR
    ```

15. IPL the system. Enter a null in response to the following message:

    ```
    ACFpgm000R Enter CA-ACF2 startup options or press enter
    ```

    The following startup options are also available:

    DDSN(group-name)

    Specifies the group names of the CA ACF2 for z/VM databases used. The ACFFDR @DDSN macro specifies these group names. DDSN(group-name) starts CA ACF2 for z/VM up using an alternate set of CA ACF2 for z/VM databases during recovery, if necessary.

    NOAUTO

    Starts the VM system in FORCE mode without activating CA ACF2 for z/VM. When running in FORCE mode, only users defined in both FORCEID and NOAUTO operands of the VMXAOPTS macro can log on. CA ACF2 for z/VM does not perform validations. Specifying NOAUTO implies NODBSYNC.

    NOBACKUP

    Service machine autologs normally. However, CA ACF2 for z/VM deactivates the automatic database backup facility. You must also set TIME=(0000) in the VMO BACKUP record.

    NODBSYNC

    Prevents the automatic start of the Database Synchronization Component (DSC). You can use the ACFSERVE ENABLE SYNC command later to start the DSC. Specifying NOAUTO implies NODBSYNC.

    NULL(EOB)

    CA ACF2 for z/VM autologs the CA-ACF2 service machine normally when you press Enter.

    SHUTDOWN

    Shuts down the VM system.

    SYSID(systemid)

    Specifies the system ID of the VMO records that are loaded and processed at IPL time. The SYSID you enter overrides the @SYSID macro setting specified in the ACFFDR.

The system displays the following message at the VM operator console when the CA-ACF2 service machine begins its initialization process:

```
[hh:mm:ss] AUTO LOGON ***    ACF2VM  USERS = 2   BY SYSTEM
```

The CA-ACF2 service machine autologs whenever the first user logs onto the system.

```
Step 4 is complete.
```

Step 5: Preliminary CMS Installation Steps

This step describes the procedures for performing preliminary CMS installation steps. CA ACF2 for z/VM security for CMS has several optional parts. You select these parts in CA-ACTIVATOR, but ACF2TASK implements them.

Basic CMS security consists of:

- Source mods to CMS text files (assembled in ACF2TASK Step 6). These files are:

  DMSBOP

  DMSCIT

  DMSEXD

  DMSEXI

  DMSNXD

  DMSSOP

  DMSSVT

- CMS nucleus generation with CAXABLD (ACF2TASK Step 7). This adds dynamic hooks and picks up the CA ACF2 for z/VM modified versions of:

  DMSCIT

  DMSEXI

  DMSSOP

  DMSSVT

- CMS MODULE files generation to include CA ACF2 for z/VM code (ACF2TASK Step 8). These modules are:

  EXECDROP

  FORMAT

  GLOBAL

  NUCXDROP

  RESERVE

  This process adds a CA ACF2 for z/VM frontend to the module for FORMAT, GLOBAL, and RESERVE, or picks up the CA ACF2 for z/VM modified version of DMSEXD for EXECDROP and DMSNXD for NUCXDROP.

VSAM security (also included with CMS/DOS security, described below). This protects VSAM files, whether they belong to VM, or VSE. They consist of:

- Basic CMS security as defined in point 1 above.
- CMSDOS segments generation to include our modified code. This picks up the modified version of DMSBOP, which is called for VSAM open processing.

CMS/DOS security (including VSAM security mentioned above in point 2). This consists of:

- Basic CMS security as defined in point 1 above.
- Source mods to CMS text files (assembled in ACF2TASK Step 6). Text files include:

  DMSDLK

  DMSFCH

  DMSOPL

  DMSPRV

  DMSRRV

  DMSSRV

- CMS/DOS segments generation to include CA ACF2 for z/VM modified code. This process picks up the CA ACF2 for z/VM modified versions of the text files:

  DMSBOP

  DMSFCH

  DMSOPL

- CMS/DOS modules generation to include CA ACF2 for z/VM code (ACF2TASK Step 8). These modules are:

  DOSLKED

  PSERV

  RSERV

  SSERV

  This process picks up the CA ACF2 for z/VM modified versions of the text files:

  DMSDLK

  DMSPRV

  DMSRRV

  DMSSRV

```
Step 5 is starting.
Perform some CA ACF2 for z/VM preliminaries for CMS needed for later steps.
Enter GO to continue or ? for more information.
```

If you respond to this question with ?, you see the following information:

```
Step 5 initiates actions necessary for later steps when
installing CA ACF2 for z/VM CMS file protection.

This step performs the following functions:

A. Copies the CAXALOAD file for your CMS release to the
  ACFCMS CAXALOAD file on your local options disk.

B. Copies the DMSAC0 TEXT file for your CMS release to the
  DMSAC0 TEXT file on your local options disk.

C. Updates the ACFCMS CAXALOAD file, depending on options
  that you selected in step 1 (Update CA ACF2 for z/VM OPTIONS
  file).

D. Puts you in xedit to review and customize the ACFCMS
   CAXALOAD file.

Refer to the CA ACF2 for z/VM Installation Guide for further details.

Valid replies and their meanings are:

 "GO" (or a null line) to continue with the installation.
 "COMPLETE" indicates you have completed all of the required
    functions this step performs. CA ACF2 for z/VM assumes that
    you have completed everything properly and continues to
    the next step.
 "HELP" displays information on using this exec and
    valid replies for all prompts.

Wait for a repeat of the prompt, then enter your reply.

Step 5 is starting.
Perform some CA ACF2 for z/VM preliminaries for CMS needed for later steps.
Enter "GO" to continue or "?" for more information.
```

If you respond to this question with GO, you see the following information:

```
go
Do you want to create your initial ACFCMS CAXALOAD and CMS
release-dependent CA ACF2 for z/VM modules (DMSAC0)?
Reply YES or NO.
The default is YES.
```

The prompts that CA ACF2 for z/VM displays depend on your responses to previous questions.

```
Step 5 is complete.
```

Step 6: Add CA ACF2 for z/VM Intercepts to CMS Modules

This step describes the procedures for adding CA ACF2 for z/VM intercepts to the CMS modules. CMS source files are required to do the assemblies.

```
Step 6 is starting.
Installing CA ACF2 for z/VM intercepts to CMS modules.
Enter GO to continue or ? for more information.
```

If you respond to this question with ?, you see the following information:

```
Step 6 adds CA ACF2 for z/VM intercepts to CMS and CMS/DOS modules.

This step performs the following functions:

A. Uses the CMSRELnn ASMLIST file to determine which CMS
   modules can be assembled with CA ACF2 for z/VM intercepts.
   Where "nn" is  21 for Release 21 CMS
                  22 for Release 22 CMS
                  23 for Release 23 CMS
                  24 for Release 24 CMS
                  25 for Release 25 CMS
                  26 for Release 26 CMS
                  27 for Release 26 CMS

B. If you specified DOS = YES in step 1, it uses the
   CMSDOSnn ASMLIST file to determine which CMS/DOS modules
   can be assembled with CA ACF2 for z/VM intercepts.
   Where "nn" is  21 for Release 21 CMS
                  22 for Release 22 CMS
                  23 for Release 23 CMS
                  24 for Release 24 CMS
                  25 for Release 25 CMS
                  26 for Release 26 CMS
                  27 for Release 26 CMS

C. Asks you which of the modules in task A and task B above are
   to be assembled.

D. Performs the assemblies based on your reply to task C.
   The minidisks with CMS maclibs must be accessible.

E. If it detects an error for any assembly, it prompts you for an
   action to take.


Valid replies and their meanings are:

  "GO" (or a null line)  to continue with the installation.
  "COMPLETE"  indicates you have completed all of the required
       functions this step performs. CA ACF2 for z/VM assumes that
       you have completed everything properly and continues to
       the next step.
  "HELP"  displays information on using this exec and
       valid replies for all prompts.

Wait for a repeat of the prompt, then enter your reply.
Press ENTER when you are finished reading:
```

If you respond with GO, you see the following information:

```
This step uses the ACF2ASM EXEC to assemble CMS and CMS/DOS
modules with CA ACF2 for z/VM intercepts.

Before proceeding with these assemblies, you must access the
disks required to generate CMS, if you have not done so already.

Reply "CMS" to enter CMS commands.
Reply "EXIT" to leave this EXEC.

Press "ENTER" to continue.
```

The prompts that CA ACF2 for z/VM displays depend on your responses to previous questions.

```
Step 6 is complete.
```

Note: Remember to run VMFSETUP for CMS to access the CMS minidisks.

Step 7: Generate and Test the CMS Nucleus

This step describes the procedures for generating and testing the CMS nucleus.

```
Step 7 is starting.
Generate the CMS nucleus with CA ACF2 for z/VM intercepts.
IPL CMS and test the new CMS nucleus.

Do this step under the same second-level CP you previously
generated with CA ACF2 for z/VM protection.

Reply ? (for more information), EXIT, or COMPLETE to this step prompt.
Reply COMPLETE only if you have already generated the new CMS nucleus
and tested CMS.
```

This is a manual step you must perform outside of the exec. When the following manual procedures instruct you to invoke the CAXABLD EXEC, the intention for most sites is that the CAXABLD command replaces the VMFBLD command in your existing system generation procedures.

The prompts that CA ACF2 for z/VM displays depend on your responses to previous questions.

1. Ensure that you properly updated your system's CNTRL file for CMS. See “Step 5: Preliminary CMS Installation Steps” for information.
2. To access the disks required to generate your CMS nucleus, enter the following command:

   For the test system, enter

   ```
   VMFSETUP ppfname CMSTEST
   ```

   For the production system, enter

   ```
   VMFSETUP ppfname CMS
   ```

   ppfname

   The name of your PPF file. The filename for the PPF file is ACFZnnn.

   where nnn is the version and release of VM
3. You must perform this step on a CP system with CA ACF2 for z/VM for VM r12 installed. Use the CAXABLD EXEC to invoke your normal system generation command. For example, if you currently generate your system with the VMFBLD $ppfname CMS (PUNCH command, enter the following command:

   For the test system, enter:

   ```
   CAXABLD ACFCMS VMFBLD PPF ACFZnnn CMSTEST CMSLOAD (ALL
   ```

   For the production system, enter:

   ```
   CAXABLD ACFCMS VMFBLD PPF ACFZnnn CMS CMSLOAD (ALL
   ```

   where nnn is the version and release of VM
4. Review the CMS nucleus load map. There should be no undefined external references. The following intercepted modules are in the CMS nucleus map:

   DMSCIT

   DMSEXI

   DMSSOP

   DMSSVT

   Review the CAXALOAD Summary Log for CMS that contains information reflecting the new references the CAXALOAD control file changed. This log is in the CAXALOAD SUMMARY A1 file. Rename this file so future nucleus generations do not erase the old file to create another with the same name.

5. To IPL the new CMS system, enter the following command:
6. Ensure that the new CMS system works correctly. For example, XEDIT a file or issue an ACCESS command. If these commands work as expected, continue installing CA ACF2 for z/VM. If they do not work, verify Steps 4, 5, and 6 completed successfully.
7. If you selected the optional CMS/DOS protection, issue the following command based on your VM release:

   For the test system, enter:

   ```
   VMFBLD PPF ACFZnnn CMSTEST DMSBLDOS (ALL
   ```

   For the production system, enter:

   ```
   VMFBLD PPF ACFZnnn CMS DMSBLDOS (ALL
   ```

   Where nnn is the version and release of VM
8. If you are installing CMS/DOS protection, see the IBM *Service Guide* for information on how to generate your CMS/DOS and related segments. If you are installing CA ACF2 for z/VM for VM VSAM AMSERV support, see the "Installation Options" chapter before resaving your VSAM segments.
9. To install CA ACF2 for z/VM support for VSAM AMSERV, See the "Installation Options" chapter.

When you respond with COMPLETE, you see the following information:

```
Step 7 is complete.
```

Step 8: Generate Module Files For CMS Commands

This step describes the procedures for generating the modules files for CMS commands.

```
Step 8 is starting.
Generate module files for various CMS commands.
Enter GO to continue or ? for more information.
```

1. Run VMFSETUP to access the disks you need to generate CMS.

   For the test system, enter:

   ```
   VMFSETUP ACFZnnn CMSTEST
   ```

   For the production system, enter:

   ```
   VMFSETUP ACFZnnn CMS
   ```

   where nnn is the version and release of VM

2. Enter one of the following commands to apply CA ACF2 for z/VM security to the CMS modules:

   For the test system, enter:

   ```
   VMFBLD PPF ACFZnnn CMSTEST ACFMLDrr (ALL
   ```

   For the production system, enter:

   ```
   VMFBLD PPF ACFZnnn CMS ACFMLDrr (ALL
   ```

   where nnn is the version and release of VM

   where rr is the CMS release number

3. If you selected the optional CMS/DOS protection, enter one of the following commands to apply CA ACF2 for z/VM security to the CMS/DOS modules:

   For the test system, enter:

   ```
   VMFBLD PPF ACFZnnn CMSTEST ACFDOSrr (ALL
   ```

   For the production system, enter:

   ```
   VMFBLD PPF ACFZnnn CMS ACFDOSrr (ALL
   ```

   where nnn is the version and release of VM

   where rr is the CMS release number

Step 9: The Final IPL with CA ACF2 for z/VM

This step describes the procedures for performing the final IPL with CA ACF2 for z/VM for VM installed.

```
Step 9 is starting.
Finish the installation.

If you are installing CA ACF2 for z/VM CMS file protection, test the
generated modules and save the new CMS.

Migrate the CA ACF2 for z/VM generated CP and CMS to first level and
perform the final installation tasks.

Reply "?" (for more information), "EXIT" or "COMPLETE"
to this prompt.

Reply "COMPLETE" after you have completed these tasks.
```

If you respond to this question with ?, you see the following information:

```
Step 9 tells you to test the generated modules with CA ACF2 for z/VM
intercepts and save CMS. It also tells you to move the CA ACF2 for z/VM
generated CP and CMS to first level, IPL and perform the final
installation tasks.

You must complete the following tasks before the installation of
CA ACF2 for z/VM is complete.

If you are installing CMS file protection, test the generated
modules and save CMS.
A. Test the FORMAT, GLOBAL, NUCXDROP, and RESERVE commands.

B. If you see no problems, save the newly generated CMS system.
   Do the following to save CMS:
      1. Issue the DEFINE STORAGE command if necessary to define
         storage large enough to IPL the 190 disk.
      2. Issue the DEFSYS command (or SAMPNSS or equivalent) to
         define the NSS file for CMS.
      3. Issue the command: "IPL 190 PARM SAVESYS CMS"

C. IPL the newly-saved CMS (eg."IPL CMS") and test.

D. If you see no problem, then migrate to first level.

To migrate to the first level:

A. Move your CA ACF2 for z/VM generated system to first level.

B. IPL the VM operating system, using the CA ACF2 for z/VM protected CP
   and CMS systems generated above. CA ACF2 for z/VM is now installed
   and active.

C. The security administrator or the VM systems programmer should
   logon, using the default CA ACF2 for z/VM logonid of MAINT.

D. Create any CMS userIDs missing from the Logonid database.

E. Compile and store access rules to avoid excessive loggings.

F. Install optional CA ACF2 for z/VM support as appropriate to your
   site. This may include user exits, and other product support.

Refer to the "CA ACF2 for z/VM Security for VM Installation Guide"
for more details.

This is the last initial installation step. You can reply
"COMPLETE" now if you intend to perform the above steps. CA ACF2 for z/VM will assume
this step was done and do final cleanups.

You may also reply "END" or "EXIT". When you finish the above
steps, issue the "ACF2TASK RESTART" command and restart with
this step (step 9).  Then reply "COMPLETE" to this step
prompt.

Valid replies and their meanings are:

  "GO" (or a null line)  to continue with the installation.
  "COMPLETE"  indicates you have completed all of the required
       functions this step performs. CA ACF2 for z/VM assumes that
       you have completed everything properly and continues to
       the next step.
  "HELP"  displays information on using this exec and
       valid replies for all prompts.
   Wait for a repeat of the prompt, then enter your reply.

If you respond with COMPLETE, you see the following information:

Step 9 is complete.
The CA ACF2 for z/VM MAINT userid tasks are complete.

Return now to the CAIMAINT userid and confirm completion
of task M9C0I032 by pressing PF2 in that panel.
```

Return to the CAIMAINT ID and press PF2 to confirm that all ACF2TASK steps are complete.

Task M9C0I033: Execute ACFCVSFS Utility

This task prompts you to convert existing SFS grants to access rules. After running this utility, you must compile resulting rule sets to add

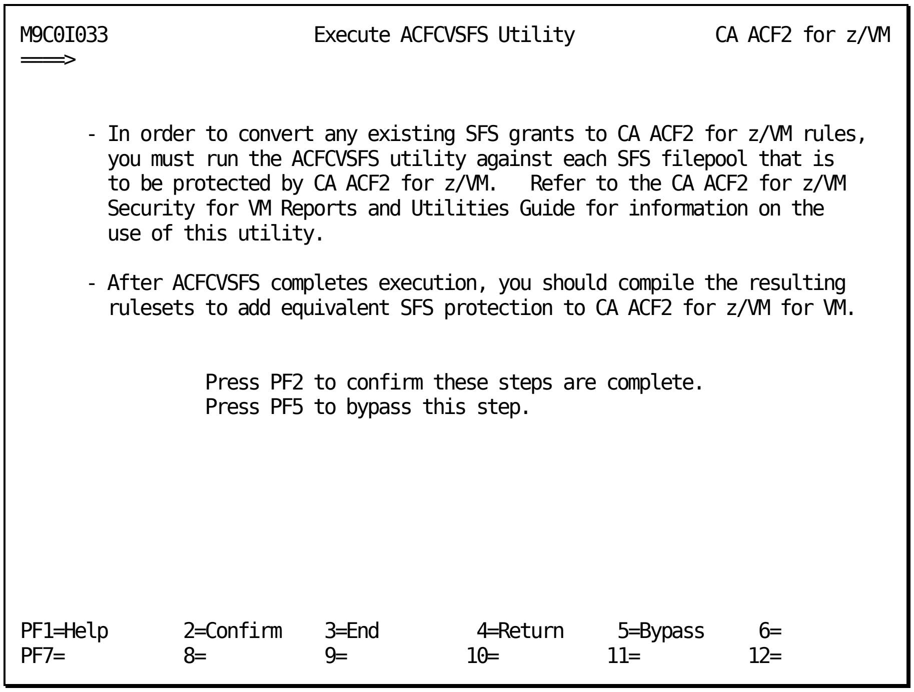
SFS protection to CA ACF2 for z/VM for VM.

See the *Reports and Utilities Guide* for information about restrictions to running this utility.

To complete this task:

1. Select task M9C0I033 from the Task Selection Menu.

   The following panel appears:



```
M9C0I033                   Execute ACFCVSFS Utility            CA ACF2 for z/VM
===>


      - In order to convert any existing SFS grants to CA ACF2 for z/VM rules,
        you must run the ACFCVSFS utility against each SFS filepool that is
        to be protected by CA ACF2 for z/VM.   Refer to the CA ACF2 for z/VM
        Security for VM Reports and Utilities Guide for information on the
        use of this utility.

      - After ACFCVSFS completes execution, you should compile the resulting
        rulesets to add equivalent SFS protection to CA ACF2 for z/VM for VM.


                 Press PF2 to confirm these steps are complete.
                 Press PF5 to bypass this step.










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

2. After you have run the ACFCVSFS utility for each SFS filepool you are protecting and compiled the rule sets, return to this panel 2. Press PF2 to confirm that the SFS grants were converted.

3. Press PF3 to return to the Task Selection Menu.

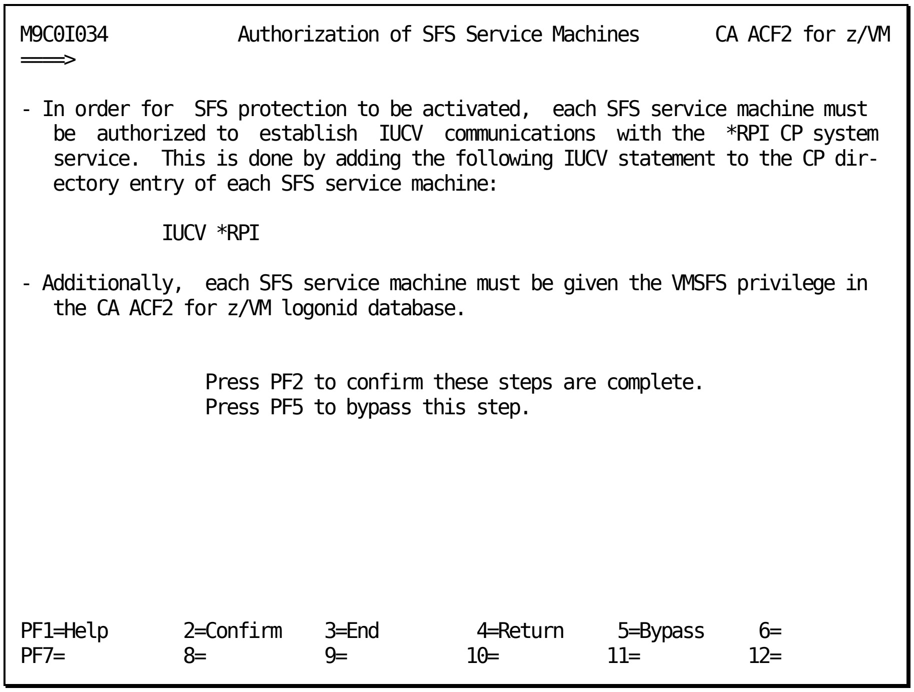
Task M9C0I034: Authorization of SFS Service Machines

This task prompts you to establish authorization for each SFS service machine to establish IUCV communications.

To complete this task:

1. Select task M9C0I034 from the Task Selection Menu.

   The following panel appears:



```
M9C0I034          Authorization of SFS Service Machines       CA ACF2 for z/VM
===>

- In order for  SFS protection to be activated,  each SFS service machine must
   be  authorized to  establish  IUCV  communications  with the  *RPI CP system
   service.  This is done by adding the following IUCV statement to the CP dir-
   ectory entry of each SFS service machine:

             IUCV *RPI

- Additionally,  each SFS service machine must be given the VMSFS privilege in
   the CA ACF2 for z/VM logonid database.


                 Press PF2 to confirm these steps are complete.
                 Press PF5 to bypass this step.








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

2. After you give IUCV authorization to each SFS service machine and give each SFS service machine the VMSFS logonid privilege, return to this panel.

3. Press PF2 to confirm that this task is completed.

4. Press PF3 to return to the Task Selection Menu.

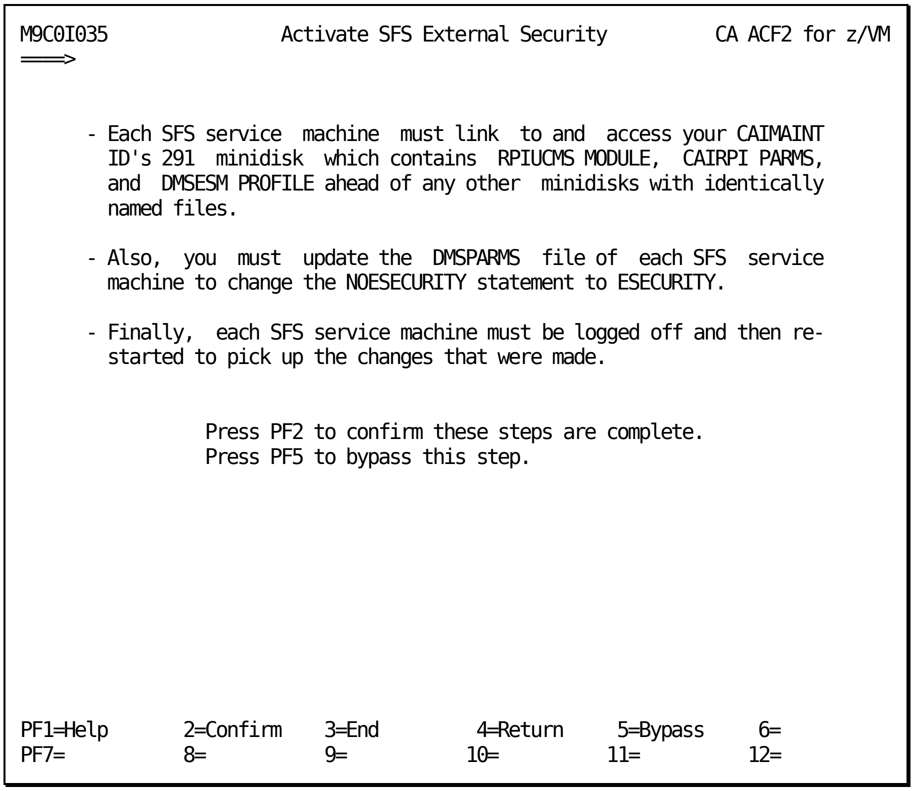
Task M9C0I035: Activate SFS External Security

This task prompts you to activate SFS external security.

To complete this task:

1. Select task M9C0I035 from the Task Selection Menu.

   The following panel appears:



```
M9C0I035                Activate SFS External Security          CA ACF2 for z/VM
===>


      - Each SFS service  machine  must link  to and  access your CAIMAINT
        ID's 291  minidisk  which contains  RPIUCMS MODULE,  CAIRPI PARMS,
        and  DMSESM PROFILE ahead of any other  minidisks with identically
        named files.

      - Also,  you  must  update the  DMSPARMS  file of  each SFS  service
        machine to change the NOESECURITY statement to ESECURITY.

      - Finally,  each SFS service machine must be logged off and then re-
        started to pick up the changes that were made.


                 Press PF2 to confirm these steps are complete.
                 Press PF5 to bypass this step.









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

2. After you have defined the links, updated the DMSPARMS file, and restarted each SFS service machine, return to this panel.

3. Press PF2 to complete the steps outlined in this task.

4. Press PF3 to return to the Task Selection Menu.

   After you complete the installation, see Final Installation Tasks (see page 152) for information about creating logonids and compiling rules.

Generate the Production System

Once you have successfully generated the test system, you can complete the CA ACF2 for z/VM for VM installation procedure. Complete this task only after you have thoroughly tested the product genlevel and all maintenance applied to the test system.

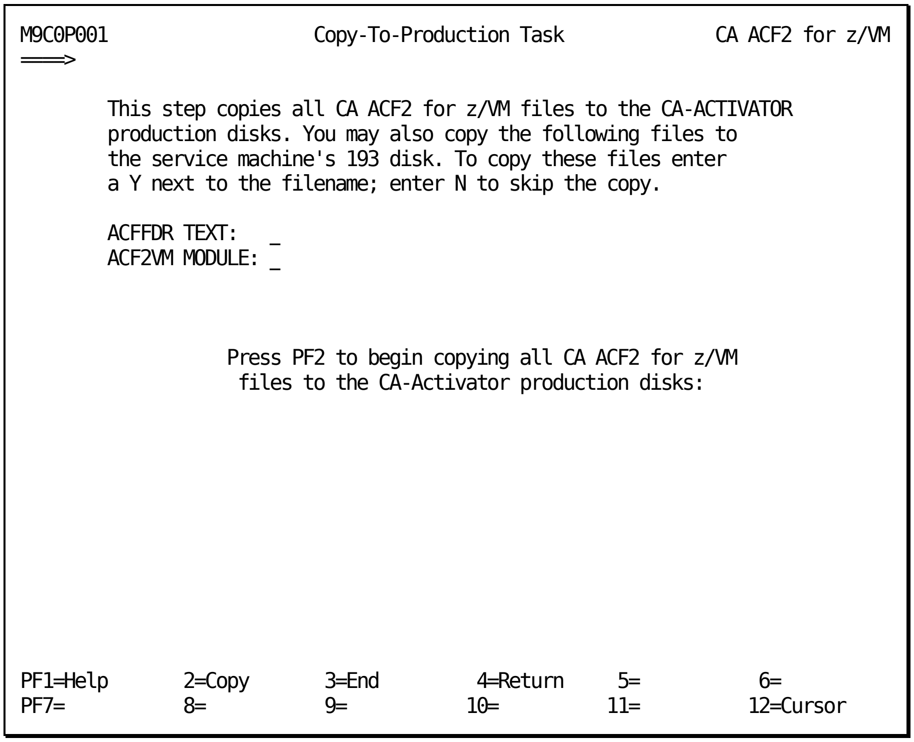
Task M9C0P001: Copy CA ACF2 for z/VM Files to Production

This task lets you copy the CA ACF2 for z/VM for VM files to the CA-ACTIVATOR production disks.

To complete this task:

1. Select task M9C0P001 from the Task Selection Menu. See the CA-ACTIVATOR *Reference Guide* for information about getting to the Task Selection Menu.

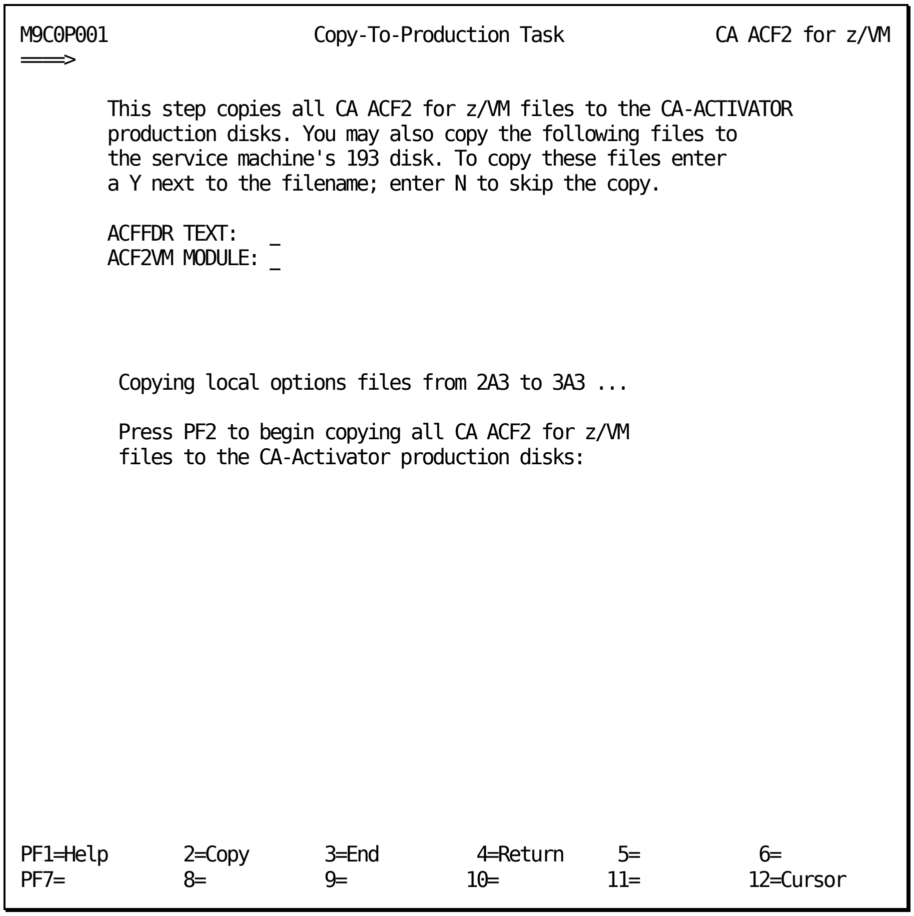
   You see the following panel, which is Option 3 from the Product Installation/Upgrade panel:



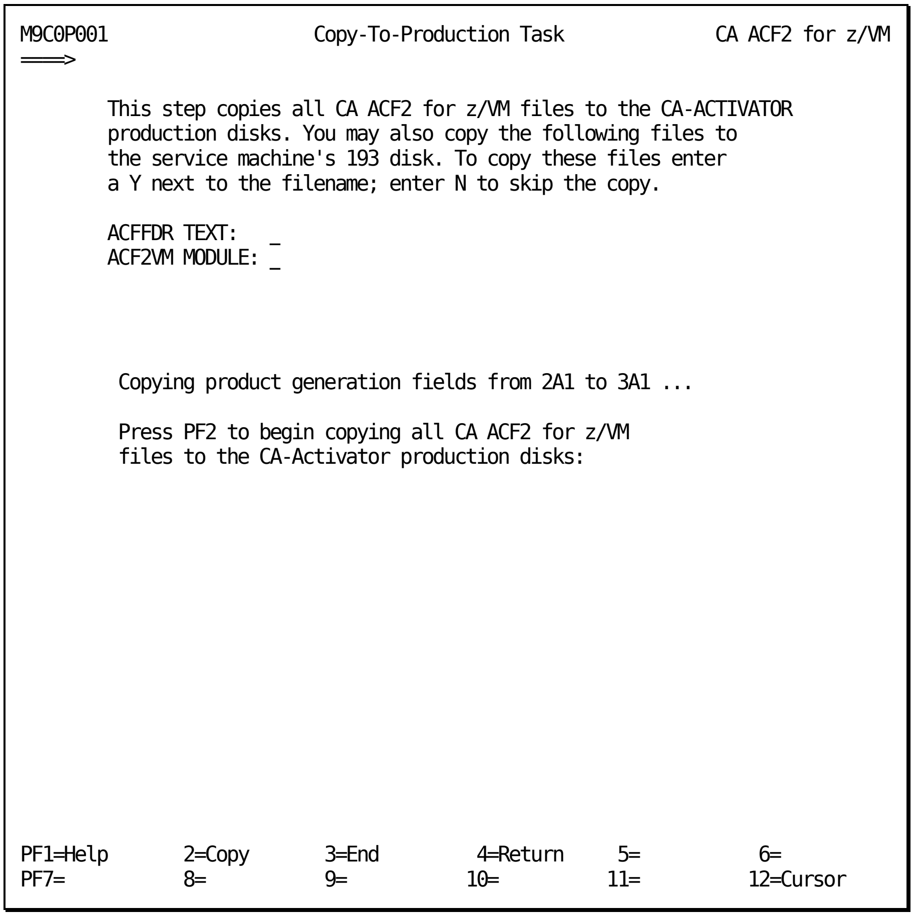
```
M9C0P001                  Copy-To-Production Task              CA ACF2 for z/VM
===>

        This step copies all CA ACF2 for z/VM files to the CA-ACTIVATOR
        production disks. You may also copy the following files to
        the service machine's 193 disk. To copy these files enter
        a Y next to the filename; enter N to skip the copy.

        ACFFDR TEXT:   _
        ACF2VM MODULE: _



                   Press PF2 to begin copying all CA ACF2 for z/VM
                    files to the CA-Activator production disks:










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

A message appears on the panel telling you that CA-ACTIVATOR is copying the files to your production disks.

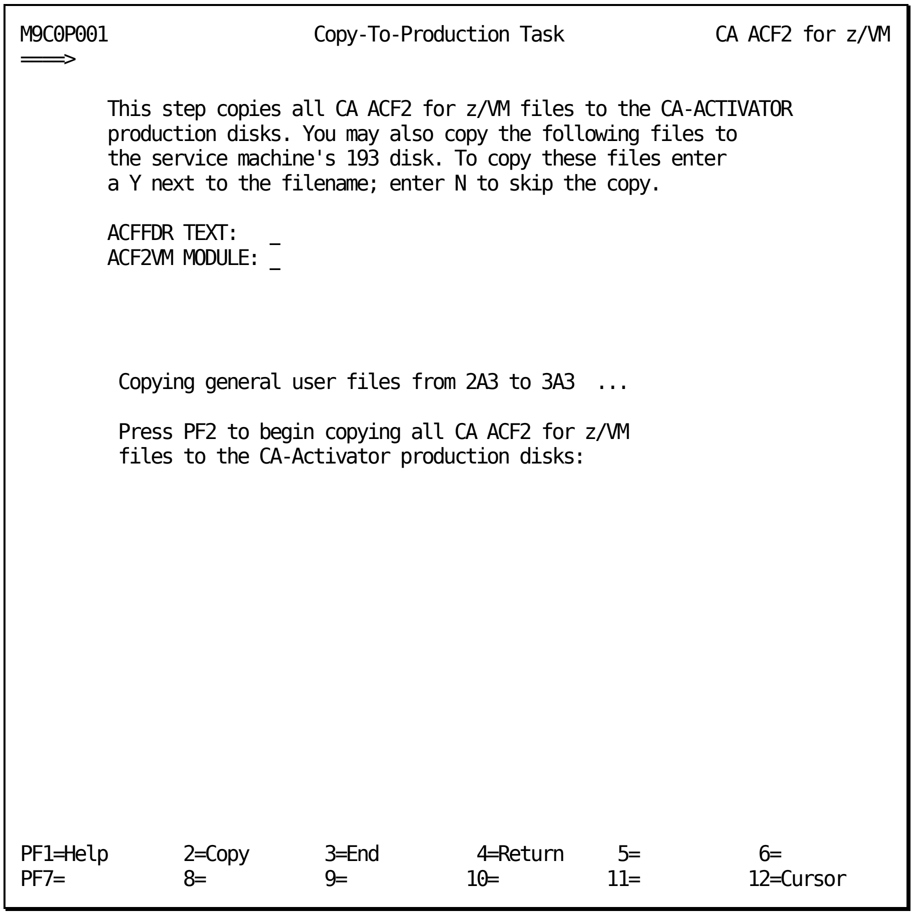
This message changes to tell you what type of files are currently being copied, as shown in the following three panels:



```
M9C0P001                 Copy-To-Production Task               CA ACF2 for z/VM
===>

        This step copies all CA ACF2 for z/VM files to the CA-ACTIVATOR
        production disks. You may also copy the following files to
        the service machine's 193 disk. To copy these files enter
        a Y next to the filename; enter N to skip the copy.

        ACFFDR TEXT:   _
        ACF2VM MODULE: _




         Copying local options files from 2A3 to 3A3 ...

         Press PF2 to begin copying all CA ACF2 for z/VM
         files to the CA-Activator production disks:














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

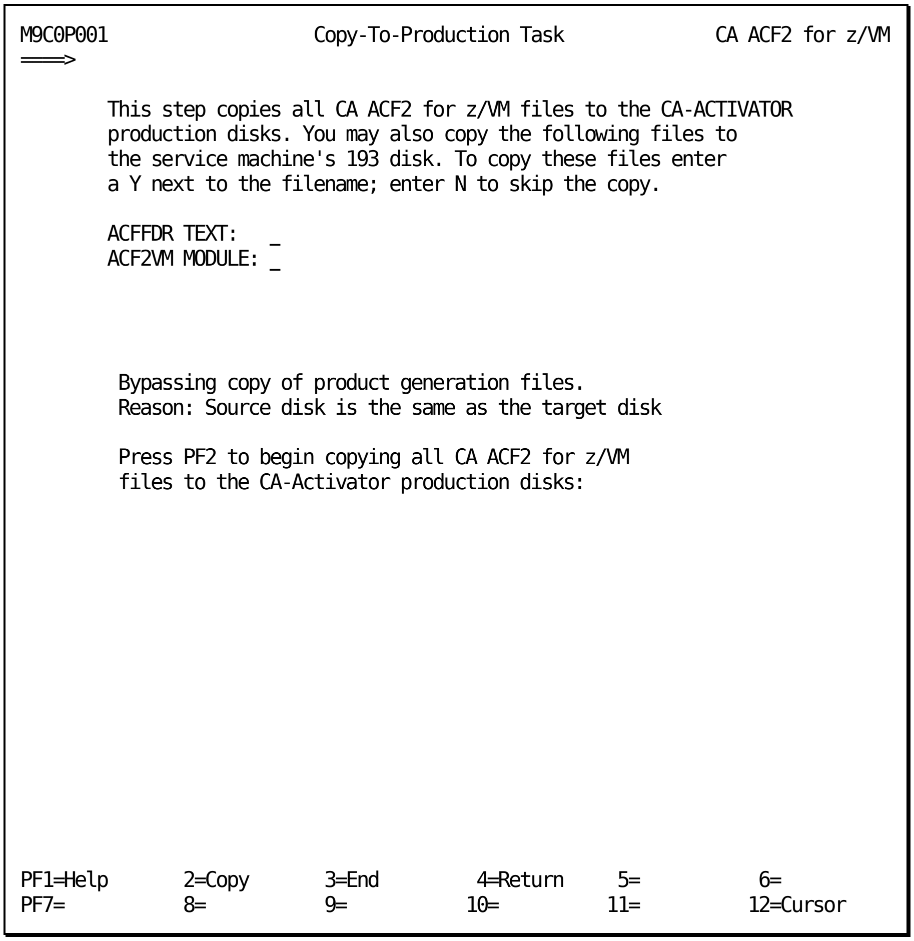


```
M9C0P001                   Copy-To-Production Task              CA ACF2 for z/VM
===>

        This step copies all CA ACF2 for z/VM files to the CA-ACTIVATOR
        production disks. You may also copy the following files to
        the service machine's 193 disk. To copy these files enter
        a Y next to the filename; enter N to skip the copy.

        ACFFDR TEXT:   _
        ACF2VM MODULE: _



         Copying product generation fields from 2A1 to 3A1 ...

         Press PF2 to begin copying all CA ACF2 for z/VM
         files to the CA-Activator production disks:










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



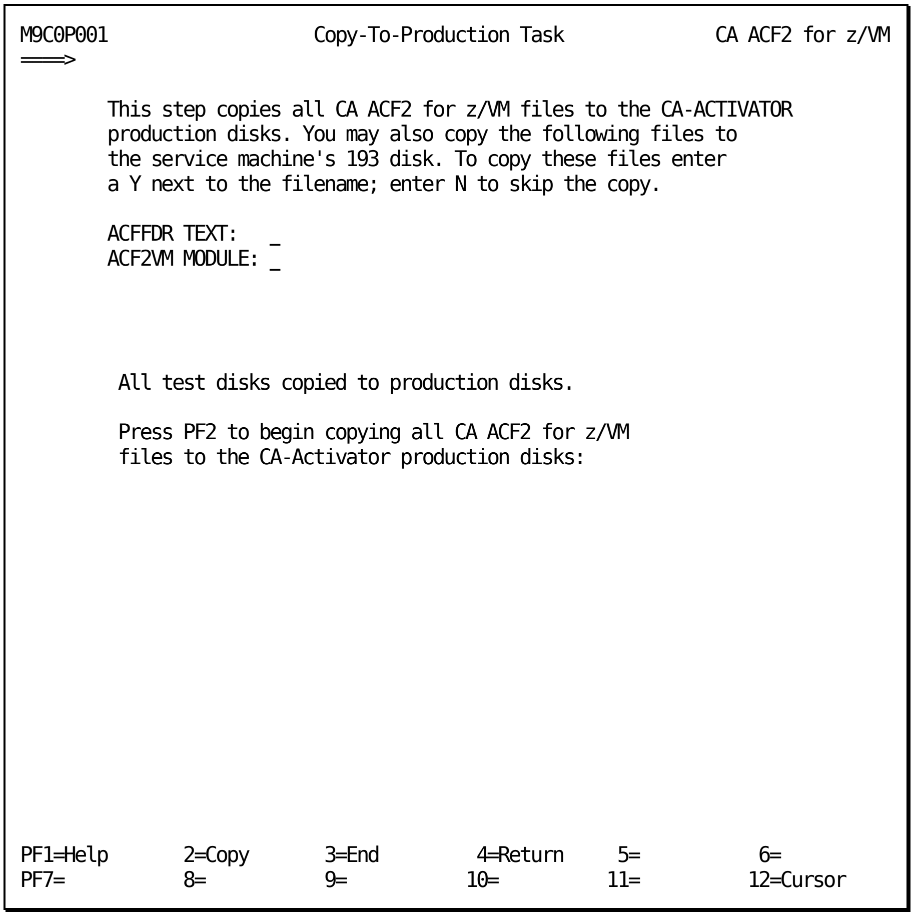
```
M9C0P001                   Copy-To-Production Task              CA ACF2 for z/VM
===>

        This step copies all CA ACF2 for z/VM files to the CA-ACTIVATOR
        production disks. You may also copy the following files to
        the service machine's 193 disk. To copy these files enter
        a Y next to the filename; enter N to skip the copy.

        ACFFDR TEXT:   _
        ACF2VM MODULE: _



         Copying general user files from 2A3 to 3A3  ...

         Press PF2 to begin copying all CA ACF2 for z/VM
         files to the CA-Activator production disks:










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

Note: If your test and production disks are the same, you will see a message telling you that CA-ACTIVATOR has bypassed the copy. For example:

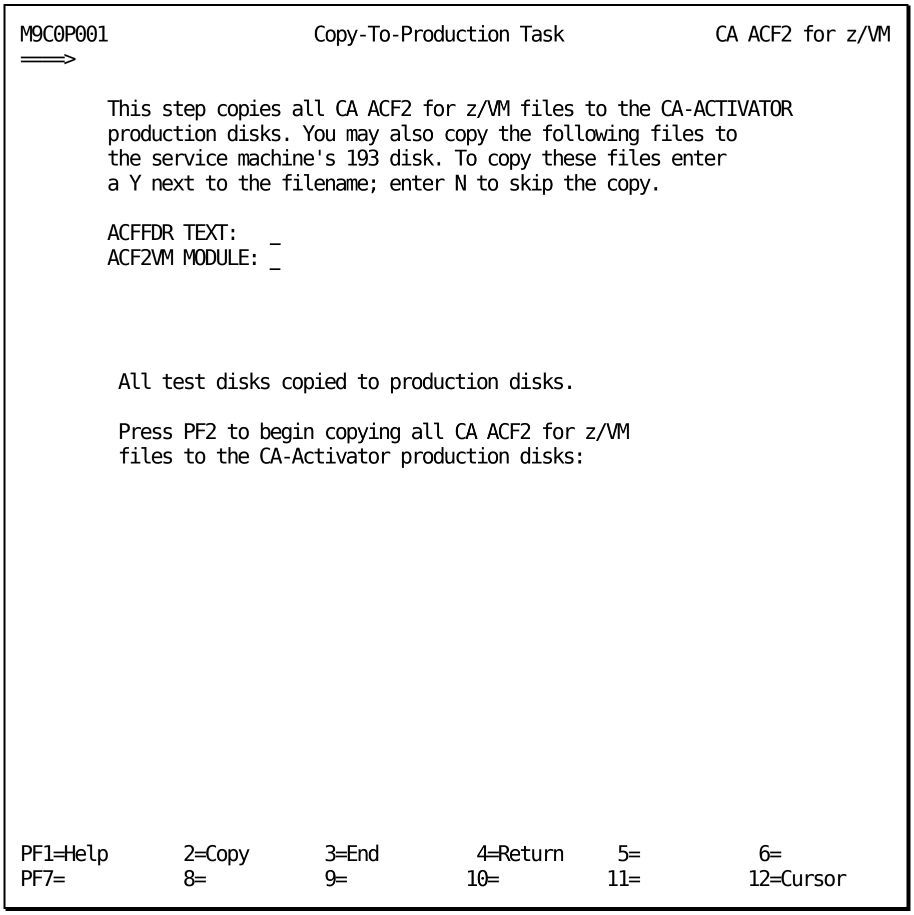


```
M9C0P001                  Copy-To-Production Task             CA ACF2 for z/VM
====>

        This step copies all CA ACF2 for z/VM files to the CA-ACTIVATOR
        production disks. You may also copy the following files to
        the service machine's 193 disk. To copy these files enter
        a Y next to the filename; enter N to skip the copy.

        ACFFDR TEXT:   _
        ACF2VM MODULE: _




         Bypassing copy of product generation files.
         Reason: Source disk is the same as the target disk

         Press PF2 to begin copying all CA ACF2 for z/VM
         files to the CA-Activator production disks:











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

When processing is complete, a message appears on the panel confirming that CA-ACTIVATOR copied all CA ACF2 for z/VM for VM files to the production disks:



```
M9C0P001                Copy-To-Production Task             CA ACF2 for z/VM
===>

        This step copies all CA ACF2 for z/VM files to the CA-ACTIVATOR
        production disks. You may also copy the following files to
        the service machine's 193 disk. To copy these files enter
        a Y next to the filename; enter N to skip the copy.

        ACFFDR TEXT:   _
        ACF2VM MODULE: _




         All test disks copied to production disks.

         Press PF2 to begin copying all CA ACF2 for z/VM
         files to the CA-Activator production disks:












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



```
M9C0P001                  Copy-To-Production Task              CA ACF2 for z/VM
====>

        This step copies all CA ACF2 for z/VM files to the CA-ACTIVATOR
        production disks. You may also copy the following files to
        the service machine's 193 disk. To copy these files enter
        a Y next to the filename; enter N to skip the copy.

        ACFFDR TEXT:   _
        ACF2VM MODULE: _



         All test disks copied to production disks.

         Press PF2 to begin copying all CA ACF2 for z/VM
         files to the CA-Activator production disks:














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

2. Press PF3 to return to the Task Selection Menu.

CA ACF2 for z/VM for VM is now installed on your production system.

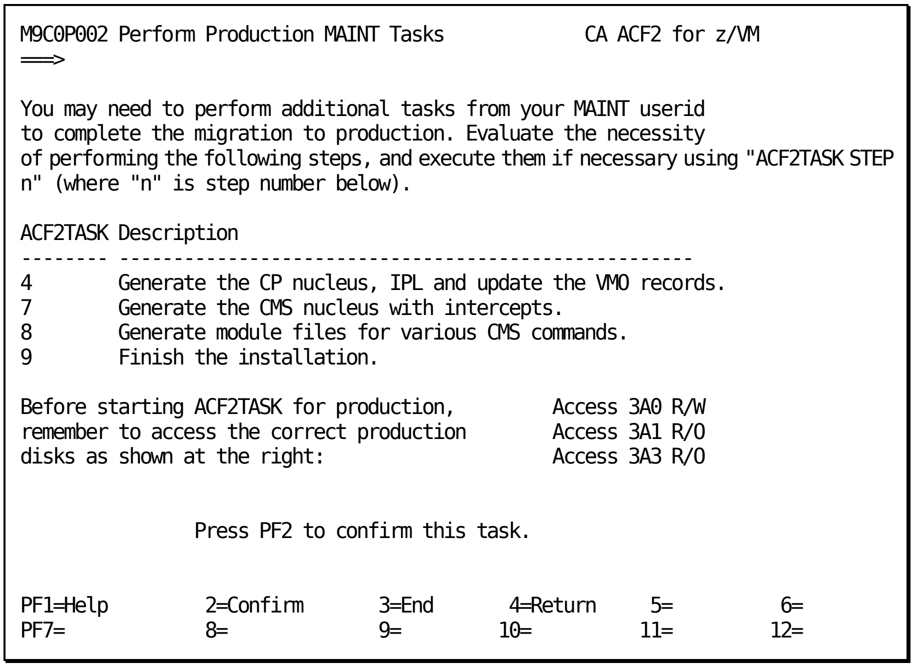
Task M9C0P002: Perform Production MAINT Tasks

This task lets you perform tasks that might be necessary to complete the migration to your production system.

To complete this task:

1. Select task M9C0P002 from the Task Selection Menu.

   You see the following panel:



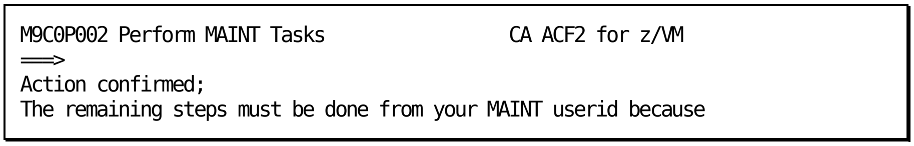
```
M9C0P002 Perform Production MAINT Tasks            CA ACF2 for z/VM
===>

You may need to perform additional tasks from your MAINT userid
to complete the migration to production. Evaluate the necessity
of performing the following steps, and execute them if necessary using "ACF2TASK STEP
n" (where "n" is step number below).

ACF2TASK Description
-------- ----------------------------------------------------
4        Generate the CP nucleus, IPL and update the VMO records.
7        Generate the CMS nucleus with intercepts.
8        Generate module files for various CMS commands.
9        Finish the installation.

Before starting ACF2TASK for production,        Access 3A0 R/W
remember to access the correct production       Access 3A1 R/O
disks as shown at the right:                    Access 3A3 R/O


                Press PF2 to confirm this task.


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

2. Access the specified disks and perform the necessary ACF2TASK steps.

3. Press PF2 to confirm that you want to perform this task. See the “Maintenance ID Installation Steps” section for information about performing these ACF2TASK steps.

4. When you have successfully completed all ACF2TASK steps, return to M9C0P002 and press PF2 to confirm the step is complete.

   A message appears on M9C0P002 confirming the action:



```
M9C0P002 Perform MAINT Tasks                     CA ACF2 for z/VM
===>
Action confirmed;
The remaining steps must be done from your MAINT userid because
```

5. Press PF3 to return to the Task Selection Menu.

Final Installation Tasks

The security administrator or VM systems programmer should now log on using the default CA ACF2 for z/VM logonid of MAINT. CA ACF2 for z/VM prompts for a password the first time this logonid accesses the system.

Reassemble SRF Applications

If you are installing a new release of CA ACF2 for z/VM for VM, reassemble any SRF applications on your system and regenerate any modules for those applications, as necessary.

Insert Logonids

The ACFLIDGN utility created logonids for you during the installation. If there are additional logonids you need to add to the Logonid database, follow the information shown below.

Use the ACF INSERT command to create additional logonids for VM/CMS users. Do this as soon as possible. When you move the CA ACF2 for z/VM service machine to the first level, any user not defined to CA ACF2 for z/VM cannot log onto VM.

The following are ACF INSERT commands:

```
insert tlcjss name(joan s. simmons) phone(x233) security
insert tlcmsb name(mark s. baker) phone(x234) account
```

In the first sample logonid record above, we gave user TLCJSS the SECURITY privilege that lets her write and store CA ACF2 for z/VM access rules for z/OS data sets, minidisks, VSE files, diagnose execution, CP command limiting, and CMS files. The second sample logonid, TLCMSB, has the ACCOUNT privilege, letting him insert, change, and delete CA ACF2 for z/VM logonid records.

To display logonid records, enter the following command:

```
ACF
```

Wait for the response indicating you are in ACF mode, and enter the following command with the logonid that you want to display:

```
LIST logonid
```

The command LIST TLCAMR displays the logonid record named TLCAMR. When you are finished displaying logonid records, enter the following subcommand:

```
END
```

See the *Administrator Guide* for detailed information about privileges and logonid records.

Compile Access Rules

Compile and store access rules as soon as possible to avoid excessive loggings. The sample rule compile sequences below are for both minidisk and CMS file access rules. You can find complete information about rule writing in the *Administrator Guide.*

```
acf
ACF
compile
$key(logonid1)
A allow all users to link to 191 minidisk owned by logonid1
V0191.volume uid(-) read(allow) write(allow)
A allow user logonid2 to read/write the cms file named
A "employee list" owned by logonid1 on the 191 minidisk
V0191.employee.list uid(logonid2) read(a) write(a)
A allow all users to read the cms file named
A "employee list" owned by logonid1
V0191.employee.list uid(-) read(a)
A (or enter a null line)
store
end
```

Compile Command Syntax Models

If you are using CA ACF2 for z/VM command limiting support or performing CA ACF2 for z/VM syntax checking, compile the appropriate set of command syntax models. These models describe the syntax of all standard CP commands. CA ACF2 for z/VM provides the following sets of syntax models.

Use the file appropriate for your site.

ZVMnnn MODEL

Where nnn is the version and release of VM.

To compile the syntax models, enter the following commands:

```
acf
ACF
set model
compile fn
```

The value of *fn* is the filename of the model file in the previous list. The default MDLTYPE from the CMDLIM VMO record is used unless the MDLTYPE is specified on each model in the file being compiled or the MDLTYPE parameter is included on the COMPILE command. For complete information on compiling command syntax models, see the *Command and Diagnose Limiting Guide.*

Compile Command Limiting Rules

You must create command limiting rules for the rules specified in the COMMANDS operand of the CMDLIM VMO record.

Compile Diagnose Limiting Rules

You must create diagnose limiting rules for the rules that are specified in the DIAGS operand of the DIAGLIM VMO record. If you have selected the CA ACF2 for z/VM Diagnose Limiting feature, be sure to write a rule for the 0ACF diagnose CA ACF2 for z/VM uses. Minimally, allow the MAINT user to use the 0ACF diagnose. This lets MAINT IPL a modified CMS system through an IPL command, but prevents other users from IPLing a nonshared secured CMS system.

Compile Resource Rules

Write resource rules for:

- AUTOLOG and XAUTOLOG command support
- DIAL command support
- Group logon support
- Dataspaces
- Inter-User Communication Support (IUCV)
- Advanced Program-to-Program Communication Support (APPC/VM)
- Virtual Machine Communication Support (VMCF)
- Account validation, if you specified ACCTVLD(FULL) in the OPTS VMO record.

Brief samples for each type of resource rule follow. For complete details on how CA ACF2 for z/VM supports each of these facilities (specifically, on related rules, logonid privileges, and validation flow), see the *Administrator Guide.*

Account Validation

Each rule key identifies an account number and the rule entries specify user access permissions for that account.

```
acf
ACF
set resource(act)
RESOURCE
compile
$key(act001) type(act)
uid(rscs1) allow
RESOURCE
store
```

The default three-character resource type code is ACT. However, you can change this locally through the ACCOUNT operand of the RESCLASS VMO record.

A utility named ACFCVACT EXEC is provided to let you create account resource rules from the CP directory. This utility also prepares a file of ACF CHANGE subcommands to create VMACCT field values for logonids in the directory. This VMACCT logonid attribute indicates the default account number for a virtual machine that is account validated.

AUTOLOG and XAUTOLOG Support

Each rule key names the target of a machine you can autolog. Rule entries specify which users can autolog that machine.

```
acf
ACF
set resource(alg)
RESOURCE
compile
$key(maint) type(alg)
uid(syspgr*) allow
RESOURCE
store
```

The default three-character resource type code is ALG. However, you can change this locally through the AUTOLOG operand of the RESCLASS VMO record.

We provide a utility named ACFCVALG EXEC to create AUTOLOG and XAUTOLOG resource rules from the CP directory.

Special Considerations

The following information applies to execution of the AUTOLOG and XAUTOLOG commands:

- AUTOLOG1 (or other virtual machines like it) and disconnected service machines typically issue the AUTOLOG and XAUTOLOG commands. An exec or program running one of these machines issues AUTOLOG and XAUTOLOG to start up other virtual machines.

  When EXECIO or DIAG x'08' does not buffer an AUTOLOG and XAUTOLOG response, the autologging virtual machine could hang, particularly when the machine is forced into a CP read, such as during a password expired condition. To prevent these machines from ever hanging, use EXECIO and DIAG x'08' with message buffering.

  Whenever AUTOLOG or XAUTOLOG ends with a nonzero return code and message buffering is in effect, CA ACF2 for z/VM cancels any CP read, displaying an appropriate message. In such cases, the exec or program that issued the XAUTOLOG or AUTOLOG command must take the appropriate action, such as notifying the operator and displaying all messages. The autologging machine should then continue with its next function. With message buffering, it is the exec or program's responsibility to display the associated messages.

- The PSWDSUP operand of the VMXAOPTS macro in HCPAC0 controls password suppression. If you set PSWDSUP=YES and CA ACF2 for z/VM requires that you supply a password with the AUTOLOG command, you must never enter the password clear text on the command line (CA ACF2 for z/VM considers the password as console input data).

  If you set PSWDSUP=NO and CA ACF2 for z/VM requires you to enter a password, you must supply the password on the command line if you issued the SET PASSWORD AUTOLOG INCLUDE command since the last IPL. If SET PASSWORD AUTOLOG SEPARATE is in effect (the system default), you cannot enter the password on the command line (CA ACF2 for z/VM considers it as console input data).

  There is one more password consideration for XAUTOLOG, when the SET PASSWORD XAUTOLOG INCLUDE mode is in effect. Under this mode, if you turn off password suppression (PSWDSUP=NO) and execute the AUTOLOG command where no password is required, you must issue a dummy password of at least one character with the AUTOLOG command.

DIAL Support

Each rule key names the target of a machine that a user can access using the DIAL command. Rule entries specify who can dial that machine.

```
acf

ACF
set resource(dia)
RESOURCE
compile

$key(testvm) type(dia)
uid(musr-) allow

RESOURCE
store
```

The default three-character resource type code is DIA. However, you can change this locally through the DIAL operand of the RESCLASS VMO record.

You must also define the names of the virtual machines that bypass CA ACF2 for z/VM DIAL command validation, such as VTAM and VTERM. Define these machines with the DIALBYP logonid attribute. All machines without the DIALBYP logonid attribute undergo CA ACF2 for z/VM DIAL resource validation. CA ACF2 for z/VM logs the dials and drops of dialed devices.

Group Logon Support

A *group machine* is any machine that has the CA ACF2 for z/VM logonid attribute of GRPLOGON. Each rule key names the target of a group machine. Rule entries specify who can log onto that machine.

```
acf
ACF
set resource(grp)
RESOURCE
compile

$key(maint) type(grp)
uid(sys-) allow

RESOURCE
store
```

The default three-character resource type code is GRP. However, you can change this locally through the GRPLOGON operand of the RESCLASS VMO record.

Dataspace Support

CA ACF2 for z/VM for VM validates all VM and VM dataspace accesses. This validation occurs at PERMIT time, not at actual access time.

Dataspace protection is activated by default at installation time. 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).

Below is a sample dataspace access rule that permits TLCAMS read access to any dataspaces TLCJAM created, and lets TLCMEG have write access:

```
acf
ACF
set resource (dsp)
RESOURCE
compile

$key(tlcjam) type(dsp)
uid(tlcams) service(read) allow
uid(tlcmeg) service(update) allow

RESOURCE
store
```

Inter-User Communication Vehicle (IUCV) Support

Each rule key names the target of a machine that you can access through an IUCV CONNECT function. The rule entries specify who can CONNECT to that machine.

```
acf
ACF
set resource(iuc)
RESOURCE
compile

$key(%msg) type(iuc)
uid(-) allow

RESOURCE
store
```

The default three-character resource type code is IUC. However, you can change this locally through the IUCV operand of the RESCLASS VMO record. We used the percent sign (%) instead of an asterisk (*) in the rules for CP system services.

You must define the names of the virtual machines and CP system services that CA ACF2 for z/VM controls. Define these machines in the COMSEC operand of the VMXAOPTS macro. The default is COMSEC=(INCLUDE,-). This lets CA ACF2 for z/VM control CONNECTs to all machines. For more information about IUCV support, the *Systems Programmer Guide.*

Advanced Program-to-Program Communication (APPC/VM) Support

Each rule key names the target of a resource ID that you can access through an APPC/VM CONNECT function. The rule entries specify who can connect to that resource ID.

```
acf
ACF
set resource(iuc)
RESOURCE
compile

$key(applsrv) type(iuc)
uid(-) allow

RESOURCE
store
```

The default three-character resource type code is IUC. It is the same as the default for IUCV support. See the Inter-User Communication Vehicle (IUCV) Support section for information about changing the default at your site.

You must define the names of the resource ID that CA ACF2 for z/VM is to control. The COMSEC operand of the VMXAOPTS macro defines these resource IDs. The default is COMSEC=(INCLUDE,-). This lets CA ACF2 for z/VM control CONNECTs to all resource IDs. See the *Administrator Guide* for additional information on APPC/VM support.

Virtual Machine Communications (VMCF) Support

Each rule key names the target of a machine that you can access through a VMCF AUTHORIZE and UNAUTHORIZE function. The rule entries specify who can use VMCF to communicate with that machine.

```
acf
ACF
set resource(vmc)
RESOURCE
compile

$key(dirmaint) type(vmc)
uid(dasdmgr) allow

RESOURCE
store
```

The default three-character resource type code is VMC. However, you can change this locally through the VMCF operand of the RESCLASS VMO record. You must define the names of the virtual machines that CA ACF2 for z/VM controls. Define these machines in the COMSEC operand of the VMXAOPTS macro. The default is COMSEC=(INCLUDE,-). This lets CA ACF2 for z/VM control VMCF communications to all machines. For more information about VMCF support, see the *Systems Programmer Guide*.

Install the Postbackup Service Machine

CA ACF2 for z/VM autologs the postbackup service machine after any backup, whether automatic or the ACFSERVE BACKUP command invoked it. This service machine

- Restores the backup files to an alternate set of databases to build alternate CMS or VSAM databases
- Copies backup files for archiving or to keep multiple backups
- Performs any other user-required postbackup activity.

You can set up this service machine any way you like. CA ACF2 for z/VM only autologs the postbackup service machine. See the *Administrator Guide* for sample procedures.

To activate the postbackup service machine option, modify the AUTOLOG operand of the BACKUP VMO record. This operand specifies the name of the postbackup service machine.

```
set control(vmo)
VMO
CHANGE BACKUP AUTOLOG(ACF2PBSM)
```

ACF2PBSM is the CA ACF2 for z/VM postbackup service machine. Add a directory entry for the service machine:

```
USER ACF2PBSM DUMMYPWD 4M 8M
.
MDISK 191 . . . (a one-cylinder minidisk)
```

Add a logonid record for the postbackup service machine. For example, you can enter the following command:

```
INSERT ACF2PBSM PASSWORD(secret) NOPSWD-EXP VM AUTONOPW AUDIT
```

Modify or create the rule for the CA ACF2 for z/VM service machine to let the postbackup service machine access its disks. Assuming the name of your service machine is ACF2VM, here is a sample rule:

```
$key(acf2vm)
v0195.- uid(acf2pbsm) r(a)
v03a-.- uid(acf2pbsm) r(a) w(a)
```

Create an autolog resource rule to let the service machine autolog the postbackup service machine, as shown below:

```
$key(acf2pbsm) type(alg)
uid(acf2vm) allow
```

Format the postbackup service machine 191 disk, if necessary, and create a profile exec to do the necessary processing. For details on the processing the postbackup service machine performs after CA ACF2 for z/VM autologs it and the sample execs available for configuring this machine to restore the databases, see the *Administrator Guide.*

Restart the CA ACF2 for z/VM Service Machine

There might be times when you want to restart the CA ACF2 for z/VM service machine or you might want to dump the service machine storage and then restart it. To restart the CA ACF2 for z/VM service machine, enter the following command:

```
ACFSERVE RESTART
```

To restart the service machine and take a dump, enter the following command:

```
ACFSERVE RESTART DUMP
```

The default privilege for these commands is SECURITY. The system operator can also issue these commands.

Chapter 6: Generating a CP and CMS Nucleus

This chapter is intended as reference material. For installation instructions see the chapter "Installation Procedure."

To generate an CA ACF2 for z/VM protected CP and CMS nucleus, use the CAXALOAD control file to include and integrate CA ACF2 for z/VM intercept routines in the appropriate nucleus.

The CAXALOAD control files function as an IBM LOADLIST EXEC. When generating a CP or CMS nucleus without CA ACF2 for z/VM protection, the IBM HCPLDR uses the IBM LOADLIST EXEC to identify IBM text files and text file placement in an output load deck. This load deck is link-edited to a CP or CMS nucleus.

When generating a CP or CMS nucleus with CA ACF2 for z/VM protection, the CAXALOAD control files tailor the IBM LOADLIST so you can generate a CA ACF2 for z/VM protected CP or CMS nucleus.

This section contains the following topics:

Types of CA ACF2 for z/VM Control Files (see page 163)
Obtain Control Files (see page 164)
Control Statement Syntax (see page 164)
Bundle Control Statement into Control Files (see page 170)
Generate a Nucleus Using a Base Control File (see page 173)
Check the CAXALOAD Control Statements (see page 174)

Types of CA ACF2 for z/VM Control Files

There are two types of CAXALOAD control files:

- Base control files
- Component control files.

The base control file identifies the base and component control files that update an IBM LOADLIST for a particular VM operating system release. It contains the filenames of base and component control files. The base and component control files contain statements to control the changes applied to a particular IBM LOADLIST. The component control file contains CA ACF2 for z/VM control statements to implement CA ACF2 for z/VM on a specific VM operating system base release and implement site operating system dependencies or customization requirements.

To generate an CA ACF2 for z/VM protected CP and CMS nucleus on a base VM operating system release with no site-developed system dependencies, use the unmodified distributed CAXALOAD control files. (See the Bundle Control Statement into Control Files section for information about site-developed system dependencies.) CA ACF2 for z/VM selects the appropriate control files for you. You do not need to understand the CAXALOAD control statement syntax or how to bundle these control statements into control files.

If you do have site dependencies, CA ACF2 for z/VM can still select the appropriate control files for your release. It automatically tailors your base control file for these dependencies and prompts you for your site dependencies. You do not need to fully understand control statement syntax or packaging.

To customize your CP and CMS nucleus, you need to understand CAXALOAD control file syntax and how to bundle control statements. Put your customized control statement in a component file named USER CAXALOAD or in a USERCMS file for CMS. Then uncomment the component filename in the base control file. Never change the CA ACF2 for z/VM distributed base and component files.

The rest of the sections in this chapter explain:

- How to obtain your CAXALOAD control files
- The CAXALOAD control statement syntax
- How to bundle control statements into control files
- How to generate a CP and CMS nucleus using base control files.

Obtain Control Files

As part of the normal CA ACF2 for z/VM installation process, you create two CAXALOAD update control files; one for CP and one for CMS.

These control files are named ACFCP CAXALOAD and ACFCMS CAXALOAD. These base control files identify the base and component files you need for your specific VM operating system release. You might find it helpful to have printed copies of these control files available while you review the rest of this chapter.

Control Statement Syntax

We can divide the CAXALOAD control statements into six categories, as shown below:

- Adding and relocating modules
- Deleting modules
- Defining intercepts

- Renaming external references
- Bypassing intercept processing
- Comment statements.

Add and Relocate Modules

The syntax for adding or relocating modules in the CP or CMS nucleus is:

```
{CMSNUC      }
{CPIRES      }
{CPPAG       }   [modname modtype] [LANG]
{CPRES       }
{CPRESMP     }
{CPPAGMP     }
```

If one of these control statements identifies an IBM CP module (defined in an IBM CP LOADLIST), CA ACF2 for z/VM relocates it to the CP nucleus portion the control statement keyword describes.

Note: In z/VM Version 5 Release 1.0 (and above) the entire CP nucleus is resident. Therefore, CPPAG, CPPAGMP, and CPIRES should no longer be used. Only CPRES and CPRESMP are valid in z/VM Version 5 Release 1.0 and above.

CMSNUC

Includes the CA ACF2 for z/VM or user modules in the CMS nucleus to manipulate CA ACF2 for z/VM and user CMS modules. Only the CMS CAXALOAD files use it. A portion of the base CMS CAXALOAD control file contains a keyword for managing an CA ACF2 for z/VM module:

```
*************************************************************
* ACF2 MODULES FOLLOW:                                      *
*************************************************************
CMSNUC DMSAC0
```

Do not delete the CMSNUC keyword. You can add additional CMSNUC keywords to customize your CMS load deck. Add these keywords to your own USERCMS CAXALOAD component control file. (See the Bundle Control Statement into Control Files for additional information.)

CPIRES

Includes CA ACF2 for z/VM, IBM, or user modules in the IPL-resident portion of the CP nucleus.

CPPAG

Includes CA ACF2 for z/VM, IBM, or user modules in the pageable portion of the CP nucleus.

CPRES

Includes CA ACF2 for z/VM, IBM, or user modules in the resident portion of the CP nucleus.

CPRESMP

Includes CA ACF2 for z/VM, IBM, or user modules in the CP resident MP (multiprocessing) nucleus.

CPPAGMP

Includes CA ACF2 for z/VM, IBM, or user modules in the CP pageable MP (multiprocessing) nucleus.

modname

The filename of the module placed in the CP or CMS nucleus.

modtype

The optional filetype of the module placed in the CP or CMS nucleus. If you do not specify a filetype, your site's standard control (CNTRL) file determines the text file search order.

CA ACF2 for z/VM r12 supports the CPMOVIRS, CPMOVPAG, and CPMOVRES keywords to relocate modules. However, future CA ACF2 for z/VM releases may not support these keywords. We recommend you convert these statements for module relocation as follows:

CPMOVIRS

Change to CPIRES

CPMOVPAG

Change to CPPAG

CPMOVRES

Change to CPRES.

Delete Modules

The syntax for deleting modules in the CP or CMS nucleus is:

```
CPDEL modname
```

CPDEL

Deletes user-specified modules from the nucleus

modname

Indicates the module to delete.

Define Intercepts

INTERCPT control statements gain control before CP or CMS entry points without changing the CP or CMS module source code changes. CA ACF2 for z/VM identifies the front-ended CP or CMS entry point as the intercepted entry point. It replaces the intercepted entry points by one or more intercepter entry points. These intercepter entry points gain control in a predetermined order before finally giving control to the intercepted entry. It modifies all modules that refer to intercepted entry points to call the identified intercepter entry points.

The syntax to call intercepter entry points is:

```
             [APPEND]
INTERCPT     [REPLACE ] intrpt  [intrpt1...5]
             [CLEAR   ]
```

INTERCPT

Identifies entry points to intercept.

APPEND

Adds a new intercepted entry or appends an intercepter entry to an existing intercepted entry point. APPEND is the default setting.

REPLACE

Adds a new intercepted entry or replaces intercepter entries for existing intercepted entry points.

CLEAR

Removes matching INTERCPT statements. The key is the intercepted entry point.

intrpt

Specifies the intercepted entry point (required).

intrpt1...5

Specifies up to five intercepter entry points. When you specify more than one intercepter, the intercepters ordered on the right side of the INTERCPT statement gain control first, moving left until the intercepted entry gains control last. You must specify at least one intercepter.

Intercept Processing

The following is an excerpt showing how INTERCPT keywords appear in a sample CAXALOAD control file for CP (not necessarily the exact keywords we shipped):

```
LISTTYPE CP        <=== Defines this as a CP load list
   .
*
*       INTERCEPTED   *************  INTERCEPTER ENTRIES
*          ENTRY      LEVEL 1   LEVEL 2   LEVEL 3   LEVEL 4   LEVEL
*        ********     ********  ********  ********  ********  *****
*
INTERCPT HCPCFCMD     HCPAF7CF
INTERCPT HCPCFPRR     HCPAA1RR
```

This portion of the file contains only one level of intercepters. The intercept hierarchy can specify up to five levels of intercepters. If you specified three intercepters on a control statement, the one immediately after the intercepted entry is the first level intercepter, the middle one is the second level intercepter, and the one on the right is the third level intercepter.

The following is an example of an INTERCPT record that has two intercepter entry points:

```
INTERCPT HCPCFGII     HCPAA9II HCPAB3II
```

The logic for this intercept record is:

- The IBM module HCPCFG calls HCPCFGII
- HCPCFGII is intercepted by HCPAB3II, the second level intercepter
- HCPAB3II gains control and calls HCPAA9II, the first level intercepter
- HCPAA9II gains control and calls HCPCFGII, the intercepted entry point
- HCPCFGII gains control and terminates
- HCPAA9II, the first level intercepter, regains control and terminates
- HCPAB3II, the second level intercepter, regains control and terminates
- The IBM module HCPCFG regains control.

The coded logic for the HCPAB3II and HCPAA9II modules is such that they both have internal calls to HCPCFGII. However, the CA ACF2 for z/VM loader program handles the entry point intercept logic for these modules, and bases pathing on the position of the module in the INTERCPT control record. Do not change any of the supplied INTERCPT records without consulting CA ACF2 for z/VM Technical Support.

Rename External References

The RENAMEXT control statement renames external references. It is a reference to an external routine in a CSECT (a V-type address constant in assembler). The entry point the rename-from external reference refers to may already exist. It could also be an unresolved reference.

A text filename limits the rename to apply only in a specified text file. If you do not specify a text filename, the rename applies to all text files containing the rename-from external reference.

RENAMEXT also changes all CAXALOAD control statement entry point names that match the rename-from external reference name to the rename-to name.

The syntax to rename external references is:

```
RENAMEXT [CLEAR] renfrm rento [textfn]
```

RENAMEXT

Identifies external references to rename.

CLEAR

Removes matching RENAMEXT statements. The key is the from and to external reference names. This parameter is optional.

renfrm

Specifies the rename-from external reference (required).

rento

Specifies the rename-to external reference (required).

textfn

Specifies the text filename that contains the external references to rename (optional).

An example of a fictional RENAMEXT record is:

```
RENAMEXT HCPRPICN HCPAF1CN HCPIUG
```

In this example, the rename-from external reference is HCPRPICN. (This can be an unresolved reference.) The rename-to external reference is HCPAFICN. The HCPIUG text file name limits the rename function to only this text file.

Bypass Intercept Processing

The syntax for bypassing intercept processing is:

```
BYPASS [CLEAR] module [extrn]
```

BYPASS

Identifies modules that contain external references that are not intercepted.

CLEAR

Removes matching BYPASS statements. The key is the module name and extrn you specified, if any.

MODULE

Specifies the name of the module containing the external references that intercept processing should not alter (required).

extrn

Indicates the external reference in the specified module that is to remain intact during intercept processing (optional).

The extrn operand identifies the only external reference in the module. An example of a fictional RENAMEXT record is:

```
RENAMEXT HCPRPICN HCPAF1CN HCPIUG
```

Specify multiple BYPASS statements to identify multiple extrn in one module. A sample of specifying multiple BYPASS statements follows:

```
BYPASS     HCPVIO  HCPHVCAL
BYPASS     HCPSYM
```

The first BYPASS statement bypasses the external reference HCPHVCAL in the HCPVIO module. With this statement, DMKHVCAL is the only external reference that remains intact during intercept processing. The second BYPASS record bypasses all INTERCPT external reference processing for the module HCPSYM; all external references to HCPSYM remain intact during intercept processing.

Do not insert BYPASS records for any of the INTERCPT records CA ACF2 for z/VM supplies. Improper use of BYPASS records can cause integrity exposures.

Bundle Control Statement into Control Files

The following sections explain how to use the base and component control files to customize your system.

Base Control Files

CA ACF2 for z/VM automatically builds base control files for you or you can build your own. Each base control filename identifies the CP or CMS release. These files contain a list of component control filenames that the CA loader program (CAXALOAD) processes. Each CA ACF2 for z/VM component control file may be required, depending on your CP or CMS release, generation level, and your site dependencies. Site-dependent component control files define options that CA ACF2 for z/VM must know about (for example, user-selected VSAM database sharing).

CA ACF2 for z/VM prompts for information regarding your CP and CMS release and site dependencies, then tailors your base control file for you. Edit the base control file to make necessary modifications (such as adding a component file).

The following is a diagram showing you how CA ACF2 for z/VM builds the base control file:

```
LISTTYPE CONTROL
component control filename
     .
     .
     .
component control filenamex
```

In the previous example,

LISTTYPE

Must be the first uncommented statement (identifies this file as a base control file)

component control filename

Component CAXALOAD control statement files. Options of the CA ACF2 for z/VM component files name the option they provide. CA ACF2 for z/VM selects the appropriate component files for you. See your base control file (ACFCP or ACFCMS) for the component files that apply to your CP and CMS release.

When you run a CA loader program, it processes the base control file base and component CAXALOAD files and builds an incore table of CAXALOAD control statements. Component control files that appear lower in the base control file can override or augment previously listed component control files. As CA ACF2 for z/VM reads each CAXALOAD statement, it replaces or adds duplicate entries to the table, as required by the control statement syntax. See the Add and Relocate Modules section for this syntax. The combined CAXALOAD incore table drives your CA ACF2 for z/VM CP/CMS nucleus generation.

Sample CP Base Control File

The following is an example of a CP CAXALOAD base control file:

```
LISTTYPE CONTROL
*
 ACFCP630          * VM 6.3.0 CAXALOAD component file for CP
 UCENGRES          * Optional uppercase English lang support
*USER              * User maintenance override file
***
```

We do not supply the USER component control file. The USER component control file should be last.

If you create a USER component control file, uncomment the USER statement in the base control file. To uncomment the USER statement, remove the asterisk preceding USER.

Sample CMS Base Control File

The following is an example of a CMS CAXALOAD base control file:

```
LISTTYPE CONTROL
 ACFCMS27         * VM CMS 27 CAXALOAD component file
*USERCMS          * User maintenance component file
***
```

We do not supply the USERCMS component control file. The USER component control file should be last.

If you create a USERCMS component control file, uncomment the USERCMS statement in the base control file. To uncomment this statement, remove the asterisk preceding USERCMS.

Component Control Files

Component control files contain functional statements for module names required for your VM operating system release. Do not change these files. Use the USER component files to make changes.

The optional features of the component control files or an IBM APAR categorize optional statements they provide (such as shared database support). They can be the ones shipped with CA ACF2 for z/VM or user-generated.

Sample UCENG Optional Component File

Listed below is a sample module for the upper case English language support:

```
***
* Following module added for uppercase English language support
*
CPPAG HCPAL0
***
```

Generate a Nucleus Using a Base Control File

This section contains information for generating a nucleus when you have defined base control files for your system.

Execute CAXABLD

CAXABLD is an CA ACF2 for z/VM exec that runs a CA loader program named CAVMLOAD. CAXABLD is a front-end to the IBM VMFBLD EXEC that CA ACF2 for z/VM invokes as part of the CAXABLD procedure.

CAVMLOAD reads information from control files that must have a filetype of CAXALOAD. These CAXALOAD control files perform two main functions:

- Specify IBM-defined entry points that CA ACF2 for z/VM security modules intercept to implement in object-code-only form.
- Dynamically manage the LOADLIST modules to augment the function of the LOADLIST EXEC. As a result, you do not need to modify the LOADLIST EXEC. Also, you can modify the CAXALOAD base control file or USER component control file to customize the LOADLIST.

CA ACF2 for z/VM invokes the CAVMLOAD module as part of the CAXABLD procedure. The syntax of the CAXABLD command is:

```
CAXABLD acf2cntl command
```

acf2cntl

The filename of the CA ACF2 for z/VM update control file that is applicable to your VM system. The filetype is always CAXALOAD. Create these files during the CA ACF2 for z/VM installation procedure to accommodate the versions of CP and CMS you are running on your system:

- ACFCP generates a CP nucleus that includes CA ACF2 for z/VM Protection
- ACFCMS generates a CMS nucleus that includes CA ACF2 for z/VM protection.

command

The command that generates the system without CA ACF2 for z/VM.

The following examples demonstrate how to use CAXABLD to generate a CP nucleus that includes CA ACF2 for z/VM protection:

- Enter the following:

  ```
  CAXABLD ACFCP VMFBLD PPF ACFZnnn CP CPLOAD32 (ALL
  ```

Whenever you execute CAXABLD, you must specify the CAXALOAD base control filename.

When CAXABLD invokes the CAVMLOAD module to generate your CP or CMS nucleus, it does the following:

- Produces a file named CAXALOAD SUMMARY
- Renames the $$$TLL$$ EXEC procedure the IBM VMFBDNUC EXEC produces to $$$TLL$$ IBMEXEC
- Produces a new exec named $$$TLL$$ EXEC that includes all CAXALOAD modifications to the LOADLIST EXEC.

Check the CAXALOAD Control Statements

To check the CAXALOAD control statements for syntax validation or to view the resulting combined CAXALOAD update, base, and component control files, issue the following command:

```
CAXABLD CHECK acf2cntl
```

The following is an explanation of the syntax:

CHECK

The command to begin syntax checking.

acf2cntl

The filename of the CA ACF2 for z/VM control file applicable to your VM system (ACFCP or ACFCMS). The filetype is CAXALOAD. Create this file during the CA ACF2 for z/VM installation procedure to accommodate the version of CP CMS you run on your system. Valid values are:

- ACFCMS generates a CMS nucleus that includes CA ACF2 for z/VM protection
- ACFCP generates a CP nucleus that includes CA ACF2 for z/VM protection.

This check of control statements creates a file with a filename of CAXALOAD and a filetype of CHECK.

If the summary in the next section is correct, you can generate your nucleus now.

Sample CAXALOAD CHECK Summary Log for CP

The following is a list of definitions for terms used in the sample:

CNTRL

Identifies your standard control file.

LOADLIST

Identifies your standard LOADLIST EXEC file.

UPDCNTRL

Identifies the CAXALOAD base control file.

CAXALOAD

Identifies all CAXALOAD base and component control files the UPDCNTRL file identifies. CA ACF2 for z/VM processes all base and control files in top to bottom order.

FUNCTION

Represents the composite CAXALOAD table.

ORIGIN

Identifies the CAXALOAD filename of the file where this control statement originated (if no components applied), by default. If a component did apply to this control statement, it identifies the last file containing the overriding statement.

Sample CAXALOAD Summary Log for CP

The CAXALOAD Summary Log for CP contains information that reflects the references that the ACFCP CAXALOAD control file modifies. When you invoke CA loader program during installation to generate a CP nucleus, it produces the log.

Chapter 7: Installation Options

This section contains the following topics:

CAISSF (see page 177)
CMS Batch Facility (see page 178)
DASD Dump and Restore (DDR) Support (see page 179)
Database Synchronization Component (see page 180)
DIAG 'A0' Subcode '04': CA ACF2 for z/VM Password Validation (see page 193)
Support for DirMaint (see page 194)
National Language Support (see page 197)
PAM Server Support (see page 204)
PASSTHRU Release 1.3 Support (see page 218)
RSCS System Component (see page 219)
CA-REGISTER (see page 220)
Shared Database Support (see page 222)
TCP/IP FTP Interface (see page 237)
TCP/IP NFS Interface (see page 240)
TCP/IP REXEC Interface (see page 241)
Unit Record Logging Support (see page 242)
VM Batch System Support (see page 243)
VSAM AMSERV Support (see page 250)

CAISSF

The CA Standard Security Facility (CAISSF) is an interface for CA and other products. CAISSF calls CA ACF2 for z/VM to perform access validations. This section contains information on implementing CAISSF. See the *Administrator Guide* for information about writing rules for CAISSF.

Before you can implement CA ACF2 for z/VM security for CAISSF, your system must meet the following conditions:

- You must identify the virtual machine to use the System Request Facility (SRF). See the *System Programmer's Guide* for information about on SRF.
- You must enable the SRF logonid privilege to allow the virtual machine to issue SRF requests. See the *System Programmer's Guide* for additional information.

- You must make the CAS9SEC MODULE, the translator module, and the ACFSRF LOADLIB available to the virtual machine. You can call the translator CAS9M9*rr*, where *rr* is the CA ACF2 for z/VM release number.
- You must define the resource class in the SSFTYPE VMO record. The format of this resource class is an eight-character resource name that CA ACF2 for z/VM translates to a three-character resource type. Each product will inform you of the resource names they use. See the *Administrator Guide* for information about the SSFTYPE VMO record.

CMS Batch Facility

There is a CA ACF2 for z/VM interface for the CMS Batch system component that provides logonid and password inheritance for jobs submitted to the CMS Batch. To submit a job to the CMS Batch under your own logonid, you do not need to perform any additional steps (such as inserting a password in your job).

The interface also lets you submit a CMS batch job on behalf of another user. We provide an exec, called SUBATCH, that prompts you for an CA ACF2 for z/VM password. SUBATCH automatically inserts a password into the job stream before it submits it.

CMS Batch requires you to supply link passwords in the CMS Batch job whenever a CP link command is executed. With the CA ACF2 for z/VM interface, even if link password check is ignored, you must still supply a dummy password in the CMS Batch job.

Required BATEXIT2 Exit

You must use the BATEXIT2 exit we provide with CA ACF2 for z/VM to implement the CMS Batch interface.

Installing the CMS Batch Interface

Use the following procedure to install the CMS Batch interface:

- Give each CMS batch machine the VMD4TARG logonid privilege. We recommend you also assign the AUTOONLY privilege to the CMS batch machine to prevent its use on a terminal device. **Do not** assign the CMS batch machine additional CA ACF2 for z/VM privileges. The only privileges you should grant to this machine are VMD4TARG, the system VMCHK value, and AUTOONLY.
- Assemble the supplied CA ACF2 for z/VM version of BATEXIT2 and place it on the S-disk. If you use this exit for local purposes, be sure to compare the supplied CA ACF2 for z/VM version to yours, make any necessary adjustments, and then merge the two versions.

- Copy the supplied CA ACF2 for z/VM SUBATCH EXEC to a commonly accessible disk, such as the Y-disk.
- Copy the supplied CA ACF2 for z/VM SUBTPRF XEDIT to the same disk.
- Copy the supplied CA ACF2 for z/VM SUBTPRF2 XEDIT to the same disk
- Copy the supplied CA ACF2 for z/VM BATPROF EXEC to the same disk that contains the supplied CA ACF2 for z/VM version of BATEXIT2 (usually the S disk). Review BATPROF EXEC and modify it to reflect your CMS batch virtual minidisk configuration. CA ACF2 for z/VM automatically detaches any minidisk this exec does not include at the start of each new job. This ensures subsequent jobs cannot access the minidisk of a previous job.
- Define the CMS batch machine strictly as a class G virtual machine in the VM directory.
- Installation is now complete.

Submitting Jobs on Behalf of Another User

To submit a CMS batch job on behalf of another user, use the SUBATCH EXEC to enter the password and have CA ACF2 for z/VM automatically insert it into the job stream. Or you can manually insert a job statement with the password using the following format:

```
/JOB machid acct# job PX password
```

DASD Dump and Restore (DDR) Support

This section contains information for installing DASD Dump and Restore support. CA ACF2 for z/VM protects the DDR module only when it resides on the CMS system disk. To turn on DDR protection, the security administrator writes a rule that makes the DDR MODULE execute-only to everyone on the system. In addition, you must not allow DDR to copy the CMS system disk.

If the CMS system disk is the $CMSSYS$ 190 disk, the following rule set correctly activates DDR security:

```
acf
ACF
$key($cmssys$)
 v0190.ddr:module.uid(-) exec(a)
 v0190.-  uid(-) pgm(ddr)
 v0190.-  uid(-) read(a) exec(a)
```

CA ACF2 for z/VM assumes the DDR function belongs to the owner of the CMS system disk. Therefore, it does not validate the owner of the CMS system disk for DDR. To avoid this circumvention of DDR security, there are two possible courses of action:

1. Put the actual MDISK entry for the CMS system disk under a NOLOG user's directory entry and replace the old MDISK entry for this disk (MAINT 190) with a LINK to the real CMS system disk. Then, if users normally link to MAINT 190, MAINT's entry for 190 is LINK $CMSSYS$ 190 190 RR. This is not necessary if MAINT is a NOLOG user.
2. Change the PREFIX field of the CMS system disk owner so that CA ACF2 for z/VM no longer recognizes this user as the owner of its own disks. This requires rules allowing the user to get at their own disks and files.

If you are only logging DDR validations, you normally see both a READ and a WRITE logging each time a user uses DDR to write on a disk because DDR always reads from an output disk before it actually writes to the disk. CA ACF2 for z/VM also performs a format validation for DDR output operations. Thus, if you are logging (but not preventing) both DDR and FORMAT operations, you see a DDR READ, a DDR WRITE, and a FORMAT logging for a single output operation.

We recommend you secure the IPL DDRXA standalone module from general system users. For example, you can place it on a MAINT minidisk so that users cannot use it. You can use CA ACF2 for z/VM command limiting to prevent users from IPLing nonsecured virtual machines.

Database Synchronization Component

The Database Synchronization Component (DSC) is specifically designed for sites that are sharing CA ACF2 for z/VM databases between VM and z/OS. It is a formal response to IBM dropping support of non-ICF catalogs in z/OS after December 31, 1999, and ICF catalog support is not being added to VM or VSE. The DSC allows synchronization of CA ACF2 for z/VM database changes between VM and z/OS systems using CAICCI (and VTAM) as the communications method.

DSC provides the necessary features that keep your CA ACF2 for z/VM Security for z/OS and CA ACF2 for z/VM for VM databases synchronized. That is, all updates, inserts, changes, and deletes you make to any CA ACF2 for z/VM record, from z/OS or VM, are shipped (propagated) to the target systems. You can journal any change you make with DSC to a log file. In case one of the two systems is unavailable, the Database Synchronization Component service machine stores the request in a communication log file until the change is shipped to the target system for processing. When the system becomes available and the necessary connections are in place, the Database Synchronization Component service machine ships any changes that were logged and not processed.

Note: See Database Synchronization Component Processing for information about using the DSC.

Prerequisites

To use DSC:

- VTAM (any IBM-supported release) must be installed and operational.
- CA-CIS Services release 1.0, genlevel 0002 or above, including CAICCI, must be installed and operational.
- CA ACF2 for z/VM r12 must be installed.
- Be sure the same set of @CFDE macros exist on each system that you are synchronizing.
- Be sure that the time-of-day (TOD) clocks on all systems are synchronized.
- Be sure that all databases are equal before you start using the Database Synchronization Component.

Installing the Database Synchronization Component

To install the Database Synchronization Component (DSC), you need the Database Synchronization service machine. The name of this machine can be any name you choose. Specify the name in the DBSYNC VMO record. If not specified in the DBSYNC VMO record, the default is ACFSYNC. This guide uses ACFSYNC as the name of the Database Synchronization Component service machine in all examples.

To install the Database Synchronization Component:

1. Add the ACFSYNC service machine to the VM directory. You can use the supplied file ACFSYNC DIRECT as a sample directory. Edit this file and make any changes necessary.

   The three MDISK statements required for the ACFSYNC machine are commented out in the supplied ACFSYNC DIRECT. You must change these MDISK statements before you add ACFSYNC to the VM directory or you must add them just after you add ACFSYNC to the VM directory, depending on your directory maintenance software and site standards and procedures.

   The three minidisks are:

   - 100-ACF2DSC PARMS file and the ACFSYNC nucleus
   - 191-PROFILE EXEC for the ACFSYNC machine
   - 600-ACFSYNC recovery file.

2. Add the CA ACF2 for z/VM logonid record for ACFSYNC and give it the ACCOUNT, SECURITY, and AUTONOPW privileges. Also add any other changes your site standards and procedures require.

3. Modify the DBSYNC VMO record to indicate which databases you want to synchronize, and the name of the ACFSYNC and CCIVM service machines. See the *Administrator Guide* for information about changing the DBSYNC VMO record.

4. If you want to allow access to the ACFSYNC minidisks, write an access rule for ACFSYNC now.

5. Logon to the ACFSYNC service machine and format the minidisks as follows:

6. Use the following command to format the ACFSYNC 100 disk as a standard CMS minidisk to prepare it for the ACFSYNC nucleus:

   ```
   FORMAT 100 Z
   ```

   The above command prompts to make sure the format is appropriate. It also prompts for the DISK label. The disk label can be any name (for example, DSC100).

7. Issue the following command to reserve the first cylinder of the two-cylinder 100 disk as available for use by CMS files.

   ```
   FORMAT 100 Z 1 (RECOMP
   ```

   This command also leaves the last cylinder of the disk for the ACFSYNC nucleus.

8. Format the ACFSYNC 191 disk as a standard CMS minidisk with the following command:

   ```
   FORMAT 191 A
   ```

   This command prompts you to make sure the format is correct. It also prompts for the DISK label. You can name the label any appropriate name (for example, DSC191).

9. Link to the CAIMAINT minidisk where you installed CA ACF2 for z/VM, normally the CAIMAINT 291 disk. Access this disk as the B-disk.

   ```
   LINK CAIMAINT 291 291 RR
   ```

   ```
   ACCESS 291 B
   ```

10. Create the PROFILE EXEC for the ACFSYNC machine with the following command:

    ```
    COPYFILE ACFSYNC PROFILE B PROFILE EXEC A
    ```

11. You can edit the new PROFILE EXEC file on the A-disk. The only change that is required is if your site uses a disk other than the CAIMAINT 291 disk. This profile IPLs the nucleus on the ACFSYNC 100 disk if the ACFSYNC machine is running disconnected. If the ACFSYNC machine is logged on normally, it assumes maintenance is being done and stays running CMS and accesses the CAIMAINT 291 disk.

12. Create the ACF2DSC PARMS file on the ACFSYNC 100 disk with the following command:

    ```
    ACCESS 100 Z
    ```

    ```
    COPYFILE ACF2DSC PARMSAMP B ACF2DSC PARMS Z
    ```

13. Edit the ACF2DSC PARMS file and make any local modifications that are needed. Refer to the section titled Parameter Files for more information.

14. Prepare the ACFSYNC 600 disk for use as the ACFSYNC recovery file. This file saves unprocessed database synchronization transactions until they can be processed. Issue the following commands:

    ```
    DSCCATDK 600 DSCFIL
    ```

    ```
    DSCMAI DSC
    ```

    Note: The 600 disk must have a VOLSER of DSCFIL, and the DSCCATDK command makes it DSCFIL. DSCMAI uses the name from the DSN= statement in the DSC MAIDATA file as the data set name to use when initializing the recovery file. If you use a name other than the default of CAI.ACF2.DSC.RECOVERY.FILE, you must modify the DSC MAIDATA file (on the CAIMAINT 291 disk) before running the above commands.

    If you are using the RULELONG feature to use CA ACF2 for z/VM rules over 4 Kb in length, you must modify the BLOCKSIZE=5120 statement in the DSC MAIDATA file. To modify the BLOCKSIZE, specify the VSAM CISIZE+256, up to a maximum of 32512. For example, if you are using a CISIZE of 8192 (8K), change the statement to read BLOCKSIZE=8448. The BLOCKSIZE= cannot be set to less than 5120 or more than 32512.

    The DSCGEN EXEC creates the Database Synchronization Component nucleus. This nucleus is created by a process similar to the CP or CMS nucleus. It punches a standard load deck back to the virtual reader. The DSGEN EXEC IPLs 00C to create the actual nucleus, assuming the load deck was created WITH RC=0.

To generate the nucleus, issue the following commands:

```
ACCESS 291 B
DSCGEN
```

Make sure that, when the DSCGEN finishes, it finishes with a disabled wait PSW with the second word as all zeroes. Also, there must not be any other error messages issued, such as unresolved references or text files not found, even if the PSW has a second word of zeroes.

You must IPL CMS (or log off and log back on) after the DSCGEN to get out of the disabled wait. Then, read in the LOADMAP with the following command:

```
DSCRDMAP
```

Using the Database Synchronization Component

There are two ways to pass commands to the Database Synchronization Component service machine, through the SMSG command or through the ACF2DSC PARMS parameter file. The next two sections provide additional information.

SMSG Command

The syntax of the SMSG DSC service machine command follows:

```
SMSG dscsm    DSC {OFF|ON|KILL}
              DSCOUT
              DSCOUT(ID=userid)
              DSCOUT(sysid)
              DSCTRACE(ON|OFF)
              ST
              STATS
              STATUS
              RESETSTATS
              SYSOUT {userid}
```

ACF2DSC PARMS Parameter Files

The following commands are transmitted to the Database Synchronization Component service machine through the ACF2DSC PARMS parameter file:

```
CCIVM(CCIVM|userid)
DSC {OFF|ON|KILL}
DSCADMIN(user1, user2, ..., user10)
DSCFILE(datasetname)
DSCLOCAL(sysid[(Jc)])
DSCNODES(sysid[([Jc][R])])
DSCOUT(ID=userid)
DSCTRACE(ON|OFF)
SYSOUT(userid)
```

Parameters

The following list contains descriptions of each parameter:

CCIVM

Identifies the virtual machine that runs CAICCI, the Common Communications Interface component of CA-CIS Services. The Database Synchronization Component uses CAICCI for communication requirements between nodes.

The entry method is the parameter file.

userid

The user ID of the virtual machine that runs the CAICCI component of CA-CIS Services.

DSC

Specifies whether the Database Synchronization Component is activated at startup. You must enter at least one of the DSC-related control options at startup to use the Database Synchronization Component. If you do not specify one control option, DSC might not be activated until the next startup and no DSC control options are honored until that time.

The entry method is SMSG or the parameter file.

OFF

This node does not transmit or receive records from other nodes.

ON

Loads DSC support modules into memory. Routing can start as soon as startup is complete.

KILL

The DSC subtask terminates and produces a dump of the virtual machine. Once the subtask is killed, you can reactive it using DSC(ON).

The following indicates that the user does not want to use the Database Synchronization Component.

```
SMSG dscsm DSC(OFF)
```

`dscsm-Specifies the ID of the Database Synchronization Component service machine.`

DSCADMIN

Identifies the virtual machines that are authorized to communicate with the Database Synchronization Component service machine through SMSG. You can specify up to ten virtual machine IDs.

The entry method is the parameter file.

DSCFILE

Identifies the OS data set name containing the DSC recovery file. The data set must reside on the DASD volume or minidisk at virtual address x'600' of the server machine.

The entry method is the parameter file.

datasetname

Contains the OS data set name of the DSC recovery file.

The following example indicates that the data set name for the DSC recovery file is CAI.ACF2.DSC.RECOVERY.FILE. Enter the following statement in the parameter file:

```
DSCFILE(CAI.ACF2.DSC.RECOVERY.FILE)
```

This file cannot be shared with another system.

DSCLOCAL

Identifies the CCI sysid for this system and whether requests received from other CA ACF2 for z/VM DSC systems are written to a journal file.

sysid

The CCI sysid value for this system.

(Jc)- J

Creates a journal file for DSC commands received from other DSC systems.

c

Indicates a specific output class for DSC journal files. The default is A.

The following example identifies that the CCI sysid for this system is VMSYSA and that all incoming DSC requests are written to a journal file with a default output class of A.

```
DSCLOCAL(VMSYSA(JA))
```

DSCNODES

Identifies the CCI sysids of the remote CA ACF2 for z/VM nodes from or to which DSC can propagate requests.

Note: When used in reference to the Database Synchronization Component, node refers to the unique identifier that is assigned to a node when it is defined using CAICCI. For more information on defining a node, refer to the CA-CIS Reference Guide.

The entry method is the parameter file.

sysid

Specifies the CCI sysid of the remote CA ACF2 for z/VM nodes from and to which database sync requests are transmitted. For example, the following identifies nodes ABC123 and ABC456 as targets of DSC requests:

```
DSCNODES(ABC123,ABC456)
```

(Jc,R) - (Jc)

Creates a DSC journal file containing all Database Synchronization Component activity sent to this DSC node. This file is optional. The journal file is spooled class A as a default. You can change this by adding a class ('c').

For example, the following enables both nodes for journal files with ABC123 journal file as class A and ABC456 journal file as class E:

```
DSCNODES(ABC123(J),ABC456(JE))
```

(R)

Specifies this node is receive only. You can receive synchronization requests from another system, but you cannot send requests. For example, to make node ABC456 be receive only:

```
DSCNODES(ABC123(J),ABC456(JE,R))
```

DSCOUT

Identifies the virtual machine that receives the DSC journal files. You can also use DSCOUT to close all or a selected DSC node journal file for output.

The entry method can be the parameter file. The syntax is:

```
DSCOUT(ID=userid)
```

Or the entry method can be SMSG, where the syntax is:

```
DSCOUT(sysid)
```

userid

The user ID of the virtual machine to receive the DSC journal files. The default is id=SYSTEM.

sysid

The CCI sysid of the node whose journal is closed.

Note: When specified without operands, all DSC files are closed.

Enter the following command to close all DSC journal files for output:

```
SMSG dscsm DSCOUT
```

In the above example, dscsm represents the Database Synchronization Component service machine ID.

DSCTRACE

This option is used as a debugging tool. DSCTRACE allocates an internal DSC trace table. Use it only at the request of CA Technical Support.

The entry method is SMSG or the parameter file.

ON-

Activates the trace.

OFF

Indicates that trace is inactive or is deactivated. This is the default.

ST

Performs both the STATUS and STATS functions described below.

STATS

Used mainly for techical support:

```
RPI=      233    VSD=      122    VSM=      122    VSU=        0
REC=        0    MRC=      111    MRP=      111
```

RPI

Count of sync requests from CA ACF2 for z/VM main processing. This count includes confirmations from CA ACF2 for z/VM main processing indicating that an incoming request was processed.

VSD

Count of sync requests sent out to target systems.

VSM

Count of acknowledgements received for outgoing requests.

VSU

Count of acknowledgements received for unknown requests.

REC

Count of sync requests currently stored in the recovery file.

MRC

Count of sync requests received from target systems.

MRP

Count of acknowledgements sent back for incoming requests.

The VSD, VSM, and REC counters reflect multiple counts if there are multiple nodes defined. In other words, when a sync request comes in from CA ACF2 for z/VM main processing, it is converted into a separate request for each node it is sent to. Therefore, the VSD, VSM, and REC counts are incremented for these separate requests for each node instead of just by 1.

STATUS

Returns the current status of the ACFSYNC machine. The values are very similar to the ACF2DSC PARMS file. Sample output follows.

```
DSC(ON)                   CCIVM(CCIVM)

DSCFILE 00% (CAI.ACF2.DSC.RECOVERY.FILE)

DSCADMIN(MAINT,OPERATOR,MIKE)

DSCOUT(SYSTEM)

DSCTRACE(INACTIVE)

DSCNODE(M002)     STATUS(ACTIVE,SPOOL,RETRY) JOURNAL(031,Y)

SYSOUT(SYSTEM)
```

The percentage full for the recovery file displays on the DSCFILE line and the status of a particular node displays on the DSCNODE lines.

RESETSTATS

Resets the statistical counters displayed with the ST and STATS command. The REC= counter is not reset because it indicates the number of transactions in the recovery file.

SYSOUT

Identifies the virtual machine that receives the server machine console log (which records operator communications, messages, and dumps).

The entry method for SYSOUT(userid) is SMSG or the parameter file.

userid

The user ID of the virtual machine that receives the console log and server dumps. The default is SYSTEM.

Note: If the SYSOUT command is issued through SMSG without the user ID, the current console log is closed.

Database Synchronization Component Processing

When an update to one of the CA ACF2 for z/VM databases occurs, the CA ACF2 for z/VM service machine generates and sends a database synchronization request to the Database Synchronization Component service machine. These synchronization requests are sent according to the selection criteria specified in the DBSYNC VMO record. There are four main selection options:

- LIDS
- INFO
- RULES
- LASTACC

Updates to logonid records are split between the LIDS criteria and the LASTACC criteria. The LIDS selection synchronizes updates to logonid records, except for updates to logonids that are due to a logon validation that just changes the last access information. The LIDS selection includes updates due to logon requests if the password field was changed.

The LASTACC selection includes updates to logonid records due to logon validations that just change the last access information unless the password field was changed. Also, the ONCEADAY selection limits the synchronization requests the LASTACC selection generates to just one synchronization request per logon per day. This lets your site eliminate the overhead of synchronization requests for last access information when a logonid has multiple logon validations on the same day, but can still see what day the logonid record was last used for a logon.

The Database Synchronization Component service machine records the synchronization request in its recovery file. The recovery file retains this request until the target machine accepts it. After recording the request in the recovery file, the synchronization request is sent to CCIVM. CCIVM then sends the request to the target system. If the VM system or the Database Synchronization Component service machine restart, the transactions that were not processed in the recovery file are processed when the restart is complete. The ACF2DSC PARMS file defines several options for the Database Synchronization Component service machine, including the nodes the synchronization requests are sent to. This file is located on the 100 disk. All synchronization request activity can be written to journal files to allow for problem solving if necessary.

Incoming database synchronization requests come in to the Database Synchronization Component service machine through CCIVM from the originating system. They are then sent on to the main CA ACF2 for z/VM components. CA ACF2 for z/VM then makes the appropriate updates to the databases as follows:

- Logonid database.

  Synchronization requests for the Logonid database are field level updates if the logonid record already exists in the receiving database. In other words, if the NAME field is changed on the sending system, only the NAME field (and the last update time stamps, and so on) is changed on the receiving system. If the logonid record does not exist, then the entire logonid record from the sending system is inserted into the receiving system database.

- Rules and Infostorage databases.

  Synchronization requests for these database records are full record replacements. The full record from the sending system is inserted or replaced into the receiving system database. When a synchronization request is received in this case, the time stamp (TOD format) of the synchronization request record is compared to the time stamp of the record in the database. If the time stamp of the synchronization request record is not higher, the synchronization request is rejected. This prevents a synchronization request that was delayed because a link or process was down from replacing a more recent update to the database. For this reason, it is critical that all systems that are involved in database synchronization have their clocks synchronized to the same TOD (time of day clock) value.

- Delete requests.

  If a synchronization request for a record to delete is received, then the time stamp of the delete (not the record) in the synchronization request is compared to the time stamp of the record in the database. If the time stamp of the delete is not higher, then the synchronization request is rejected.

Because there is a separate synchronization request generated for each change to a CA ACF2 for z/VM database record, subject to the selection options, be sure to take care when doing mass changes to the databases. A CHANGE LIKE or CHANGE IF can generate hundreds or thousands of synchronization requests. Larger sites should be aware of some of the CA ACF2 for z/VM settings, including the SYNCQLMT setting in the DBSYNC VMO record and the size of the recovery file. Also, the Database Synchronization Component service machine should be tuned so that it can get the resources it needs when it gets a large number of requests.

We strongly recommend that you limit changes to the databases that would generate a large number of requests to a relatively low activity time frame, such as a normal maintenance window. You can disable synchronization on each system, then perform the mass change on each individual system. After the changes are done, re-enable synchronization. CA ACF2 for z/VM provides the ACFSERVE DISABLE SYNC and ACFSERVE ENABLE SYNC commands for this purpose.

Complex Synchronizing Environments

In environments where there are more than a single VM system or more than a single z/OS system sharing the same database, you can use the examples below as a guide in setting up your DSC parameters.

- Sharing the database between two VM systems and synchronizing to one z/OS system.

  In both VM systems, the ACF2DSC PARMS file should specify the z/OS system. Choose one VM system as the primary system. In the z/OS system NODEDEF record, set up the primary VM system as a normal node, and set up the other VM system as a receive only node. This lets both VM systems send a synchronization request to the z/OS system, but the z/OS system only sends its synchronization requests to the primary VM. Since the two VM systems are sharing the database, the second VM system does not need synchronization requests from z/OS.

- Sharing the database between two z/OS systems and synchronizing to one VM system.

  In both z/OS systems, the NODEDEF records should specify the VM system. Choose one z/OS system as the primary system. In the ACF2DSC PARMS file on the VM system, set up the primary z/OS system as a normal node. Set up the other z/OS system as a receiving only node. This lets both z/OS systems send a synchronization request to the VM system. The VM system only sends its synchronization requests to the primary z/OS. Because the two z/OS systems are sharing the database, the second z/OS does not need synchronization requests from VM.

DIAG 'A0' Subcode '04': CA ACF2 for z/VM Password Validation

DIAG 'A0' subcode '04' is a security subfunction that interfaces between an application (such as DirMaint) and a security product (such as CA ACF2 for z/VM). Subcode '04', in particular, applies to password validation. CA ACF2 for z/VM's support of this diagnose subfunction lets users validate CA ACF2 for z/VM passwords from their own unique applications.

Note: Requirement for VMSAF Logonid Attribute:

The logonid issuing the DIAG 'A0' subcode '04' must have the VMSAF logonid attribute. For details about this logonid field, see the *Administrator Guide*.

The instruction format for this diagnose and subcode follows:

```
           LA    Rx,UIDPW      Point Rx to user ID/PW plist
           LA    Ry,4          Set Ry to subcode 4
           DC    X'83xy00A0'   Issue PW validation Diag
           BZ    OK            cc = 0 then goto OK
           C     Ry,=F'32'     Is security system not-aval
           BE    NOACF2        Yes - then goto NOACF2
           B     ERROR         Serious Error contact
*                              CA-Technical Support
           ...
UIDPW      DC    CL8'userid'
           DC    CL8'password'
```

The completion codes for DIAG 'A0' subcode '04' are:

```
CC0 --> Successful, Ry = 0
CC1 --> Unsuccessful, Ry = return code
        08 = bad password
        32 = Security system CA-ACF2 VM not active.  User application
             must determine whether to allow.  Not applicable if
             NOAUTO=DIRPASS in VMXAOPTS because the password
             is validated against the VM directory.
```

For any process issuing DIAG 'A0' subcode '04', complete the password validation as soon as possible. If a CA ACF2 for z/VM group virtual machine submits a request with the password of the group user and logs off before CA ACF2 for z/VM validates the password, CA ACF2 for z/VM validates the password against the group virtual machine, not the group user. Consequently, the validation fails.

Any application that issues DIAG 'A0' subcode '04' should issue them immediately after receiving a transaction instead of waiting for the transaction processing. For example, a server machine can receive a request for nighttime processing; CA ACF2 for z/VM should validate the password when it receives the request instead of at night.

CA ACF2 for z/VM supplies the ACFSAFA0 module that can be used by user applications to issue DIAG 'A0' subcode '04'. The syntax is as follows:

```
ACFSAFA0 userid password
```

DirMaint Consideration with ISF/CSE Complexes

CA ACF2 for z/VM password validation for DirMaint occurs through the DIAG 'A0' subcode '04' interface. The CA ACF2 for z/VM support for this interface incorporates special logic for CA ACF2 for z/VM group logon machines (defined with the GRPLOGON privilege) when you install the IBM Inter-System Facility (ISF) or when you are running in a CSE complex.

For example, assume DirMaint is running on one processor in an ISF/CSE complex. A user running on another processor can issue a DirMaint command that is sent to DirMaint. When DirMaint requires CA ACF2 for z/VM to validate the password, it uses a special check to determine whether the user is defined as a CA ACF2 for z/VM group machine. This check is needed because, if the user is a group machine, CA ACF2 for z/VM must validate the password against the group user of the machine instead of the group machine itself.

The GRP-USER field, defined in the Access section (Group 3) of LID records, contains the logonid of the last group user to log onto a group machine. The password of this logonid undergoes validation whenever a user invokes DIAG 'A0' subcode '04' to validate the password of a group machine in an ISF/CSE complex.

Support for DirMaint

CA ACF2 for z/VM support for DirMaint is implemented in a series of DirMaint exits. These exits then call our interface module, ACFVMDMI. ACFVMDMI performs various functions, including interfacing with the rest of the CA ACF2 for z/VM system to perform command limiting, logging support, and logging all DirMaint messages. There are basically three pieces to the CA ACF2 for z/VM support for DirMaint:

- Password validation-This is implemented through CA ACF2 for z/VM diagnose A0 support and through DVHDA0 that is supplied with DirMaint.
- Command Limiting-CA ACF2 for z/VM command limits all DirMaint commands the DirMaint service machine executes for users.
- Message logging-CA ACF2 for z/VM logs all DirMaint messages to SMF records. CA ACF2 for z/VM can also run reports on them.

You do not need to implement all of these three pieces. Password validation is a stand alone piece. Command limiting is also a stand alone piece, but is required if you are going to use DirMaint message logging.

Installing Password Validation support.

To enable the DirMaint ESM Password Authentication exit, tailor the CONFIG DATADVH file to include the following line:

```
ESM_PASSWORD_AUTHENTICATION_EXIT= DVHDA0 MODULE
```

This line probably exists with a / or // in front of it. If it does, simply remove the / or // to activate the exit.

Be sure the DirMaint machine logonid has the AUDIT, DG84DIR, and VMSAF privileges.

If you are installing DirMaint in a CSE/ISF complex, assign the DG84DIR logonid attribute to each of the satellite server machines.

Installing Command Limiting and Message Logging Support

To install command limiting and message logging support, use the following steps.

1. See the *Command and Diagnose Limiting Guide* for more information about the DirMaint command limiting feature.
2. CA ACF2 for z/VM supplies the command limiting models for the standard DirMaint commands in sample MODEL files. These files have a filename specific to your release of DirMaint, and a file type of MODEL.

   Compile the DirMaint command models for your release of DirMaint:

   ```
   DIRMR500-DirMaint Version 1 Release 5.0
   ```

   ```
   DIRML410-DirMaint Function Level 410
   ```

   ```
   DIRML510-DirMaint Function Level 510
   ```

3. Write and store command limiting rules for the DirMaint commands or review existing rules and modify them as necessary.

4. Be sure the DirMaint machine logonid has the AUDIT, DG84DIR, and VMSAF privileges.

5. If you are installing DirMaint in a CSE/ISF complex, assign the DG84DIR logonid attribute to each of the satellite server machines.

6. Ensure that the following eight files exist on a disk that DirMaint always accesses:

   ```
   ACFCKDMI EXEC
   ACFESMLR EXEC   (only used if message logging is enabled)
   ACFXDN   EXEC
   ACFXRA   EXEC
   ACFXRB   EXEC
   ACFXRC   EXEC
   ACFVMDMI MODULE
   ACFSRF   LOADLIB
   ```

   You can place these files on the DirMaint 191, or wherever you put local exits, if you have a special disk for local modifications.

7. Tailor the DirMaint CONFIG DATADVH file.

   To enable the DirMaint exits that CA ACF2 for z/VM uses, make sure that the CONFIG DATADVH file contains the following lines:

   ```
   DASD_OWNERSHIP_NOTIFICATION_EXIT=    ACFXDN   EXEC
   REQUEST_BEFORE_PARSING_EXIT=         ACFXRC   EXEC
   REQUEST_BEFORE_PROCESSING_EXIT=      ACFXRB   EXEC
   REQUEST_AFTER_PROCESSING_EXIT=       ACFXRA   EXEC
   ```

   To optionally include message logging, also add the following line:

   ```
   ESM_LOG_RECORDING_EXIT=              ACFESMLR EXEC
   ```

8. These lines probably exist in some form already, specifying DirMaint sample exits with a / or // to comment out the line. You can find these lines and change them as shown above, making sure to remove the / or //. Also, be sure to use the CA-ACF2 exit names.

   Add the following lines to the CONFIG DATADVH file:

   ```
   REQUIRED_SERV_FILE=     ACFCKDMI EXEC
   REQUIRED_SERV_FILE=     ACFESMLR EXEC
   REQUIRED_SERV_FILE=     ACFXDN   EXEC
   REQUIRED_SERV_FILE=     ACFXRA   EXEC
   REQUIRED_SERV_FILE=     ACFXRB   EXEC
   REQUIRED_SERV_FILE=     ACFXRC   EXEC
   REQUIRED_SERV_FILE=     ACFSRF   LOADLIB
   REQUIRED_SERV_FILE=     ACFVMDMI MODULE
   ```

   Tailor the DirMaint LCLASERV MSGADVH file.

Add the following lines to your LCLASERV MSGADVH file:

```
* Message 395E is for the CA-ACF2 VM DirMaint support
```

```
39510101E _1_ _2_ _3_ _4_ _5_ _6_ _7_ _8_ _9_
```

If you do not have a LCLASERV MSGADVH file, you can create one and place it on the same disk as your DirMaint 150ASERV MSGADVH file.

Be aware that CA ACF2 for z/VM messages issued from the DirMaint service machine have the following format:

```
ACFmod3951E errn message text
```

mod

The module issuing the message

errn

The error message number. You use this value to look up the message in the *Messages Guide*.

9. If you modified DirMaint to autolog the DATAMOVE machine (normally done by AUTOLOG1), make sure the DirMaint machine can autolog the DATAMOVE machine. To do this, write a resource rule that lets DirMaint autolog the DATAMOVE machine.

   ```
   $key(datamove) type(alg)
   uid(dirmaint) allow
   ```

   This example assumes that the default resource type for AUTOLOG commands is ALG, as set by the RESCLASS VMO record.

   You might also consider assigning the AUTONOPW attribute to the DATAMOVE machine. This lets DirMaint autolog DATAMOVE without requiring a password.

10. IPL the DirMaint virtual machine.

National Language Support

CA ACF2 for z/VM supports any site-generated language for IBM national language support. You can generate a language through the CA ACF2 for z/VM full-screen feature or through the CA ACF2 for z/VM help facility. You can also display CA ACF2 for z/VM messages in the generated language.

We provide two languages: AMENG (American English, in mixed case), and UCENG (Upper Case English). AMENG is the CA ACF2 for z/VM base language. It is the default when there is no site-defined language. We also use the AMENG when a CA ACF2 for z/VM response does not exist in a site's defined language. We provide UCENG as an alternative to the CA ACF2 for z/VM base language.

Each user can set his own language (AMENG, UCENG, or one that is site-defined) through the SET LANG command or DIAG x'C8'. You can also automatically set the language for each user through the OPTIONS statement in the VM directory.

Defining Language Messages

The next two sections contain information on defining messages for a national language. You should also be aware that CA ACF2 for z/VM provides the following default warning message:

```
After July 1, 1999 this access will not be allowed
```

To change the text of this message, you must change the MSG operand of the WARN VMO record. For information on changing this VMO record, see the *Administrator Guide*.

CA ACF2 for z/VM requires you define messages for a national language in separate modules. HCPAF0 is a required language module. As shipped, this module contains American English (AMENG). HCPAL0, another CA ACF2 for z/VM language module, contains Upper Case English (UCENG). We provide nine separate dummy modules for your use, HCPAL1 through HCPAL9. Use these modules to define a site-generated language for CA ACF2 for z/VM.

You must define any additional national languages through the LANG operand of the VMXAOPTS macro in HCPAC0. This operand specifies each national language module and the text file where CA ACF2 for z/VM defines the message text for the language. The *System Programmer's Guide* contains complete information about defining your national languages through the LANG operand.

CA ACF2 for z/VM codes the language modules using the ACFCPMST, ACFMDEST, and ACFMSGEN macros. The Altering Message Attributes Through ACFCPMST and Generating the Message Table through ACFMSGEN sections explain each of these macros in detail.

Altering Message Attributes through ACFCPMST

The ACFCPMST macro defines each message. It always generates the text of a message. CA ACF2 for z/VM base language module (HCPAF0) only generates the attributes of a message. If an alternate language module defines attributes, CA ACF2 for z/VM validates but does not use them.

You can customize CA ACF2 for z/VM messages to meet your site's needs. For example, you can modify the messages that CA ACF2 for z/VM normally sends to the operator so that they are sent elsewhere. (This is an alternative to setting up the PROP message routing tables for CA ACF2 for z/VM messages.) Failure to modify the routing of messages when you are running PROP can result in performance problems.

You can modify a message's ACFCPMST macro to alter the text and attributes of any CA ACF2 for z/VM message.

```
ACFCPMST 242W,'ACFFDR mode is invalid - mode set to abort',  X
      HILIGHT=YES,                                           X
      USER=NO,                                               X
      OPER=YES
```

The attributes specified in this macro indicate that CA ACF2 for z/VM highlights the message at the system operator's terminal.

You can customize this message for different results. The following changes send the message to a list of logonids in charge of implementing security (as opposed to the operator). We have also modified the message text.

```
ACFCPMST 242W,'ACFFDR mode is invalid - mode set to abort - sX
      ecurity team take action if necessary',                X
      HILIGHT=YES,                                           X
      USER=NO,                                               X
      DEST=secteam
```

The ACFAMENG copybook defines American English messages. There must be an entry for ACFAMENG in the ACF2USER EXEC. The ACFUCENG copybook defines Upper Case English messages. The ACFAMENG copybook contains the above message. We ship HCPAF0 using the ACFAMENG copybook and HCPAL0 using the ACFUCENG copy book.

You can use the following operands to customize CA ACF2 for z/VM messages.

ACF2SM=YES|NO

Specifies whether CA ACF2 for z/VM sends the message to the service machine console. NO is the default. CA ACF2 for z/VM only accepts YES if a CP module or the service machine issued the message.

ALARM=YES|NO

Specifies whether an audible alarm sounds at the destination. NO is the default. YES has no effect if CA ACF2 for z/VM returns the message in a buffer for a module running in CMS or GCS. This includes messages the ACF command, CA ACF2 for z/VM reports, and SRF issue. Messages that programs running in CMS issue respect the YES setting only if you set CMS FULLSCREEN on. It does not respect YES in line mode (a limitation of the CMS TERMWRT macro).

COMPRES=YES|NO

Specifies whether CA ACF2 for z/VM compresses multiple contiguous blanks into a single blank and removes all trailing blanks before display. YES is the default.

DEST=label

Defines a ACFMDEST macro label that defines a list of logonids where CA ACF2 for z/VM sends the message. Use DEST with the USER and OPER attributes. If the defined user is disconnected, CA ACF2 for z/VM does not display the message unless you use IUCV MSG. There is no default value. CA ACF2 for z/VM only respects this attribute if a CP module or the service machine issued it.

DISPLAY=EMSG|IMSG|FULL|NOID

Specifies how CA ACF2 for z/VM sends the message. EMSG indicates that the message is a CP EMSG (respects all EMSG settings). IMSG indicates that the message is a CP IMSG (respects all IMSG settings). FULL indicates that CA ACF2 for z/VM always displays the message ID and text, regardless of EMSG and IMSG settings. NOID indicates that CA ACF2 for z/VM displays only the message text, regardless of EMSG and IMSG settings. The default is EMSG.

If CA ACF2 for z/VM returns the message in a buffer for a module running in CMS or GCS (including command modules, reports, and SRF), it respects the user's EMSG setting unless you specify NOID, displaying the message text. Message programs running in CMS respect the NOID and FULL operands, otherwise CA ACF2 for z/VM outputs the message according to the EMSG setting.

HILIGHT=YES|NO

Specifies whether CA ACF2 for z/VM highlights the message at the destination terminal. NO is the default. YES has no effect if CA ACF2 for z/VM returns the message in a buffer for a module running in CMS or GCS. This includes messages the ACF command, CA ACF2 for z/VM reports, and SRF issue. Message programs running in CMS respect the YES setting only if you set CMS FULLSCREEN on. CA ACF2 for z/VM does not respect YES in line mode (a limitation of the CMS TERMWRT macro).

OPER=YES|NO

Specifies whether CA ACF2 for z/VM routes the message to the operator's console. NO is the default. CA ACF2 for z/VM respects YES only if a CP module or the service machine issued the message.

PRIOR=YES|NO

Specifies whether the message is a priority warning, immediately interrupting the user's virtual machine. NO is the default. YES has no effect if CA ACF2 for z/VM returns the message in a buffer for a module running in CMS or GCS. This includes messages the ACF command, CA ACF2 for z/VM reports, and SRF issue.

PRND=YES|NO

Sets the terminal input field to nondisplay. Used for password prompt messages, the YES setting prevents the display of any of the password between the time CA ACF2 for z/VM issues this message and the time that CA ACF2 for z/VM reads the password.

USER=YES|NO

Specifies whether CA ACF2 for z/VM routes the message to the user's terminal. YES is the default. CA ACF2 for z/VM respects NO only if a CP module or the service machine issued the message.

The following variables indicate substitutions in the message text:

&x1,...,&x9

CA ACF2 for z/VM substitutes data from the message request for the variable. The number denotes the relative position in the message request substitution list of the data that replaces the variable. *x* specifies the format of the data. CA ACF2 for z/VM uses this information to convert data to a readable format in the message. Valid values for x are:

&C-

Character format

&D

Decimal data (binary converted to decimal)

&H

Hexadecimal data

&T

Character standard CMS tokenized data in character format.

Below are examples of using these variables in messages:

Invalid data &C1 in field &C2 in record &D3

Record &D3, field &C2, has invalid data &C1

File &T1 not found, FSOPEN RC=&D2

Program &C1 loaded at address &H2

&DATE

The current date

&LID

The VM user ID of the logged on user

&SMFID

The SMFID parameter value of the ACFFDR @SMF macro

&TIME

The current time

&UID

The UID of the logged on user (CA ACF2 for z/VM does not substitute this variable during a NOAUTO IPL)

&USERID

The VM user ID of the logged on user.

You must enter all the above variables in upper case, as shown. A message without any ACFCPMST macro operands assumes the defaults.

We recommend the following step-by-step procedure for altering message text:

1. Use XEDIT with the CTL option to update the copy book containing the ACFCPMST macros for your messages (for example:

   ```
   XEDIT ACFAMENG COPY (CTL ACF2)
   ```

   ACFAMENG is the copybook containing CA ACF2 for z/VM messages in mixed case American English. HCPAF0 uses this copybook. Alter the ACFCPMST macro attributes for the messages you want to change.

2. Use XEDIT to insert the following line into the ACF2USER EXEC file:

   ```
   &1 &2 &3 ACFAMENG COPY *
   ```

3. Use VMFMAC to rebuild the ACF2USER maclib as follows:

   ```
   VMFMAC ACF2USER ACF2
   ```

4. Copy ACF2USER MACLIB to the local options disk (2A0) and erase it from the A-disk.
5. Reassemble the HCPAF0 message module.

   ```
   VMFHASM HCPAF0 ppfname component
   ```

   ppfname

   The name of your PPF file used for genning CP

   component

   The component ID, normally CPTEST.

   Note: If you use the high level assembler, then use VMFHLASM instead of VMFHASM.

6. Regenerate CP.

Defining User IDs through ACFMDEST

The ACFMDEST macro defines a list of user IDs. One or more ACFCPMST macros can specify the label of an ACFMDEST macro to route messages to users defined in the list. This macro is valid only in the CA ACF2 for z/VM base language module, HCPAF0.

Generating the Message Table through ACFMSGEN

The ACFMSGEN macro generates a message reference table and defines the substitution characters CA ACF2 for z/VM uses in the message defined in this module. The default is an ampersand (&).

The ACFNLANG macro defined in the base language module references the entry point for this table. This must be the last macro defined in the message module.

Implementation

To implement national language support:

1. Create the site-generated text file for the language.

   - Define the language messages
   - Assemble the alternate language message module.

   ```
   VMFHASM HCPALn ppfname component
   ```

   HCPALn

   The name of your alternate language file, normally HCPAL1 through HCPAL9

   ppfname

   The name of your PPF file used for genning CP

   component

   The component ID, normally CPTEST.

   Note: If you use the high level assembler, then use VMFHLASM instead of VMFHASM.

2. Review the ACFCP CAXALOAD control file. To support an alternate language, specify a CPPAG entry for the alternate language module, defined in the LANG operand. See Defining Language Messages for dummy language module names.

3. Modify the LANG operand of the VMXAOPTS macro in HCPAC0. See the *System Programmer's Guide* for complete instructions.

Considerations

Keep in mind the following information about national language support with CA ACF2 for z/VM:

- The CA ACF2 for z/VM execs are shipped in mixed case. To convert your execs to uppercase (if, for instance, you have uppercase only terminals), run ACFUPCAS EXEC. This exec converts your CMS and EXEC messages to uppercase until you reIPL CMS.
- If the LANG operand of VMXAOPTS does not define a language set for a user or if the corresponding language module is unresolved when you generate CP, CA ACF2 for z/VM uses the CA ACF2 for z/VM base language.
- CA ACF2 for z/VM uses the base language for any message not defined in the language module. This means that a site-written language module does not need to include all messages. For example, you could include only the most common messages, leaving CA ACF2 for z/VM base language to issue the least common messages.
- The CA ACF2 for z/VM base language does not have to be the same as the system default language. You can define the IBM default language as a CA ACF2 for z/VM alternate language. CA ACF2 for z/VM always uses the IBM CP logic language.
- During system access through LOGON and DIAL, CA ACF2 for z/VM uses the IBM system default language because IBM sets the VMDBK during logon to the system default language. CA ACF2 for z/VM does not use the directory specified language to reset the VMDBK language pointer until logon processing verifies the user is allowed to log on.

PAM Server Support

This section describes how to install and configure the PAM Server on a VM system. For information on the PAM Server see the CA PAM Client for Linux for System z *Product Guide*.

CA-ESM release 1.1 or above, a component of CA-CIS, needs to be installed before attempting to use the PAM server.

Configuring the PAM Server

Configuring the PAM Server on a VM system requires setting up the PAM service machine and the setting of some options.

Step 1: Create the PAM Server Service Machine

The PAM Server needs to run in a service machine, normally called PAMSERVE.

Create a VM directory entry for the PAM Server service machine. The PAM Server service machine needs to have at least a 1 cylinder 191 minidisk or equivalent SFS space. We supply a sample directory entry in file PAMSERVE DIRECT on the CAIMAINT 291 minidisk.

```
USER PAMSERVE PAMDPSWD 24M 48M G
   IPL CMS PARM AUTOCR
   IUCV *RPI
   MACHINE ESA
   OPTION ACCT MAXCONN 00032 QUICKDSP
   CON 0009 3215 T
   SPOOL 000C 2540 READER A
   SPOOL 000D 2540 PUNCH A
   SPOOL 000E 1403 A
   LINK MAINT    0190 0190 RR
   LINK TCPMAINT 0198 0198 RR
   LINK MAINT    019D 019D RR
   LINK MAINT    019E 019E RR
   LINK TCPMAINT 0591 0591 RR
   LINK TCPMAINT 0592 0592 RR
   MDISK 0191 3390 xxxx 1 vvvvvv MR RPW WPW MPW
```

Create a standard TCP/IP PROFILE EXEC similar to your other TCP/IP service machines, and place it on the PAMSERVE 191 minidisk. We supply a sample profile in file PAMSERVE PROFILE on the CAIMAINT 291 minidisk:

```
/* PAM Server service machine PROFILE EXEC */
'Access 198 D'
'Access 591 E'
'Access 592 F'
queue “EXEC TCPRUN”
```

Step 2: Add the CA PAM Server to SYSTEM DTCPARMS

Define the CA PAM Server as a TCP/IP service machine by adding an entry in your SYSTEM DTCPARMS file, normally on the TCPMAINT 198 minidisk. We supply the following sample entry in file PAMSERVE DTCPARMS on the CAIMAINT 291 minidisk:

```
.* PAM server (PAM) daemon
:nick.PAMSERVE  :type.server  :class.pam
:nick.pam     :type.class
              :name.PAM daemon
              :command.SRVRPAM
              :runtime.C
              :diskwarn.YES
              :anonymous.NO
              :ESM_Enable.Yes
              :ESM_Validate.ACFSAFA0
              :ESM_Racroute.RPIUCMS
              :VMLINK.CAIMAINT 291 (NONAMES
```

Step 3: Modify the TCPIP Configuration File

Modify your TCPIP Configuration File, normally on the TCPMAINT 198 minidisk, as follows:

1. Add an entry in the AUTOLOG section for the PAM server:

   ```
   PAMSERVE password    ; PAM Server
   ```

2. Add an entry in the PORT section for the port to be used by the PAM server. For example, to use port 1091:

   ```
   1091 TCP PAMSERVE            ; PAM Server
   ```

Step 4: Create the PAM Server Configuration File

The PAM Server needs a configuration file named PAMD CONF on the PAMSERVE 191 if a CMS file is used, or pamd.conf if a BFS file is used. This section describes the options that can be specified. You need to create this file with at least the userid statement.

- You can define the options in any order. If you specify an option more than once, the last value is taken.
- None of the keywords are case sensitive, but the values are. So make sure that you enter things like file names, including the directory portion, in the correct case for the values.
- If an option has a default value, it is documented. If the default value is the desired value, you do not need to specify that option in the configuration file.

The following options can be specified in the pamd.conf configuration file:

threads

Specifies the maximum number of threads the PAM Server can start. The default is 32.

userid

Specifies how to handle the mapping of the Linux for zSeries user ID to VM security. Valid values are:

LINUX

Requires that there is a user map record to convert the Linux for zSeries name to a VM security id. If the mapping record does not exist, the logon fails. The default is LINUX.

MVS

Bypasses the user map record and tries to validate the user ID passed directly to the VM security product if the user ID is less than 8 bytes. If the user ID is greater than 7 bytes, the logon fails.

MIXED

Maps the Linux for zSeries user ID to a VM security ID. If the mapping exists, that user ID is used for validation. If the mapping does not exist and the user ID is 8 bytes or less, it will try to perform the validation using the passed in user ID.

port

(optional) Specifies the port number to which clients must send requests to connect to this server. Number must be a positive integer number in the range of 1 to 65535.

Note: There is no default for this option. Therefore, if this configuration file option is not specified, then the -p command line option must be. For example:

```
port number
```

requests

(optional) Specifies the maximum number of requests that the server can have pending. Number must be a non-negative integer. If number is zero, then there is no limit to the number of pending requests. If this option is not specified, the maximum number of pending requests is "0". For example:

```
requests number
```

linuxdata

Specifies which segment is used to extract UID, GID, Home and Shell values. This operand applies only when the PAM r8 client is being used. If the PAM r12 (or above) client is used, the Linux segment is always used

LINUX

This configures the PAM Server to extract the data from the users Linux segment. This is the default.

OMVS

This configures the PAM Server to extract the data from the users OMVS segment.

LowerCase

(optional) Controls what values the server forces to lower case before returning to the Linux for zSeries system. The allowed values of option are:

username

Server will force the mainframe security id to lower case. Note that a Linux for zSeries name is always returned unchanges. This option applies only to a mainframe security id.

groupname

Server will force the mainframe security group name to lower case.

homedir

Server will force the Linux for zSeries home directory name to lower case.

al

Equivalent to specifying all of the above values.

You may specify multiple values on the command, separated by blanks.

GentleHup

(optional) Controls the response of the server to a hangup signal. The allowed values of option are:

ON

Server will stop listening for new connections. However, the server will continue to accept requests from the connections to current clients. The server will terminate when all clients have closed their connections

OFF

Server will stop listening for new connections. The server will wait for all pending requests to finish and will then close the connections to current clients. The server will then terminate

GentleStop

(optional) Controls the response of the server to an operator STOP command. The allowed values of option are:

ON

Server will stop listening for new connections. However, the server will continue to accept requests from the connections to current clients. The server will terminate when all clients have closed their connections

OFF

Server will stop listening for new connections. The server will wait for all pending requests to finish and will then close the connections to current clients. The server will then terminate

host

Specifies the address of the interface over which the server is to accept connections. This value is optional.

```
host network-address
```

Where *network-address* specifies a domain name or an IP address in dotted decimal notation. If a domain name is specified, the server will convert it to an IP address.

If this option is specified, the server will only accept connection requests from the interface address specified. If this option is omitted, then the server will accept connection requests from all interface addresses configured for this host.

debug

(optional) Specifies the amount of information that the server should write to the stderr file. The value is set to the bit-wise OR of all of the arguments on the configuration line. Each number is a decimal integer value. The value is taken as a bit string, with each bit corresponding to a different kind of trace information. Available log levels are listed in the following table. There is no debug level by default.

Value	Debug Information
1	General trace information
2	Trace packets that are read or written to a TCP socket
4	Trace arguments to selected functions
8	Trace connection management
16	Not used
32	Not used
64	Configuration file processing
65535	All tracing

TLSRandomFile

Specifies the file from which the server obtains the initial seed for the pseudo-random number generator (PNG). The server updates this file each time the server starts so that the starting value of the PNG changes each time the server is run.

TLSCertificateFile

Specifies the path and name of the server's certificate. This certificate must be in PEM format. The server sends this certificate to a client so that the client can validate the server.

This option is required to use TSL or SSL for communications with clients.

Note: If you do not specify a server certificate and the associated private key, no SSL conversation can be started. If SSL_Required is specified on the Linux client, no communications will ever be started.

TLSCertificateKeyFile

Specifies the path and name of a file that contains the secret private key that matches the certificate stored in the TLSCertificateFile file. This file must be in PEM format. If this file is password protected, the server prompts for the password at the time that the server reads the configuration file.

Except as noted below, this option is required to use TLS or SSL for communications with clients.

If the server certificate and the associated private key are stored in the same file, this option can be omitted.

Note: Since you run the server as a disconnected service machine, prompting for a password is not possible. In this case, you should ensure that the private key is not password protected.

TLSCACertificateFile

Specifies the path and name of a file that contains the certificate for all Certificate Authorities that can sign a client certificate. Each certificate must be in PEM format.

If you have a single CA certificate that is used to sign all client certificates, just specify this file with this keyword. If you have more than one CA certificate, concatenate them together into a composite file and specify the path and name of this composite file with this keyword.

TLSVerifyClient

Specifies whether a client is required to present a certificate when attempting to establish a SSL or TLS connection with the server. Valid values are:

NEVER

The server does not request a certificate. This is the default.

Note: The values OFF, NO, or FALSE are accepted and are equivalent to NEVER.

ALLOW

The server requests a certificate. If no certificate is provided, the session proceeds normally. If a bad certificate is provided, it is ignored and the session proceeds normally.

TRY

The server requests a certificate. If no certificate is provided, the session proceeds normally. If a bad certificate is provided, the session terminates immediately.

DEMAND

The server requests a certificate. If no certificate is provided, or a bad certificate is provided, the session terminates immediately.

Note: The values HARD, ON, YES, and TRUE are accepted and are equivalent to DEMAND.

Step 5: Create the SRVRPAM EXEC

Create the SRVRPAM EXEC on the PAM Server 191 minidisk. This exec contains the OPENVM RUN command that starts the PAM server module LXPAMD.

There are two required parameters and one optional parameter used by LXPAMD:

```
LXPAMD -f config_file -p port [-d debug_level]'
```

where:

-f config_file

Specifies the configuration file to use for startup. This file can be a variable length record CMS file on the 191 minidisk, or it can be a BFS file. The file name can be anything you set up.

The suggested value if you are using a BFS file is:

```
-f pamd.conf
```

For a CMS file on the 191 minidisk:

```
-f “//PAMD CONF”
```

-p port

Specifies the startup TCP/IP port that it is running with.

-d debug_level

Specifies the level of debug and tracing messages to generate. The value can be from 0 to 65535. The default value is 0.

For example, to start PAM using the CMS file PAMD CONF and port 1091, create a SRVRPAM EXEC with the following lines: (We supply this sample in file PAMSERVE SRVRPAM on the CAIMAINT 291 minidisk.)

```
/* SRVRPAM EXEC - Start the PAM Server */
'OPENVM RUN LXPAMD -f “//PAMD CONF” -p 1091'
```

If any BFS files are used by the PAM server, an OPENVM MOUNT command for the BFS root is required, as well as an OPENVM SET DIRECTORY command to set the current directory.

For example, to start PAM using port 1091 and the BFS file /usr/lpp/capam/pamd.conf, create a SRVRPAM EXEC with the following lines:

```
/* SRVRPAM EXEC - Start the PAM Server */
'OPENVM MOUNT /../VMBFS:VMSYS:ROOT/ /'
'OPENVM SET DIRECTORY /usr/lpp/capam'
'OPENVM RUN LXPAMD -f pamd.conf -p 1091'
```

Step 6: Define the PAM Server ID to Security

Use the following commands to define the PAM Server user ID and started task information in the CA ACF2 for z/VM database:

```
ACF

SET LID

INSERT PAMSERVE VMESM SECURITY SYNERR(ALLOW) PASSWORD NOPSWD-EXP

SET PROFILE(USER) DIV(OEVM)

INSERT PAMSERVE UID(0)

END
```

Use the following command to rebuild the OpenExtensions tables:

```
ACFSERVE RELOAD PROFILE USER
```

Configuring CA ACF2 for z/VM Security for Use With the PAM Server

Now that the PAM Server has been installed and configured, you must setup CA ACF2 for z/VM to support the PAM Server.

The following sections detail the steps required to setup CA ACF2 for z/VM to support the PAM Server.

Step 1: Defining the Linux for zSeries Hosts

Using VMO LINUX records, the first step in enabling the PAM Server is defining all Linux for zSeries hosts and their TCP/IP addresses that are using this security database. Without these definitions, all logons fail.

```
SET CONTROL(VMO)
INSERT LINUX.qual MACHNAME(the linux machine name) IPADDR(the.ip.addr.here)
```

Repeat inserts

```
ACFSERVE RELOAD CONTROL VMO LINUX
```

Note: The MACHNAME value can be determined by issuing 'hostname -f' on the Linux system. The IP can then be determined by PINGing the hostname value from the z/VM system.

Step 2: Defining the Linux for zSeries Group Profile Records

Before you assign the Linux group name to a users profile record, you should define it and the GID that is associated with it. During RACROUTE EXTRACT processing, if the GID cannot be found for a group name, processing will fail and the user will not be allowed on to the Linux system.

```
SET PROF(GROUP) DIV(LINUX)
INSERT grpname LINUXGID(gid#)
```

Repeat the inserts for each group that needs to be defined, then issue the following to enable the changes:

```
ACFSERVE RELOAD PROFILE LINUX GROUP
```

Note: When defining groups, you should ensure that GID values are unique and that one group is not assigned the same value as another.

Step 3: Defining the Linux for zSeries User Profile Records

When signing on to a Linux for Series system, the user can use their existing 8-byte (short) CA ACF2 for z/VM user id, or if it is desired to allow the user to log on with a long name (greater than 8-byte), it can be mapped to their existing CA ACF2 for z/VM id.

The PAM Server installs by default with MIXED mode mapping enabled. Mixed mode configures the PAM Server to map the logon id to a short name before issuing the RACROUTE VERIFY call. If the mapping is successful, then the RACROUTE VERIFY is issued. If the mapping failed, but the length of the id is 8-bytes or less, it will issue the RACROUTE VERIFY with the original value.

If you want to define a user profile record that supports id mapping, issue the following.

```
SET PROF(USER) DIV(LINUX)
```

```
INSERT lid_here LINUXNAM(the linux long name) LINUXUID(uid#) LINUXGRP(grpname)
LINUXHOM(home.dir) LINUXPGM(shell_name)
```

If id mapping is not required, omit the LINUXNAM parm.

For the LINUXGRP parameter, specify a group name that matches one of the Linux group profiles you have defined.

Repeat inserts for each user that needs to be configured, then issue the following to enable the changes:

```
ACFSERVE RELOAD PROFILE LINUX USER
```

Note: If id mapping will never be used, you can disable the mapping call prior to the VERIFY. Refer to the userid parameter in the Installing and Configuring the PAM Server chapter.

Tip: When defining users, you need to ensure that UID values are not assigned to more than one user.

Step 4: Defining the Linux for zSeries Resource Rules

Using the resource rule type of LNX, you now define who can access which Linux for zSeries system.

```
SET RESOURCE(LNX)
COMPILE
$KEY(linux_Machine_name) TYPE(LNX)
 UID(string) ALLOW
```

Step 5: Define Secondary Group Membership

The resource rule type LGR defines who is a member of which group. Linux makes a distinction between a user's primary and secondary groups. You define the user's primary group when you define the Linux User Profile record for the user (the LINUXGRP option). Secondary groups are defined here.

Note: Skip this step if no user is a member of a secondary group.

To define who is a member of which group

1. Issue the following commands:

   ```
   SET RESOURCE(LGR)
   COMPILE
   . $KEY(groupname) TYPE(LGR)
   . UID(uidmask) ALLOW
   . additional resource rule entries
   .
   END
   STORE
   ```

   Repeat the COMPILE-END-STORE sequence for each group that needs to be defined.

2. Issue the following to enable the changes:

   ```
   ACFSERVE RELOAD RESOURCE LGR
   ```

   Include one resource rule entry for each user that is a member of this group. Do not list users who have this group as a primary group; only list users that have this group as a secondary group.

Note: Because CA ACF2 for z/VM implements secondary group membership through resource rules, any user who has the SECURITY attribute becomes a member of any group configured here whether or not they are listed.

Console Interface

The CA PAM Server recognizes the following commands. These commands are issued on the virtual console of the PAM server. They may also be issued as secondary console commands with the CP SEND command from the secondary console user.

Stop Command

This command causes the PAM Server to terminate.

REINIT Command

This command causes the server to reread the configuration file. If pathname is specified, it is the name of a new configuration file. It replaces the last configuration file specified, on the command line with the -f command line option or on a previous MODIFY command. When rereading the configuration file, not all configuration options are applied; some are ignored. The options from the configuration file that are applied are as follows:

DEBUG

Specified debug level replaces the existing debug level. The new value controls what trace information is subsequently written to the trace file.

LOGFILE

Server closes the current trace file and begins writing trace records to the file specified on the LOGFILE option.

LOGLEVEL

Specified syslog level replaces the existing syslog level. The new value controls what trace information is subsequently written to the syslog file

REQUESTS

Requests value specified replaces the existing value.

THREADS

Specified thread values replaces the existing threads value. If the new value is less than the old value, and the number of started worker threads exceeds the new threads value, the server will terminate enough worker threads to bring the total number of started threads down to the threads value.

GENTLEHUP

Requests value specified replaces the existing value.

GENTLESTOP

Requests value specified replaces the existing value.

LOGOFFAFTER

Requests value specified replaces the existing value.

ENABLEVERIFY

Requests value specified replaces the existing value.

All other configuration file options are checked for syntax but ignored.

Note: If pathname is specified, it cannot name an HFS file whose name has both upper and lowercase characters. This limit is imposed because the operator interface translates all characters on the MODIFY command to uppercase before passing the data to the application. The server will attempt to open the file using the name passed. If that fails, the server will convert pathname to lowercase and try again.

STATUS Command

This command causes the server to display the current options in effect to the console. The options displayed are:

Security product, debug level, syslog level, log file name, maximum threads, maximum requests, host address, port, gentlehup, gentlestop, identity mapping, segment being used and max password length.

To display only a single value, issue the following:

```
STATUS,option
```

where option can be:

```
PROD, DEBUG, LOGLEVEL, LOGFILE, THREADS, REQUESTS, HOST, PORT, GENTLEHUP, GENTLESTOP,
MAPPING, SEGMENT, MAXPSWD
```

SET Command

SET,DEBUG,value

Causes the server to change the current debug level in effect to the new value. This change takes effect immediately without having to stop and restart the server.

SET,LOGLEVEL,value

Causes the server to change the current syslog level in effect to the new value. This change takes effect immediately without having to stop and restart the server.

SET,MAXPSWD,value

Causes the server to change the maximum password length that it supports to the new value. This change takes effect immediately without having to stop and restart the server. The maximum is 200.

SET,SEGMENT,value

Causes the server to change the segment to extract the user's Linux information from. The valid values are Linux and OMVS. The default is OMVS. This change takes effect immediately without having to stop and restart the server.

PASSTHRU Release 1.3 Support

CA ACF2 for z/VM support for the PASSTHRU program product consists of:

CA ACF2 for z/VM Logical Device Interface Validation

When a CA ACF2 for z/VM secured PVM MODULE tries to create a logical device, it validates if the requestor is authorized to use the Logical Device Interface. Assign the LDEV logonid attribute to any logonids that are authorized to create logical devices through the CA ACF2 for z/VM secured PVM MODULE.

CA ACF2 for z/VM Logical Device Resolution

If a real device on the local PASSTHRU node initiates a PASSTHRU session, the logical devices PASSTHRU created for that session are resolved back to the real device. You can use this feature to create CA ACF2 for z/VM infostorage source entry records to protect through physical input sources rather than through logical devices. You can implement the optional unresolved Logical Device Creation Exit (LDEVXIT) to resolve logical devices created for sessions originating from remote PASSTHRU nodes to enforce additional controls.

CA-ACF2 PASSTHRU Release 1.3 support includes the DVMAPP JACF01DM and DVMAPP AUXACF3 CMS files.

To install this support:

1. Copy the CA ACF2 for z/VM PASSTHRU Release 1.3 support files (listed above) to the minidisk containing the PASSTHRU update files and AUX files.
2. Add the entry ACF AUXACF3 in your PVM CNTRL file.
3. Enter the following command to assemble DVMAPP with the updated control file:

   ```
   VMFASM DVMAPP PVM
   ```

   This creates a text deck called DVMAPP TXTACF that should include the JACF01DM update.
4. Copy DVMAPP TXTACF to the minidisk containing the PASSTHRU text files.
5. Enter the following command to execute the PVMBLD EXEC to create a new PVM MODULE:

   ```
   PVMBLD {loadlist-exec cntlfn}
   ```

   The PVMBLD EXEC uses the default control file PVM CNTRL. If you use some other control file to apply the DVMAPP update, you must specify the PVM LOADLIST EXEC with this control filename when you invoke PVMBLD.

6. Examine the PVM PRELMAP file PVMBLD created to ensure that it included the DVMAPP TXTACF file.
7. Copy the rebuilt PVM MODULE to the PASSTHRU system disk. You can copy the PVM MAP and PVM PRELMAP files to the same disk.
8. Assign the LDEF logonid attribute to the PVM logonid as follows:

   ```
   ACF
   ```

   ```
   set lid
   ```

   ```
   LID
   ```

   ```
   change pvm ldev
   ```

9. Start PASSTHRU and test.

RSCS System Component

CA ACF2 for z/VM RSCS support lets a job you submit to JES2 or JES3 over the NJI line driver inherit the logonid of the submitter. This means that you no longer need to put logonid statements in your z/OS job streams for them to run under your logonid. CA ACF2 for z/VM RSCS support works with any level system, although it requires CA ACF2 for z/VM for z/OS to take advantage of the logonid inheritance feature.

The CA ACF2 for z/VM RSCS Version 3 Release 1 and later support consists of the DMTAC2V3 TEXT text file. To install the CA ACF2 for z/VM RSCS Version 3 Release 1 and later support:

1. If you have previously installed CA ACF2 for z/VM RSCS Version 2 Release 1 support, remove the existing CA ACF2 for z/VM update to DMTNTR.
2. Modify the RSCS CONFIG file to call CA ACF2 for z/VM when an exit 11 condition occurs. The basic format of the EXIT statement in the CONFIG file follows:

   ```
   EXIT nn {ON|OFF} entry1 (entry2 ... entryn)
   ```

   To specify that an EXIT 11 condition calls CA ACF2 for z/VM, code the EXIT 11 statement as EXIT 11 ON DMTAC2. You can call multiple entry points for a particular exit. You can specify DMTAC2 anywhere in the EXIT 11 list. To specify multiple entry points for an EXIT 11, code the statement as EXIT 11 ON DMTAC2 myexit11. For additional information on RSCS exits, see the IBM publication *Virtual Machine Remote Spooling Communications* Subsystem *Exit Customization*.

3. RSCS must be able to locate the CA ACF2 for z/VM exit. You can put the exit in the RSCS LOADLIB or in your own LOADLIB. XEDIT the RSCS LKEDCTRL file and add these three lines to the bottom of the file:

   ```
   INCLUDE DMTAC2V3      (name of text deck)
   ENTRY DMTAC2          (name of entry point)
   NAME DMTAC2           (name in LOADLIB)
   ```

   To put the CA ACF2 for z/VM exit in your own LOADLIB, create a LKEDCTRL file with these three lines of text:

   ```
   INCLUDE DMTAC2V3      (name of text deck)
   ENTRY DMTAC2          (name of entry point)
   NAME DMTAC2           (name in LOADLIB)
   ```

   Execute VMFLKED to rebuild the LOADLIB.

   RSCS must have the name of your LOADLIB in its search sequence. To tell RSCS where to find your exit, modify the GLOBAL LOADLIB statement in the PROFILE exec. For example, if you put the CA ACF2 for z/VM exit into a LOADLIB called USEREXIT, the GLOBAL statement reads GLOBAL LOADLIB RSCS USEREXIT.

CA-REGISTER

CA-REGISTER is a component of CA-CIS that lets an administrator define a new user to all system products at the same time through one set of definition panels. For example, CA-REGISTER lets you start a single transaction that defines a new user to CA ACF2 for z/VM, IBM's Shared File System, and even to user-defined subsystems through user exits to REGISTER.

Configuring CA-REGISTER

To configure CA-REGISTER, follow these steps:

1. Make the ACF2EXIT EXEC file available to the REGISTER service machine. Typically, this is the CAIMAINT 291 disk that is linked to the REGISTER service machine as the 192 disk.

2. Modify the REGISTER APPLS for CA ACF2 for z/VM.

   To change the REGISTER APPLS file, invoke the CA-ACTIVATOR task S510I002 or modify this file manually through XEDIT and add the following line:

   ```
   Col. 1        Col. 20       Col. 30      Col. 40
   CA ACF2 for z/VM      prototype     ACF2EXIT     ACF2ZOOM
   ```

The format of the REGISTER APPLS file is:

Columns	Description
01-18	Application name, such as CA ACF2 for z/VM
20-27	Prototype name. This is the model that an ACF INSERT USING(prototype) command would use. Your users can override the default that you initially choose or you can give each user his own REGISTER APPLS file.
30-37	Filename of the EXIT to be invoked in the REGISTER service machine. For CA ACF2 for z/VM, this is ACF2EXIT EXEC.
40-47	Filename of the ZOOM program to be invoked in the user's virtual machine. For CA ACF2 for z/VM, this is ACF2ZOOM CAIPANEL.

3. Copy the ACF2ZOOM CAIPANEL and M9PFZOOM PFKEYS file to the same disk you created to hold the CA ACF2 for z/VM full-screen panels. If the REGISTER panel files are not on the same disk as the CA ACF2 for z/VM full-screen panels, you must access both disks to use the REGISTER ZOOM function with the CA ACF2 for z/VM application.

The REGISTER ZOOM function requires the CA ACF2 for z/VM full-screen LID maintenance panels to operate. Below is a subset of the CA ACF2 for z/VM full-screen panel files needed to execute the ZOOM function:

- M9PA1110 CAIPANEL
- M9PA1120 CAIPANEL
- M9PA1130 CAIPANEL
- M9PA1140 CAIPANEL
- M9FU1100 CAIPANEL
- M9FU1000 CAIPANEL
- M9FU0000 CAIPANEL
- $ACF2 MODULE
- ACF MODULE
- $ACF2MSG MODULE

- ACFMSG MODULE
- ECAIGLO MODULE

The ZOOM program for CA ACF2 for z/VM uses the CA-supplied full-screen panels. If you have developed your own panels using the new CA ACF2 for z/VM REXX interface, you must write your own ZOOM program. For information on how to write a REGISTER ZOOM program, see the *Administrator Guide*.

Shared Database Support

CA ACF2 for z/VM provides a means of sharing VSAM database clusters between CA ACF2 for z/VM systems. This shared database lets sites that use multiple VM systems establish common databases to reduce maintenance and system overhead. This section explains how to activate and maintain shared databases. For sharing between VM and z/OS, use the database synchronization component.

Shared database support uses VSE/VSAM Version 2. For VSAM databases, CA ACF2 for z/VM requires you to specify more than one data set name level. If you specify only one level, CA ACF2 for z/VM assumes that you are specifying a CMS data set name. If your site uses VM on multiple CPUs, see the Shared Database Considerations for VM-Only Sites section see the Shared Databae Considerations for VM-Only Sites section.

Activating Shared Databases on VM Systems

Follow these steps to implement CA ACF2 for z/VM shared databases:

1. Modify the user directory and, if necessary, the IBM HCPRIO module. There are different requirements for this step, depending on how your site intends to share databases (sharing with guest virtual machines, sharing with other processors, or sharing with both). In addition, you can share databases on a full disk pack or a partial pack.
2. Modify the service machine ACFSTART EXEC.
3. Modify, assemble, and reload the ACFFDR.
4. Merge the databases using the ACFDBVSM utility program.

Details follow on the next few pages.

Step 1: Modify the User Directory and HCPRIO

Some sites share their database with a guest VM system. Other sites share their database with one or more processors. Whatever the case, your configuration setup depends on whether the database resides on a full pack minidisk or a partial pack minidisk.

Skip to Requirements for Sharing on a Partial Pack Minidisk if you intend to share databases on a partial pack minidisk.

Requirements for Sharing on a Full Pack Minidisk

The requirements differ, depending on whether you are sharing with other processors, guest virtual machines, or both.

For each processor sharing the full pack minidisk, it is imperative that there be only one system directory MDISK definition in MWV mode. All other accesses by that processor must be through a directory LINK definition in MW mode to the pack's MDISK definition in MWV mode.

For each processor sharing the full pack minidisk, we recommend you give the CA-ACF2 service machine the sole MDISK definition in MWV mode.

Full Pack Scenarios

Different scenarios are possible for sharing databases on a full pack minidisk. Follow the instructions under the heading that coincides with the scenario at your site:

- Sharing Only with Other Guest Machines:

  The system directory for the CA ACF2 for z/VM service machine must include an MDISK definition in MWV mode for access to the full-pack minidisk or must include a LINK definition in MW mode to the full pack's MDISK definition in MWV mode.

  The system directory for the guest machine can include an MDISK definition in MWV mode for access to the full pack minidisk (as long as you did not define another MWV MDISK statement) or it must include a LINK definition in MW mode to the full pack's MDISK definition in MWV mode.

  You can define more than one guest machine on the VM processor.

The following processor directory entries illustrate a possible scenario for sharing CA ACF2 for z/VM databases with a guest machine on a full-pack minidisk:

```
USER ACF2VM . . .

    .

    .

MDISK 300 3380 000 885 VSMPCK MWV

    .

    .

    .

USER GUESTVM

    .

    .

LINK ACF2VM 300 300 MW
```

Proceed to Step 3: Modify, Assemble, and Reload the ACFFDR after you modify the user directory.

- Sharing Only with Other Processors:

  The system directory for the CA ACF2 for z/VM service machine on the first processor must include an MDISK definition in MWV mode for access to the full pack minidisk or a LINK definition in MW mode to the full pack's MDISK definition in MWV mode.

  The system directory for the CA ACF2 for z/VM service machine on another processor must include an MDISK definition in MWV mode for access to the full pack minidisk or a LINK definition in MW mode to the full pack's MDISK definition in MWV mode.

  You must generate the real DASD as SHARED in the HCPRIO definition for the device.

  The following processor directory entries illustrate a possible scenario for sharing CA ACF2 for z/VM databases with another processor on a full pack minidisk:

  ```
  USER ACF2VM . . .

      .

      .

  MDISK 300 3380 000 885 VSMPCK MWV
  ```

  Other Processor Directory Entries:

  ```
  USER ACF2VM . . .

      .

      .

  MDISK 300 3380 000 885 VSMPCK MWV
  ```

  Proceed to Step 3: Modify, Assemble, and Reload the ACFFDR after you modify the user directory and HCPRIO.

- Sharing with Guest Machines and Other Processors:

  The system directory for the CA ACF2 for z/VM service machine on the first processor must include an MDISK definition in MWV mode for access to the full-pack minidisk or a LINK definition in MW mode to the full pack's MDISK definition in MWV mode.

  The system directory for the guest machine on the first processor can include an MDISK definition in MWV mode for access to the full pack minidisk (as long as you did not define another MWV MDISK statement) or a LINK definition in MW mode to the full pack's MDISK definition in MWV mode. You can define more than one guest machine.

  The system directory for the CA ACF2 for z/VM service machine on another processor must include an MDISK definition in MWV mode for access to the full pack minidisk or a LINK definition in MW mode to the full pack's MDISK definition in MWV mode.

  The system directory for the guest machine (if there is one) on the other processor can include an MDISK definition in MWV mode for access to the full pack minidisk (as long as you did not define another MWV MDISK statement) or a LINK definition in MW mode to the full pack's MDISK definition in MWV mode. You can define more than one guest machine.

  You must generate the real DASD as SHARED in the HCPRIO definition for the device.

  The following directory entries illustrate a possible scenario for sharing CA ACF2 for z/VM databases with a guest machine and another processor on a full pack minidisk:

  ```
  USER ACF2VM . . .

    ...

  MDISK 300 3380 000 885 VSMPCK MWV

    ...

  USER GUESTVM . . .

    ...

  LINK ACF2VM 300 300 MW
  ```

  Proceed to Step 3: Modify, Assemble, and Reload the ACFFDR once you modify the user directory and HCPRIO.

Requirements for Sharing on a Partial Pack Minidisk

The requirements differ, depending on whether you are sharing with other processors, guest virtual machines, or both.

For each processor sharing the partial pack minidisk, it is imperative that there be only one system directory MDISK definition in MWV mode. All other accesses by that processor must be through a directory LINK definition in MW mode to the pack's MDISK definition in MWV mode. There is one exception to this rule. For database sharing on a partial pack minidisk with other processors, the processor requires two MDISK statements: one for the partial pack and one for the full pack the partial pack is on.

Partial Pack Scenarios

Different scenarios are possible for sharing databases on a partial pack minidisk. Follow the instructions under the heading that coincides with the scenario at your site:

- Sharing Only with Other Guest Machines:

  The system directory for the CA ACF2 for z/VM service machine must include an MDISK definition in MWV mode for access to the partial pack minidisk or a LINK definition in MW mode to the partial minidisk's MDISK definition in MWV mode.

  The system directory for the guest machine can include an MDISK definition in MWV mode for access to the partial-pack minidisk (as long as you did not define another MWV MDISK statement) or a LINK definition in MW mode to the partial minidisk's MDISK definition in MWV mode. You can define more than one guest machine on each processor.

  The following processor directory entries illustrate a possible scenario for sharing CA ACF2 for z/VM databases with a guest machine on a partial pack minidisk:

  ```
  USER ACF2VM . . .

    ...

  MDISK 300 3380 440 020 VSMPCK MWV

    ...

  USER GUESTVM

    ...

  LINK ACF2VM 300 300 MW
  ```

  Proceed to Step 3: Modify, Assemble, and Reload the ACFFDR after you modify the user directory.

- Sharing Only with Other Processors:

  The system directory for the CA-ACF2 service machine on the first processor must include an MDISK definition in MWV mode for access to the partial-pack minidisk or a LINK definition in MW mode to the partial minidisk's MDISK definition in MWV mode.

  To queue real reserves to the hardware, the processor requires another MDISK statement. We recommend you assign both MDISK statements to the CA ACF2 for z/VM service machine.

  The directory for the CA ACF2 for z/VM service machine on the first processor must include an MDISK definition in MWV mode for access to the full pack minidisk or a LINK definition in MW mode to the full pack's MDISK definition in MWV mode.

  The system directory for the CA ACF2 for z/VM service machine on another processor must include an MDISK definition in MWV mode for access to the partial pack minidisk or a LINK definition in MW mode to the partial minidisk's MDISK definition in MWV mode.

  You must generate the real DASD as SHARED in the HCPRIO definition for the device.

  The following processor directory entries illustrate a possible scenario for sharing CA ACF2 for z/VM databases with a VM processor on a partial pack minidisk:

  ```
  USER ACF2VMEA . . .

    ...

  MDISK 300 3380 440 020 VSMPCK MWV

  MDISK 308 3380 000 885 VSMPCK MWV
  ```

  Proceed to Step 3: Modify, Assemble, and Reload the ACFFDR after you modify the user directory and HCPRIO.

- Sharing with Guest Machines and Other Processors:

  When sharing with guest machines and other processors, the partial pack minidisk must start on cylinder zero. Define this partial pack minidisk for the CA ACF2 for z/VM service machine and the guest's directory entries as a full pack minidisk. It is the VTOC that DSF created to be a minipack that forces VSAM to treat the full pack as a partial pack.

  To queue real reserves to the hardware, the system directory for the CA ACF2 for z/VM service machine on the first processor must include an MDISK definition in MWV mode for access to the partial pack minidisk (defined as a full pack) or a LINK definition in MW mode to the partial (defined as full) minidisk's MDISK definition in MWV mode.

  The system directory for the guest machine on the can include an MDISK definition in MWV mode for access to the partial pack minidisk (defined as a full pack), as long as you did not define another MWV MDISK statement, or a LINK definition in MW mode to the partial (defined as full) minidisk's MDISK definition in MWV mode. You can define more than one guest machine.

  The system directory for the CA ACF2 for z/VM service machine on another processor must include an MDISK definition in MWV mode for access to the partial pack minidisk or a LINK definition in MW mode to the partial minidisk's MDISK definition in MWV mode.

  The system directory for the guest machine (if there is one) on the other processor can include an MDISK definition in MWV mode for access to the partial pack minidisk (as long as you did not define another MWV MDISK statement) or a LINK definition in MW mode to the partial minidisk's MDISK definition in MWV mode. You can define more than one guest machine.

  You must generate the real DASD as SHARED in the HCPRIO definition for the device.

  The following directory entries illustrate a possible scenario for sharing CA ACF2 for z/VM databases with a guest machine and another processor on a partial-pack minidisk:

  ```
  USER ACF2VMEA . . .

    ...

  MDISK 300 3380 000 885 VSMPCK MWV

    ...

  USER GUESTVM

    ...

  LINK ACF2VMEA 300 300 MW
  ```

Proceed to Step 2: Modify the Service Machine ACFSTART EXEC after you modify the user directory and HCPRIO.

Step 2: Modify the Service Machine ACFSTART EXEC

Modify the service machine ACFSTART EXEC to include ACCESS and DLBL statements for the VSAM files.

```
   'ACCESS 300 V'            /* access vsam database disk      */

/* The VSAM DSN's on the DLBL statements below should be changed
   to reflect the actual DSN's of your VSAM CATALOG & datasets  */

   'DLBL IJSYSCT V DSN CATALOG MASTER'
   'DLBL RULES V DSN VMVSAM.ACF.RULES (VSAM'
   'DLBL LID V DSN VMVSAM.ACF.LOGONIDS (VSAM'
   'DLBL INFO V DSN VMVSAM.ACF.INFOSTG (VSAM'
```

Step 3: Modify, Assemble, and Reload the ACFFDR

To define the database groups, modify the @DDSN macro of the ACFFDR. You can define up to 16 database groups. See the *Administrator Guide* for complete instructions.

```
@DDSN
  PRIMARY,                    ** PRIMARY GROUP
  RULE='RULES',               ** CMS RULES CLUSTER
  LID='SYS1.ACF.LOGONIDS',    ** VSAM LOGONID CLUSTER
  INFO='INFO'                 ** CMS GEN RESOURCE CLSTR
```

Assemble the ACFFDR. After moving the ACFFDR to the service machine 193 disk, issue the following CP command to reload the new version into storage:

```
ACFSERVE RELOAD FDR
```

See the *Administrator Guide* for more information about ACFSERVE commands.

Step 4: Merge the Databases Using the ACFDBVSM Utility

Use the ACFDBVSM utility to merge the CMS databases into VSAM databases. To perform the merge with the ACFDBVSM EXEC, see the *Report and Utilities Guide* for more information.

Shared Database Considerations for VM-Only Sites

Sites running only VM on multiple CPUs must complete the following procedure before establishing a shared database. Because you cannot merge CMS databases with other CMS databases, you must convert one of the CMS databases to a VSAM database before you activate the shared database feature. You must be running VSE/VSAM Version 2 to support database sharing.

If you used CA-ACTIVATOR to install CA ACF2 for z/VM with VSAM shared databases or the ACF2VSAM EXEC to convert your system to shared databases, the steps outlined below are already complete. If you used these execs, this section is for reference only.

Step 1: Create VSAM Minidisk

Establish space on a volume and format the volume using the Data Services Facility (DSF) to create a VSAM minidisk. Below is an example of establishing a 20-cylinder minidisk on a 3380, linked as device 300. This step requires that no class I reader queue files exist. If any of these files exist, dispose of them or transfer them to another class.

```
query reader all class i
purge reader class i
spool punch * class i
punch ipl dsf s (noheader
spool reader class i
ipl 00c clear
ICK005E DEFINE INPUT DEVICE, REPLY 'DDDD,CUU' OR 'CONSOLE'
console
ICK006E DEFINE OUTPUT DEVICE, REPLY 'DDDD,CUU' OR 'CONSOLE'
console
ENTER INPUT/COMMAND
init unit(300) novfy volid(vmvsam) mimic(mini(20)) devtype(3380)
ICK003D REPLY U TO ALTER VOLUME 300 CONTENTS, ELSE
u
```

When processing completes, enter the following command:

```
#CP I 190 CL
```

You have created your VSAM minidisk. If any errors occurred during processing, reexamine the procedure.

Step 2: Create VSAM Master Catalog

Create a four-cylinder VSAM master catalog named CATALOG MASTER on the OS minidisk, starting at cylinder one. Following is a sample ACF2DEFM EXEC and a DEFCAT AMSERV file to create this catalog:

Sample ACF2DEFM EXEC

```
/*                                                   */
/*   Access the catalog master disk if not invoked   */
/*   thru the ACF2VSAM EXEC.                         */
/*                                                   */
'SET DOS ON  (VSAM'
'ASSGN SYSCAT V'
queue '19 76'
queue
'SET CMSTYPE HT'
'DLBL IJSYSCT V DSN CATALOG MASTER (SYSCAT PERM EXTENT'
'SET CMSTYPE RT'
'AMS ACF2DEFM'
cc = rc
'SET DOS OFF'
exit cc
```

Sample ACF2DEFM AMSERV File

```
DEFINE MASTERCATALOG -
   (NAME(CATALOG.MASTER) -
    VOLUME(VMVSAM) -
    CYL(4) -
    FILE(IJSYSCT))The ACF2DEFM EXEC works with the ACF2DEFM AMSERV to produce the
master catalog.  Insert your values into the ACF2DEFM EXEC and enter the following
command:
```

```
ACF2DEFM
```

Examine the file's ACF2DEFM LISTING for any errors. If there are no errors, your master catalog is created.

Step 3: Define VSAM Databases

Define your CA ACF2 for z/VM VSAM databases. The following sample ACF2DEFC EXEC and ACF2DEFC AMSERV file perform this action for you:

Sample ACF2DEFC EXEC

```
/*                                                                  */
/*  ACCESS THE REQUIRED DISKS IF NOT INVOKED BY THE ACF2VSAM EXEC.  */
'SET DOS ON  (VSAM'
'SET CMSTYPE HT'
'ASSGN SYS001 V'
'ASSGN SYS002 V'
'ASSGN SYS003 V'
'ASSGN SYSCAT V'
'DLBL IJSYSCT V DSN CATALOG MASTER (SYSCAT'
queue '95 76 V SYS001'
queue
'DLBL LFILE V (SYS001 EXTENT'
queue '02 02 V SYS001'
queue
'DLBL LINDX V (SYS001 EXTENT'
queue '171 76 V SYS002'
queue
'DLBL RFILE V (SYS002 EXTENT'
queue '04 02 V SYS002'
queue
'DLBL RINDX V (SYS002 EXTENT'
queue '247 76 V SYS003'
queue
'DLBL IFILE V (SYS003 EXTENT'
queue '06 02 V SYS003'
queue
'DLBL IINDX V (SYS003 EXTENT'
'SET CMSTYPE RT'
'AMS ACF2DEFC'
cc = rc
'SET DOS OFF'
exit cc
```

Sample ACF2DEFC AMSERV File

```
DEFINE CLUSTER(NAME(VMVSAM.ACF.LOGONIDS) -
               RECORDSIZE(512,1024) -
               FREESPACE(30,30) -
               UNIQUE -
               OWNER(ACF) -
               KEYS(8,0) -
               SHAREOPTIONS(4,4)) -
       DATA(NAME(VMVSAM.ACF.LOGONIDS.DATA) -
               VOL(VMVSAM) -
               FILE(LFILE) -
               CYLINDERS(4) -
               CONTROLINTERVALSIZE(4096)) -
       INDEX(NAME(VMVSAM.ACF.LOGONIDS.INDEX) -
               VOL(VMVSAM) -
               FILE(LINDX) -
               TRACKS(2) -
               CONTROLINTERVALSIZE(4096))
DEFINE CLUSTER(NAME(VMVSAM.ACF.RULES) -
               RECORDSIZE(400,4088) -
               FREESPACE(30,30) -
               UNIQUE -
               OWNER(ACF) -
               KEYS(8,0) -
               SHAREOPTIONS(4,4)) -
       DATA(NAME(VMVSAM.ACF.RULES.DATA) -
               VOL(VMVSAM) -
               FILE(RFILE) -
               CYLINDERS(4) -
               CONTROLINTERVALSIZE(4096)) -
       INDEX(NAME(VMVSAM.ACF.RULES.INDEX) -
               VOL(VMVSAM) -
               FILE(RINDX) -
               TRACKS(2) -
               CONTROLINTERVALSIZE(4096))
DEFINE CLUSTER(NAME(VMVSAM.ACF.INFOSTG) -
               RECORDSIZE(400,4088) -
               FREESPACE(30,30) -
               UNIQUE -
               OWNER(ACF) -
               KEYS(44,32) -
               SHAREOPTIONS(4,4)) -
       DATA(NAME(VMVSAM.ACF.INFOSTG.DATA) -
               VOL(VMVSAM) -
               FILE(IFILE) -
               CYLINDERS(4) -
               CONTROLINTERVALSIZE(4096)) -
       INDEX(NAME(VMVSAM.ACF.INFOSTG.INDEX) -
               VOL(VMVSAM) -
               FILE(IINDX) -
               TRACKS(2) -
               CONTROLINTERVALSIZE(4096))
```

The ACF2DEFC EXEC works with the ACF2DEFC AMSERV file to create three VSAM clusters. Each cluster contains the catalog, the data component, and index component. The data components must reside on a cylinder boundary.

CA ACF2 for z/VM creates the data component immediately following the master catalog on cylinders 5 through 8. It puts the LOGONID data component on cylinders 9 through 12. CA ACF2 for z/VM puts the INFOSTG data component on cylinders 13 through 17. There are no cylinder boundary restrictions for the index components, so CA ACF2 for z/VM puts them on cylinder zero following VTOC.

To accomplish this task, enter the following command:

```
ACF2DEFC
```

When the ACF2DEFC EXEC finishes processing, your databases are defined. Examine the ACF2DEFC LISTING for errors.

Step 4: Initialize VSAM Databases

Initialize the CA ACF2 for z/VM VSAM databases through the ACFLINIT EXEC. Below is a sample of this exec.

Sample ACFLINIT EXEC

```
/*                                                                 */
/*  Access the required disks if not invoked by the ACF2VSAM EXEC. */
arg ar                                                         TM93507
'DLBL * CLEAR'
'DLBL IJSYSCT V DSN CATALOG MASTER'
'DLBL LID V DSN VMVSAM ACF LOGONIDS (VSAM'
'DLBL RULES V DSN VMVSAM ACF RULES (VSAM'
'DLBL INFO V DSN VMVSAM ACF INFOSTG (VSAM'
'QUERY CMSLEVEL (STACK LIFO'
parse pull . . cmslevel .
cmslevel = translate(cmslevel,' ',',')
cmslevel = strip(cmslevel,t)
if cmslevel < 5.5 then 'EXECOS ACFLINIT' ar
                  else 'ACFLINIT' ar
exit rc
```

To initialize the database, enter isoid, where isoid is the logonid of the security administrator. (If the isoid operand does not exist, CA ACF2 for z/VM assumes ACFUSER is the logonid.) Ensure that the logonid exists in the system directory so that the security administrator can log on.

Step 5: Convert VM Directory

Convert the system directory for CA ACF2 for z/VM use. For information on converting your VM/SP directory to the LIDREC format, see ACF2TASK EXEC Step 1.

Expanded Rule Size

In previous releases of CA ACF2 for z/VM, the maximum access rule or resource rule that could be compiled was 4 Kb. CA ACF2 for z/VM release 4.2 relieves that constraint and allows a maximum of 32K when using VSAM databases. This is a significant benefit for security administrators because it makes rule maintenance straightforward and easy to perform. The previous release limit of 4 Kb forced the use of the NEXTKEY parameter to break up rule sets that exceeded the limit. With the expansion to 32K rule size, administrators can create larger rule sets, simplifying maintenance.

This feature does not limit rule size to 4 Kb or 32K. You, the user, determine the rule size. You can set it at any value between 4 Kb and 32K. If 8K is optimal, then use 8K. If you need 32K rule sets, then use the 32K limit. But the greater the value, the more storage is needed per user to be able to process. All validation processing requires a buffer size equal to the maximum rule size allowed. The maximum rule size is determined at CA ACF2 for z/VM start up. The Rules database (access rules) and the Infostorage database (resource rules) must be the same size, and must be VSAM databases.

If you use the RULELONG option of the RULEOPTS VMO record, you are required to increase the current size of the databases. Turning this option on lets CA ACF2 for z/VM process the current rule sets and any new rule sets. CA ACF2 for z/VM creates a new version record if this option is set and a compile of a ;record is performed. The advantage of setting this option is the restructuring of the current rule records and being able to take advantage of some of the new features of rules. For example, the ACTIVE data field is only valid if RULELONG is turned on. Also, any fields added in future releases will only be added to this new version rule record.

With RULELONG on, any compiled records are automatically converted from the old version to the new version. No conversions are required or necessary to use this new option. The only warning with this option is that, once the rule sets are converted to the new version (RULELONG), any rule sets compiled cannot be used if there is ever a need to go back to the old version (NORULELONG). Also, when the databases are shared, all systems must be running a release that supports RULELONG.

CA ACF2 for z/VM validates the rules database size during startup or a RULEOPTS record refresh if the RULELONG option is set. CA ACF2 for z/VM disables RULELONG if the databases were not expanded and displays message ACFpgm47EW.

Once this option is turned on, the databases are no longer compatible with previous CA ACF2 for z/VM releases or any release.

See the *Administrator Guide* for information about the RULELONG option.

TCP/IP FTP Interface

The CA ACF2 for z/VM interface with TCP/IP FTP is implemented by using the ESM feature of FTP. CA ACF2 for z/VM uses our SAF interface to process calls from CA-ESM (release 1.1 or above) to provide the ESM functions to FTP.

Follow the steps below to implement the TCP/IP FTP Interface:

1. If your site has installed CMS security, make sure that the FTPSERVE ID is not included in the MAINT VMO record. If it is, remove the FTPSERVE ID from the MAINT VMO record and use the ACFSERVE RELOAD CONTROL VMO MAINT command to reload the MAINT VMO record. Also, make sure that the FTPSERVE ID does NOT have the NON-CNCL attribute.

   The combination of FTPSERVE being in the MAINT VMO record and the NON-CNCL attribute will allow FTPSERVE to bypass CMS file level security. This has been replaced with the VMD4FSEC attribute that will cause CMS file access by the FTPSERVE ID to be validated against the user that initiated the file transfer.

2. From a security administrator's user ID, provide the following logonid attributes to FTPSERVE:

   ```
   VMESM
   ```

   ```
   VMSAF
   ```

   ```
   VMD4AUTH
   ```

   ```
   VMD4TARG
   ```

   ```
   VMD4RSET
   ```

   ```
   VMD4FSEC (only if CMS security is installed)
   ```

   Important! *If the FTPSERVE directory entry includes the IBM option LNKNOPAS, remove this option. With this option, FTPSERVE bypasses minidisk passwords when it does the LINK to a minidisk to make the FTP connection. Without this option, FTPSERVE respects whatever setting you specified for CA ACF2 for z/VM for minidisk passwords.*

   If you set the CA ACF2 for z/VM option DIRMDPW=RESPECT and a user tries to use FTP to get to a minidisk protected with passwords, he is prompted for the minidisk password.

   If you set DIRMDPW=DENYNOPW, then minidisk passwords are ignored except when no minidisk passwords exist at all on the MDISK statement. Then, the FTP connection to the minidisk is denied.

   If the DIRMDPW=IGNORE option is set in CA ACF2 for z/VM, it does not matter whether the LNKNOPAS option is set in the directory.

   However, we still recommend not using LNKNOPAS for FTPSERVE in case the DIRMDPW= setting is ever changed.

3. Copy the following file from the CAIMAINT 291 to the TCPMAINT 198 minidisk to make it accessible to FTPSERVE and other TCP/IP service machines, replacing any existing version:

   ```
   ACFSAFA0 MODULE
   ```

   If not already there, copy the following file from the CIAMAINT 291 to the TCPMAINT 198 minidisk tomake it accessible to FTPSERVE and other TCP/IP service machines:

   ```
   CAIRPI PARMS
   ```

   Modify the CAIRPI PARMS file on the TCPMAINT 198 minidisk to make sure that the APPLNAME is OTHER (not the default SFS):

   ```
   ESMID *RPI     APPLNAME OTHER
   ```

   In the SYSTEM DTCPARMS file, change the following lines in the :nick.ftp section. If you don't have a section with the :nick.ftp tag, you can copy the section from the IBM DTCPARMS file. The SYSTEM DTCPARMS file should be located on the TCPMAINT 198 disk, and it overides the IBM DTCPARMS file.

   From:

   ```
   :ESM_Enable.NO
   :ESM_Validate.RPIVAL
   :ESM_Racroute.RPIUCMS
   ```

   To:

   ```
   :ESM_Enable.Yes
   :ESM_Validate.ACFSAFA0
   :ESM_Racroute.RPIUCMS
   :VMLink.CAIMAINT 391
   ```

4. The use of CA-ESM requires that your FTP service machine (normally FTPSERVE) directory entry must include the following statement:

   ```
   IUCV *RPI
   ```

5. To use the FTP option that allows you to transfer files to a user's VM RDR, you need to write resource rules to allow the transfer. By default, the resource type is VMR. To allow a user with a UID value of USERA put a file into the VM RDR of TARGETID, the following rule is required:

   ```
   $KEY(TARGETID) TYPE(VMR)
    UID(USERA) SERVICE(UPDATE) ALLOW)
   ```

   To allow USERA to execute the DIR and DELETE commands, write the following rule:

   ```
   $KEY(TARGETID) TYPE(VMR)
    UID(USERA) SERVICE(UPDATE,DELETE) ALLOW
   ```

   Note: If the rule is created without the SERVICE keyword, it is equivalent to specifying all services, which allows PUT, DIR, and DELETE FTP commands.

TCP/IP NFS Interface

The CA ACF2 for z/VM interface with TCP/IP NFS is implemented by using the ESM feature of NFS. CA ACF2 for z/VM uses our SAF interface to process calls from CA-ESM (release 1.1 or above) to provide the ESM functions to NFS.

Follow the steps below to implement the TCP/IP NFS Interface:

1. If your site has installed CMS security, add the VMNFS ID to the MAINT VMO record and use the ACFSERVE RELOAD CONTROL VMO MAINT command to reload the maint vmo record. This prevents NON CNCL logging records from being created when CMS files are validated.

2. From a security administrator's user ID, provide the following logonid attributes to VMNFS:

   ```
   VMESM
   ```

   ```
   VMSAF
   ```

   ```
   VMD4AUTH
   ```

   ```
   VMD4TARG
   ```

   ```
   VMD4RSET
   ```

   ```
   NON CNCL (only if CMS security is installed)
   ```

3. Copy the following files to the TCPMAINT 198 minidisk to make them accessible to VMNFS and other TCP/IP service machines, if not already there:

   ```
   ACFSAFA0 MODULE
   ```

   If not already there, copy the following file from the CAIMAINT 291 to the TCPMAINT 198 minidisk to make it accessible to VMNFS and other TCP/IP service machines:

   ```
   CAIRPI PARMS
   ```

   Modify the CAIRPI PARMS file on the TCPMAINT 198 minidisk to make sure that the APPLNAME is OTHER (not the default SFS):

   ```
   ESMID *RPI     APPLNAME OTHER
   ```

4. In the SYSTEM DTCPARMS file, change the following lines in the :nick.nfs section. If you don't have a section with the :nick.nfs tag, you can copy the section from the IBM DTCPARMS file. The SYSTEM DTCPARMS file should be located on the TCPMAINT 198 disk, and it overides the IBM DTCPARMS file.

   From:

   ```
   :ESM_Enable.NO

   :ESM_Validate.RPIVAL

   :ESM_Racroute.RPIUCMS
   ```

   To:

   ```
   :ESM_Enable.Yes

   :ESM_Validate.ACFSAFA0

   :ESM_Racroute.RPIUCMS

   :VMLink.CAIMAINT 391
   ```

   Note: If you are upgrading from CA ACF2 for z/VM release 4.1 or earlier, change the ESM_Racroute.RPIDUMY back to the original IBM value of ESM_Racroute.RPIUCMS and add the new :VMLink. tag as listed above.

5. The use of CA-ESM requires that your NFS service machine (normally VMNFS) directory entry must include the following statement:

   ```
   IUCV *RPI
   ```

TCP/IP REXEC Interface

The CA ACF2 for z/VM interface with TCP/IP REXEC is implemented by using the ESM feature of REXEC. CA ACF2 for z/VM uses our SAF interface to process calls from CA-ESM (release 1.1 or above) to provide the ESM functions to REXEC.

Follow the steps below to implement the TCP/IP REXEC Interface:

1. From a security administrator's user ID, provide the following logonid attributes to REXECD:

   ```
   VMESM

   VMSAF
   ```

2. Copy the following file from the CAIMAINT 291 to the TCPMAINT 198 minidisk to make it accessible to REXECD and other TCP/IP service machines, replacing any existing version:

   ```
   ACFSAFAO MODULE
   ```

   If not already there, copy the following file from the CAIMAINT 291 to the TCPMAINT 198 minidisk to make it accessible to REXECD and other TCP/IP service machines:

   ```
   CAIRPI PARMS
   ```

   Modify the CAIRPI PARMS file on the TCPMAINT 198 minidisk to make sure that the APPLNAME is OTHER (not the default SFS):

   ```
   ESMID *RPI      APPLNAME OTHER
   ```

3. In the SYSTEM DTCPARMS file, change the following lines in the :nick.rexec section. If you don't have a section with the :nick.rexec tag, you can copy the section from the IBM DTCPARMS file. The SYSTEM DTCPARMS file should be located on the TCPMAINT 198 disk, and it overides the IBM DTCPARMS file.

   From:

   ```
   :ESM_Enable.NO

   :ESM_Validate.RPIVAL

   :ESM_Racroute.RPIUCMS
   ```

   To:

   ```
   :ESM_Enable.Yes

   :ESM_Validate.ACFSAFA0

   :ESM_Racroute.RPIUCMS

   :VMLink.CAIMAINT 391
   ```

   Note: If you are upgrading from CA ACF2 for z/VM release 4.1 or earlier, change the ESM_Racroute.RPIDUMY back to the original IBM value of ESM_Racroute.RPIUCMS and add the new :VMLink. tag as listed above.

4. The use of CA-ESM requires that your REXEC service machine (normally REXECD) directory entry must include the following statement:

   ```
   IUCV *RPI
   ```

Unit Record Logging Support

To start real unit record devices, issue the following CP command:

```
START
```

They are also started automatically at IPL if you do not specify the DRAIN option.

Optional unit record logging support produces an SMF record when output begins or when input from a reader begins. Unit record logging monitors the files processed on real devices.

This section provides information on implementing unit record logging support on your system.

Implementing Unit Record Logging Support

Set the VMXAOPTS macro URLOG operand to YES (the default) to activate unit record logging support. Enter the following command to display this setting:

```
ACF SHOW ACTIVE
```

To implement unit record logging support, you must perform the following steps:

1. Instruct your system operators to always respond DRAIN to the spool system startup prompt. This eliminates potentially confusing messages, problems with spool files, and the status of unit record devices.
2. Be sure no site modifications or other products use VMUSRFD (VMDUSERx field in the VMDBK of the system ID). CA ACF2 for z/VM requires that VMUSRFD, defined in VMXAOPTS, must be available for the system ID (anchored off ASYSVM in the PSA). CA ACF2 for z/VM uses VMUSRFD to build a dummy minilogonid off the system VMDBK to place user information in the reports.
3. Ensure that you issued the START command before input or output begins on a real device and modify the unit record class definitions in HCPRIO so no users can create spool files for those devices. If you start the system without the DRAIN option, CA ACF2 for z/VM does not start I/O to those devices. Or, you can note the classes defined in HCPRIO. Remember that the system can automatically start spool files for these classes if you do not specify the DRAIN option.

Output

CA ACF2 for z/VM writes an SMF record that appears in the ACFRPTCL report with a command name of $START. $START is a pseudocommand name CA ACF2 for z/VM sends when it starts input or output on a real device. For additional information about the ACFRPTCL report, see the *Report and Utilities Guide*.

VM Batch System Support

The VM Batch system components let users submit batch jobs for execution in the VM environment. Installing the CA ACF2 for z/VM support for the VM batch components extends CA ACF2 for z/VM security controls to users submitting these batch jobs.

Whenever users submit jobs to VM Batch, CA ACF2 for z/VM ensures that each job inherits the submitter's logonid. As each job initiates in the component, CA ACF2 for z/VM does normal validation processing. It makes a decision on whether to allow processing to continue based on the access authority and privileges of the submitter.

Installing VM Batch Facility Support

Follow the steps below to implement CA ACF2 for z/VM support for the IBM VM Batch Facility (product number 5664-364).

1. Make sure your VM Batch monitor machine and all task machines have read access to the system Y-disk (usually MAINT 19E) and the system S-disk (usually MAINT 190).

   ```
   $key(maint)
    v0190.-.- uid(batch) read(a)
    v019E.-.- uid(batch) read(a)
   ```

2. Set up logonids for the monitor and task machines. We provide two logonid privileges to define batch machines, VMD4TARG and VMD4AUTH. VM Batch task machines (defined by TASK keywords in the CONTROL FILE the VM Batch Facility uses) and targets of the alternate user diagnose x'D4' require the VMD4TARG privilege. The VM Batch monitor and issuers of the alternate user diagnose x'D4' require the VMD4AUTH privilege. Because VMBATCH does not use DIAG 84 on VM systems, the DG84DIR privilege is not necessary.

   Give the VMD4TARG and VMD4AUTH privileges to the appropriate logonids. For details about giving logonid privileges to virtual machines, see the *Administrator Guide*.

   Besides giving the VM Batch task machine the VMD4TARG logonid privilege, assign this machine the privilege necessary to log onto your system, defined through the VMCHK operand in the OPTS VMO record.

   We strongly recommend you also assign the AUTOONLY privilege to the VM Batch task machine so it cannot be logged onto a terminal device. Do not assign the VM Batch task machine additional CA ACF2 for z/VM privileges. The only privileges that should be granted to this machine are VMD4TARG, the system VMCHK value, and AUTOONLY.

   If you do not assign the VMD4TARG privilege to the VM Batch task machine or do not assign the VMD4AUTH privilege to the VM Batch monitor machine, the following message appears on the VM Batch monitor machine console:

   ```
   003W VM Batch Facility monitor machine not set up for RACF
   surrogate operation
   ```

Examine the CLASSES and TASK keywords in the VM Batch Facility control file. If you are using the AUTOLOG class, be sure to give the VM Batch monitor machine the authority to autolog all virtual machines. There are two ways to do this:

a. Assign the AUTOALL logonid record privilege to the monitor machine. CA ACF2 for z/VM writes a resource SMF record (ACFRPTRV) whenever the monitor autologs a machine. This method lets the monitor autolog all user machines without a valid password and without requiring rules.

b. Create a CA ACF2 for z/VM logonid record for each task virtual machine and the monitor machine:

   - If you do not assign the AUTOALL attribute to the monitor machine, assign the AUTONOPW attribute to each task machine so you can autolog them without requiring the true CA ACF2 for z/VM password.
   - Make sure the PREFIX field equals the logonid. For example, PREFIX(BATCH1) for the BATCH1 task virtual machine logonid record.
   - Assign the AUTOONLY logonid record attribute to each task virtual machine. AUTOONLY lets users autolog task machines, but prevents users from actually logging onto the machine.

3. Define each VM Batch task machine strictly as a class G virtual machine in the VM directory.
4. Write resource rules that allow the monitor machine to autolog certain user machines. This controls which virtual machines the monitor can autolog. For example, to let the monitor machine autolog the USER1, USER2, and USER3 virtual machines, you might write the following rules:

```
$KEY(user1) TYPE(alg)
 UID(batch) ALLOW


$KEY(user2) TYPE(alg)
 UID(batch) ALLOW


$KEY(user3) TYPE(alg)
 UID(batch) ALLOW
```

This example assumes you specified the default of ALG in the AUTOLOG field of the RESCLASS VMO record. For additional information about writing resource rules see the *Administrator Guide*.

5. Tailor the TASK keyword in the VM Batch Facility CONTROL FILE:

   The syntax for the TASK control statement in the CONTROL file is:

   ```
   TASK userid *NOPASS* *NOPASS* .ABCD 0000 2400
   ```

   userid

   The user ID of the user attempting the link

   NOPASS

   Indicates no directory logon password is necessary

   NOPASS

   Indicates no link password is necessary to link read/write to the task machine's 191 disk.

   Note: If your task virtual machines use a 191 minidisk password (DIRMDPW=RESPECT only), be sure the WRITE password is the same as the password used in the VM directory. If you do not use 191 minidisk passwords, specify the password as *NOPASS*.

6. Write minidisk rules to let the VM Batch monitor machine access (read, write, and execute) the 191 minidisks of all the task virtual machines.

   ```
   $key(batchx)
    v0191.--uid(batch) read(a) write(a) exec(a)
   ```

7. The ACCTVLD operand of the OPTS VMO record determines if VM account validation is in effect. If you set ACCTVLD to NO, it turns off account validation; if you set ACCTVLD to FULL or LID, it turns on account validation. (For complete details on these settings, see the *Administrator Guide*. Issue the following subcommand of the ACF command to determine the ACCTVLD setting at your site:

   ```
   SHOW ACTIVE
   ```

   If the ACCTVLD operand is set to FULL or LID, you must modify the DGRUJB EXEC, DGRUCD EXEC, and the DGRBATCH EXEC.

8. The DGRUJB EXEC normally resides on the VM Batch Facility monitor machine 191 disk.

   Locate the following two lines of code:

   ```
   allow = 0;           /* Reject remote job submission */
   allow = 1;          /* Allow remote job submission  */
   ```

   Reverse their order to permit remote jobs.

   About 20 lines farther down, locate the following line:

   ```
   when called = 'RECVD' then do;
   ```

Add the following lines to enable account validation:

```
/* Following section added to support CA-ACF2 account validation.*/
   if jobacct = '*' then exitrc = 3
          else do
             'ACFBATCH' jobuser jobacct
             if rc ^= 0 then exitrc = 3
          end

/* End of CA-ACF2 section */
```

Locate the following section, located 17 lines after the position where you entered the above account validation lines:

```
/*     queue 'ALTID' jobuser jobnode jobname 'NETJOB'   */
/* -----------------------------------------------------*/
```

Insert the following lines to allow a remote user to run a job under the same logonid as the remote ID or to provide a translation outline to fill in the local logonid in the JOBALTUSER variable:

```
/* CA-ACF2 change for remote job submission */
       queue 'ALTID' jobuser jobnode jobname jobaltuser
/* End CA-ACF2 change */
```

Locate the following section:

```
/*     otherwise                                            */
/*       queue 'ALTID' jobuser jobnode jobname jobaltuser   */
/*     end                                                  */
/*  end                                                     */
/* -------------------------------------------------------- */
```

Insert the following lines to allow a remote user to run a job under the same logonid as the remote ID or to provide a translation outline to fill in the local logonid in the JOBALTUSER variable:

```
/* CA-ACF2 change for remote job submission */
       queue 'ALTID' jobuser jobnode jobname jobaltuser
/* End CA-ACF2 change */
Locate the following lines:
/* ------------------------------------------------------*/
/*                   EXAMPLE                             */
/*  Reject CHJOB command if account field not equal to   */
/*  job owners user id but allow the job to proceed.     */
/*                                                       */
/*  if Jobacct ^= Jobuser then exitrc = 4 ;              */
/* ------------------------------------------------------*/
Add the following two lines to allow account validation:
'ACFBATCH' jobuser jobacct
if rc ^= 0 then exitrc = 4
Locate the following section of code toward the bottom of the exec:
   /* ------------------------------------------------------- */
   /*                    EXAMPLE                              */
   /* Reject all jobs that do not have the account field set */
   /* equal to the job owners user id                         */
   /*                                                         */
   /*                                                         */
   /* if Jobacct ^= Jobuser then exitrc = 3;                  */
   /* ------------------------------------------------------- */
end;
```

Change it to:

```
   /* ------------------------------------------------------- */
   /*                    EXAMPLE                              */
   /* Reject all jobs that do not have the account field set */
   /* equal to the job owners user id                         */
   /*                                                         */
   /*                                                         */
   /* if Jobacct ^= Jobuser then exitrc = 3;                  */
   /* ------------------------------------------------------- */
  'ACFBATCH' jobuser jobacct
  if rc ^= 0 then exitrc = 3
end;
```

9. The DGRBATCH EXEC normally resides on the 19E Y-disk. Locate the following section near the beginning of the exec:

```
o.submit.id      = hour||min||sec;  o.submit.account  = '*';
o.submit.class   = 'A';             o.submit.seconds  = '*';
o.submit.print   = '*';             o.submit.punch    = '*';
o.submit.wbegin  = '*';             o.submit.wend     = '*';
o.submit.restart = 'YES';           o.submit.error    ='DUMP';
o.submit.pj      = '9';             o.submit.date     = date(u);
o.submit.parms   = '*';             o.submit.append   = 'NO';
o.submit.chain   = '*';             o.submit.password = '*';
```

Add the following seven lines between the first and second line so the exec reads:

```
     o.submit.id      = hour||min||sec;  o.submit.account  = '*';
Add >'MAKEBUF'
Add >'EXECIO * CP (STRING QUERY ACCOUNT'
Add >if rc = 0 then do
Add >parse pull . o.submit.account .
Add >o.submit.account = '*'o.submit.account
Add >end
Add >'DROPBUF'
     o.submit.class   = 'A';             o.submit.seconds  = '*';
     o.submit.print   = '*';             o.submit.punch    = '*';
     o.submit.wbegin  = '*';             o.submit.wend     = '*';
     o.submit.restart = 'YES';           o.submit.error    ='DUMP';
     o.submit.pj      = '9';             o.submit.date     = date(u);
     o.submit.parms   = '*';             o.submit.append   = 'NO';
     o.submit.chain   = '*';             o.submit.password = '*';
```

Further down in the same exec (about line 335), the following line occurs:

```
if acc then o.func.account = '*' || o.func.account;
```

Insert the following line immediately after it:

```
else o.func.account = '*'o.submit.account
```

10. The DGRUCD EXEC normally resides on the VM Batch Facility monitor machine 191 disk. Locate and reverse the following two lines to allow remote commands:

```
allow = 0;           /* Reject remote commands */
allow = 1;           /* Allow remote commands  */
```

Installation is complete.

VSAM AMSERV Support

CA ACF2 for z/VM provides standard CA ACF2 for z/VM validation for VSAM catalog management allocation and deallocation requests issued from programs or by the CMS AMSERV command. Installing CA ACF2 for z/VM VSAM AMSERV support involves front ending the IBM VSAM text file IGG0CLC9 with the CA ACF2 for z/VM ACFFEAMS text file. The installation follows:

1. You must install the GA version of CA ACF2 for z/VM r12 on both CP and CMS.
2. If you installed VSAM on your system and the tape contents loaded from this installation are still available, proceed to the next step. To install CA ACF2 for z/VM support for VSAM AMSERV, you need the VSAM tape contents on the system.

   If you did not install VSAM and the tape contents are not available, fully install VSAM and AMS first, without CA ACF2 for z/VM support. Follow the IBM installation procedures for VSAM and AMS. When you are ready to install CA ACF2 for z/VM VSAM AMSERV support, the tape contents loaded from the VSAM and AMS installation must remain on the system.
3. The ACFUTFEP utility front ends the IBM IGG0CLC9 text file with the CA ACF2 for z/VM ACFFEAMS text file to install CA ACF2 for z/VM VSAM AMSERV support. To run ACFUTFEP, enter the following command:

   ```
   ACFUTFEP IGG0CLC9 TEXT * ACFFEAMS TEXT *
   ```

For complete information on the ACFUTFEP utility program, see the *Report and Utilities Guide*.

See the IBM Service Guide for information on how to generate and save your VSAM segments.

Chapter 8: Troubleshooting

This chapter contains information about:

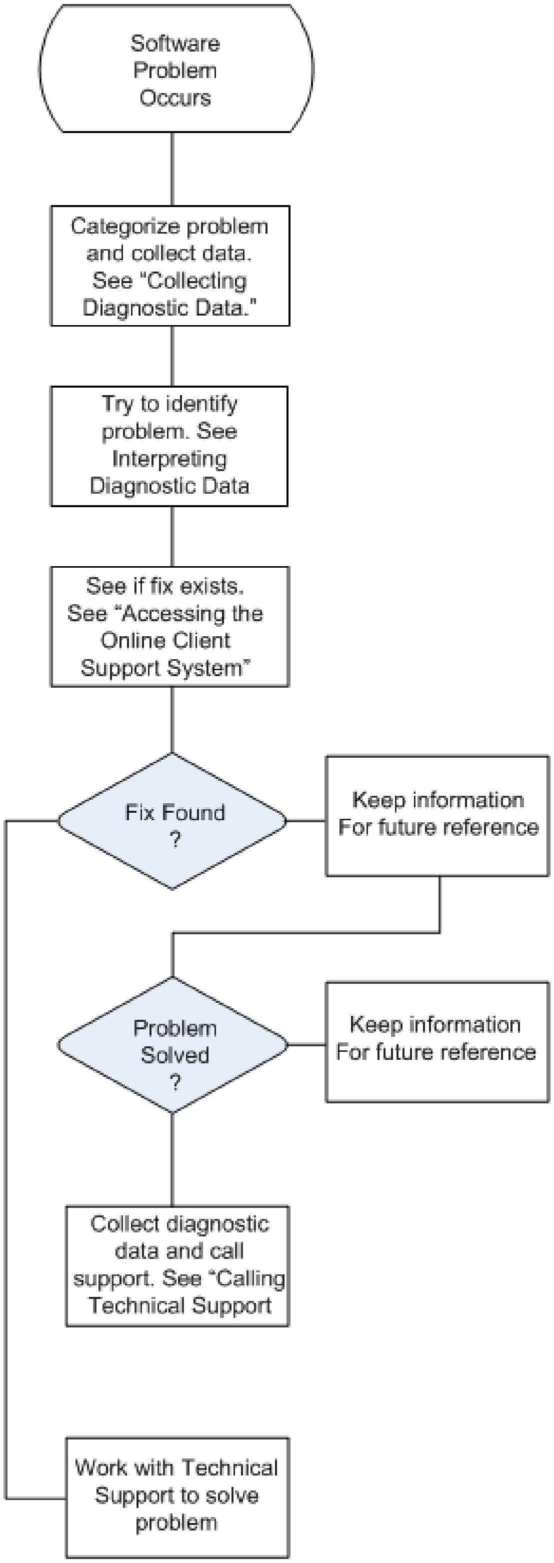
- Identifying and resolving problems
- Contacting CA Technical Support
- Receiving a new version of a product and ongoing maintenance
- Requesting product enhancements

This section contains the following topics:

Diagnostic Procedures (see page 252)
Contacting Customer Support (see page 257)
Accessing the Online Client Support System (see page 257)

Diagnostic Procedures

See the following illustration for a summary of the procedures to follow if you should encounter a problem with a CA software product. Each of these procedures is detailed on the following pages.


Software Problem Occurs
Categorize problem and collect data. See "Collecting Diagnostic Data."
Try to identify problem. See Interpreting Diagnostic Data
See if fix exists. See "Accessing the Online Client Support System"
Fix Found ?
Keep information For future reference
Problem Solved ?
Keep information For future reference
Collect diagnostic data and call support. See "Calling Technical Support
Work with Technical Support to solve problem

Incorrect Installation Symptoms

This section contains some symptoms of incorrect installation and their solutions.

Spooling Initialization is Complete

Symptom

The CA ACF2 for z/VM service machine was autologged before CA ACF2 for z/VM displayed the following message, resulting in an IUCV error:

```
HCPWRS2512I spooling initialization is complete
```

Error

The CA ACF2 for z/VM service machine is not ACF2VM and you did not specify it in the SRVMID operand of the VMXAOPTS macro.

Solution

Specify the service machine user ID in the VMXAOPTS macro. Reassemble HCPAC0 and regenerate CP.

Invalid Password Message

Symptom

After the first IPL with CA ACF2 for z/VM installed, users are getting an invalid password message when logging on with correct passwords.

Error

You did not correctly copy the Logonid database to the second level CA ACF2 for z/VM service machine. The second level machine points to the default Logonid database that CA ACF2 for z/VM shipped.

Solution

Ensure that the second level machine is accessing the correct Logonid database. If you assembled the Logonid database on the first level, delete the CA ACF2 for z/VM default Logonid database on second level and be sure to copy the Logonid database from the first level to the correct minidisk on second level.

PRG005 Error

Symptom

A PRG005 error, and possibly other error codes, occur in autologged virtual machines. Other logged on virtual machines work normally. Other symptoms of the AUTOLOG error include:

- File not found
- DMSMA104S (reason code 1) occurs when you execute the AUTOLOGGED machine XEDIT PROFILE.

Error

You probably added a local module to DMKLOG, allowing an autolog without a password.

Solution

Ensure the local mod does not bypass CA ACF2 for z/VM validation. If a local mod causes a password validation bypass for autologged machines, modify the SAVEWRK1 field in DMKLOG so the password bit (PSWRDOK) is on. (IBM turned on the other bits in the SAVEWRK1 field.) After the SAVEWRK1 field, DMKLOG branches to LOG00 and continues normal processing. Because the PSWRDOK is on, DMKAC2LG builds the minilogonid control block CA ACF2 for z/VM needs. After DMKAC2LG builds this minilogonid control block, it turns the PSWRDOK bit off.

DMSFRE161T Message Issued

Error

You applied IBM service to CMS, but did not reassemble the CMS modules with ACF2VM intercepts.

Solution

You must reassemble the CMS programs with the CA ACF2 for z/VM source code intercepts and execute the ACFGEND utility to create new modules with CA ACF2 for z/VM intercepted programs in them.

Undefined HCPAL0 and HCPAL0MC References

Error

You did not select the upper case language support and did not delete the UCENG definition in CAXALOAD.

Solution

When you comment or delete the LANG= parameter of VMXAOPTS, you must also delete the UCENG definition in CAXALOAD.

Collecting Diagnostic Data

The following information is helpful in diagnosing problems that might occur:

- Relevant system log or console listings
- Relevant system dumps or product dumps
- List of other IBM or third-party products that might be involved
- Manufacturer, model number, and capacity of your hardware
- Numbers and text of IBM or CA error messages associated with the problem
- Names of panels where the problem occurs
- Listings of all fixes applied to all relevant software, including:
  - The date fixes were applied
  - Fix numbers
  - Names of components to which fixes were applied
- Short description of problems

Interpreting Diagnostic Data

When you collect the specified diagnostic data, write down your answers to the following questions:

- What was the sequence of events before the error condition?
- What circumstances existed when the problem occurred and what action did you take?
- Has this situation occurred before? What was different then?
- Did the problem occur after a particular PTF was applied or after a new release of the software was installed?
- Have you recently installed a new release of the operating system?
- Has the hardware configuration (tape drives, disk drives, and so forth) changed?

From your response to these questions and the diagnostic data, try to identify the cause and resolve the problem.

Contacting Customer Support

For online technical assistance and a complete list of locations and phone numbers, contact Customer Support at http://ca.com/supportconnect. Customer support is available 24 hours a day, 7 days a week. For telephone assistance, call:

- U.S. and Canada 1-800-645-3042
- International (1) 631-342-4683

Accessing the Online Client Support System

CA Internet site provides a variety of information about CA products and services including:

- FAQs-documents containing the most common questions and issues experienced by our clients.
- StarTCC-Open and browse issues, DARS, technical information
- User Groups-directories, meetings, news, CA World information, new user group request and application forms, join a user group
- CA Education-learning paths, catalogs, brochures, locations/schedules, on-site services, registration

To access these any many other services go to http://supportconnect.ca.com.

Chapter 9: Field Definition Record

The CA ACF2 for z/VM Field Definition Record (ACFFDR) is a module that each site modifies and assembles to customize CA ACF2 for z/VM according to its security requirements. The ACFFDR contains parameters for specifying:

- Logonid record fields and their associated alter and list privileges
- Formatting information for the ACF command processor
- Character fields that make up the UID
- Operating parameters for CA ACF2 for z/VM
- Performance parameters for CA ACF2 for z/VM
- Privileged third party validators.

This section contains the following topics:

About the VMXAOPTS Macro (see page 259)
ACFFDR Macro Summary (see page 260)
Modify and Activate ACFFDR Changes (see page 261)
@CFDE-Create Field Definition Entry Macro (see page 262)
@CFDEDFT-Generate Default Value for Database Record Field (see page 273)
@DDSN-Dynamic Data Set Allocation Macro (see page 274)
@GNVMFDR-Field Definition Record Generator Macro (see page 275)
@GROUP-Group Names for Logonid Display (see page 275)
@HEADER-Heading Formatting for Logonid Display (see page 277)
@MLID-Logonid Compression Macro (see page 277)
@SETUP-DSECT Map Initialization Macro (see page 279)
@SMF-System Management Facility Macro (see page 279)
@SRF-System Request Facility Macro (see page 286)
@SYSID-System Identification Macro (see page 288)
@UID-User Identification String Macro (see page 289)
@VALUES-Generate Verification Values (see page 289)

About the VMXAOPTS Macro

You must specify a number of CA ACF2 for z/VM CP-based control options in the VMXAOPTS macro (or in the ACF2VM CONFIG file). You must code both the ACFFDR and the VMXAOPTS macro specifications (or ACF2VM CONFIG file statements) to generate a usable CA ACF2 for z/VM system. See the *System Programmer's Guide* for a complete description of the VMXAOPTS macro options and the ACF2VM CONFIG file.

ACFFDR Macro Summary

We summarize the ACFFDR macros below and describe them in more detail throughout the rest of this chapter.

@CFDE

Defines the name of each field in the logonid record and the associated parameters that describe each field.

@CFDEDFT

Generates a default value for a database record field.

@DDSN

Specifies the names of the files that make up each group of CA ACF2 for z/VM databases.

@GNVMFDR

Generates the CA ACF2 for z/VM Field Definition Record.

@GROUP

Defines group names for fields in the logonid record.

@HEADER

Specifies what fields appear on the first line of the logonid display by the ACF command.

@MLID

Specifies a logonid compression algorithm CA ACF2 for z/VM uses to conserve storage space.

@SETUP

Initializes assembly equates and maps necessary DSECTs.

@SMF

Specifies the individual record numbers that CA ACF2 for z/VM assigns to the various CA ACF2 for z/VM SMF records. It also contains site-defined configuration and performance options SMF recording uses.

@SRF

Defines the logonids that can issue CA ACF2 for z/VM System Request Facility (SRF) calls.

@SYSID

Defines the default startup SYSID string.

@UID

Lists the fields that make up the UID.

@VALUES

Used in an RSB module to generate field verification values.

Modify and Activate ACFFDR Changes

To modify the ACFFDR using CA-ACTIVATOR, refer to Task M9C0I023 for detailed information. The instructions below outline the proper sequence of steps to follow when activating ACFFDR changes in CMS. Read all of the instructions carefully before activating a new ACFFDR. We recommend you do not activate ACFFDR changes during periods of heavy system activity.

To modify and activate ACFFDR option changes:

1. Use CA-ACTIVATOR to modify and reassemble the ACFFDR. The default ACFFDR resides in ACFFDR ASSEMBLE. See the “Default Field Definition Record” appendix for the default ACFFDR listing.

2. After reassembling the ACFFDR, move the new ACFFDR text file to the appropriate CA-ACF2 service machine read-only minidisk. CA ACF2 for z/VM reaccesses all read-only minidisks before it reloads the ACFFDR. CA ACF2 for z/VM locates the text file through standard CMS search conventions.

   To move the ACFFDR, follow these steps:

   ■ Enter the following command to erase the old ACFFDR:

   ```
   ERASE ACFFDR TXTOLD fm
   ```

   fm

   The filemode of the disk where the ACFFDR resides.

   ■ Enter the following command to rename the current ACFFDR so it becomes your backup copy:

   ```
   RENAME ACFFDR TEXT fm = TXTOLD fm
   ```

   fm

   The filemode of the disk where the ACFFDR resides.

   – Enter the following command to copy the ACFFDR:

   ```
   COPYFILE ACFFDR TEXT A = = fm
   ```

   fm

   The filemode where you want the ACFFDR to resiEnter the following CP command to reload the new version into storage:

   ```
   ACFSERVE RELOAD FDR
   ```

   See the *Administrator Guide* contains for information about the ACFSERVE commands.

The following chart represents a brief overview of activating specific ACFFDR options:

ACFFDR Macro	ACFSERVE RELOAD	ACFSERVE RESTART	IPL System
@CFDE	X		
@DDSN		X	
@GROUP	X		
@HEADER	X		
@MLID	X		X
@SMF		X	
@SRF	X		
@SYSID	X		

CA ACF2 for z/VM only uses the @SYSID during a system IPL. To change the current SYSID, enter the following command:

```
ACFSERVE RELOAD CONTROL(VMO)
```

We do not recommend refreshing the @UID macro of ACFFDR.

– Issue the appropriate ACF SHOW subcommand to verify that your changes are active.

@CFDE-Create Field Definition Entry Macro

The @CFDE macro defines an external field name and its related internal characteristics and attributes for a field contained in a structured CA ACF2 for z/VM record, such as a logonid record or a record structure block (RSB) module. CA reserves some @CFDE operands for internal use.

The AUTH, ALTER, and LIST authorization operands assume basic access to the logonid record. Privileges such as SECURITY or ACCOUNT, defined in the requester's logonid record and associated SCPLIST value, determine a user's access. CA ACF2 for z/VM verifies access to the logonid record before, and independently of, field-level access controls. After CA ACF2 for z/VM grants access to the logonid record, it verifies individual field authorizations (AUTH, ALTER, and LIST). Logonid record access privileges supersede any logonid field-level controls.

The syntax of the @CFDE macro is:

```
@CFDE   name,symbol,type,AUTH=fieldname,ALTER=0|list,
        LIST=0|list,FLAGS=0|list,BITMAP=0|bitmap,
        PRTN=0|nn,RRTN=0|nn,GROUP=0|nn,MVFLAGS=0,
        MVMIN=0|min,MVMAX=0|max,VRTN1=0|num,
        VRTN2=0|addr,VPRM1=0|addr,VPRM2=0|addr,
        DFTAD=0|addr,DFT=0,STATUS=0,INFOFLG=0,
        INFOCLS=0,CFDENME=0,ZERO=NO|YES,PROMPT=NO|YES,
        TRIM=YES|NO,VER=0,XTYPE=0,XSYMBOL=0,COUPLE=0,
        COUPTYP=0,CBPROC=NO|YES,COUNTER=NO|YES
```

name

Specifies the external logonid field name. See the *Administrator Guide* for a list of CA ACF2 for z/VM-supplied fields. For RSBs, name specifies the external name for a field&mdash.the name the ACF command refers to. It can be from one to eight characters and contain any characters valid in an assembler character constant. If it contains any special characters, you must enclose them in single quotes. If it contains single quotes or ampersands, you must double them, as in standard assembler character constant practice.

symbol

Specifies the symbolic label assigned to the logonid field in the LIDREC DSECT. For RSBs, symbol specifies the label on the field in the mapping DSECT that describes the infostorage record.

type

Specifies the field type. Valid types are:

BINARY

A one- to four-byte binary field.

BIT

A bit field used as a switch or flag.

CHAR

A text field of one to 255 bytes.

CHEN

A four-byte encoded character field (password). For RSBs, CHEN specifies an encrypted character field up to 255 bytes long.

HEX

A one- to 255-byte hexadecimal field.

PACKED

An eight-byte EBCDIC date field.

TIMEBIN

A four-byte binary format time, expressed in units of 0.01 seconds past midnight.

TOD

An eight-byte time stamp in number of microseconds past January 1, 1900 (store clock instruction). For RSBs, you can create your own processing routine to convert this field to the format stored on the CA ACF2 for z/VM database. To convert to display format, you can use the CA ACF2 for z/VM reconstruction routine.

AUTH=fieldname

Specifies the external name of a bit field in a user's logonid record that lets him alter this field. The external field name must refer to an CA ACF2 for z/VM-supplied bit field or a user-defined bit field name. CA ACF2 for z/VM checks for AUTH validation after it checks for ALTER validation.

ALTER=0|list or LIST=0|list

Specifies privileges that can modify (ALTER) or display (LIST) this logonid field. Select multiple entries using a plus (+) sign. For example, ALTER=SECURITY+ACCOUNT indicates that any user having the SECURITY or ACCOUNT privilege can alter this field. A dash (-) limits the ALTER or LIST operands. For example, LIST=ALL-USER indicates that all requesters can list the field, except those with the USER privilege only. The ALTER and LIST operands have no default values. If you omit them, CA ACF2 for z/VM does not allow list or alter accesses. The privileges that you can specify are:

ACCOUNT

Account manager

ALL

All of these privileges

AUDIT

Auditor

CONSULT

Consultant

LEADER

Project leader

SECURITY

Security administrator

USER

Normal user

FLAGS=0|list

Provides a set of special field handling options that we describe below. Use a plus sign (+) to separate multiple options. The options are:

HUNDRED

For binary fields, the internal form is .01 units, while the external form is in units (ones).

If you are using the database synchronization component, you must also code the CBPROC=YES operand.

LIMIT

Do not return this field to requesters suppressing trivial fields. The SET NOTRIVIA subcommand of the ACF command supports LIMIT. This flag indicates CA ACF2 for z/VM displays the field only when you request the entire logonid record (that is, all fields in the logonid record).

MULTIVAL

This field supports multiple values.

MUTEXC

All bits in the byte are mutually exclusive and zeros all bits in the byte before turning any on. This is for bit fields only.

NEVER

Never return this field in response to a formatted retrieval request. Do not print this logonid field when you issue the LIST command.

NULL

If you do not specify this field, CA ACF2 for z/VM does not display the field name or value.

RESTRICT

Only unrestricted security administrators can change this field, such as users with the SECURITY privilege and no SCPLIST restrictions specified in their logonid record.

SPECIAL

Indicates that the ACF command bypasses validity checking of an input field value. The ACF command normally checks the length of character information and the maximum size of binary information. Use this option when a processing or validation routine will process this field value.

BITMAP=0|bitmap

Indicates the bit pattern that represents the on condition for bit fields. This is a one-byte value with a single bit set (for example, X'20'). You must represent the on condition by a 1 bit; you cannot have a flag that is on when its bit is off.

For RSBs, BITMAP applies only to fields with a type of BIT. It indicates the bit configuration that represents the particular bit flag in a byte. For example, suppose there was a byte containing two bit flags, defined like this in the mapping DSECT for the structured infostorage record:

```
FLAGBYTE DS  X   A BYTE OF BIT FLAGS
BITFLAG1 EQU  X'80' ...THE FIRST BIT FLAG
BITFLAG2 EQU  X'40' ...THE SECOND BIT FLAG
```

The @CFDE macros for these two flags look like this:

```
@CFDE FLAG1,FLAGBYTE,BIT,BITMAP=BITFLAG1,...
@CFDE FLAG2,FLAGBYTE,BIT,BITMAP=BITFLAG2,...
```

PRTN=0|nn

Indicates the processing routine ID. This indicates which routine CA ACF2 for z/VM uses to convert the input data entered using the CHANGE or INSERT commands to the proper format for storage in the CA ACF2 for z/VM database. You do not need to specify a processing routine ID for standard CA ACF2 for z/VM data types. See the *System Programmer's Guide* for information about supporting user-written processing routines.

RRTN=0|nn

Indicates the reconstruction routine ID. This indicates the routine CA ACF2 for z/VM uses to convert data stored in the database into display format for the LIST command output. You do not need to specify a reconstruction routine ID for standard CA ACF2 for z/VM data types. The CA ACF2 for z/VM-supplied processing and reconstruction routines are:

RTNID	PROCESS (PRTN)	RECONSTRUCT (RRTN)	Notes
0	Null	Null	1
1	Character	Character	
2	Packed	Packed	
3	Switch	Switch	
4	Binary	Binary	
5	Password	TOD	
	Cancel/Suspend	Construct UID	
	Line/Attn	Line/Attn	
9	Char		2

RTNID	PROCESS (PRTN)	RECONSTRUCT (RRTN)	Notes
10	Char(mask)	Char(mask)	3
11	Hexadecimal	Hexadecimal	4
12	Data encryption		5
13	Prevent/Log/Allow	Prevent/Log/Allow	6
31	16-bit field processing	Storage zie	
32	Special field replace	Right-justified hexadecimal	
33		Multistring fields	
34		One-byte field conversion	

See the *System Programmer's Guide* for information about service machine access of user-written routines.

Note the following:

- If you specify PRTN= (no processing routine) or RRTN= (no reconstruction routine), CA ACF2 for z/VM determines the appropriate routine from the TYPE= operand. If you specify a routine ID of zero (PRTN=0 or RRTN=0), you cannot use ACF subcommands to alter or display the field.
- The character field must be in the SCPLIST of the logonid that initiates the action ACF accepts.
- These routines assign the specified string to the designated field. They also expand a trailing dash to all asterisks and, upon reconstruction, reverse the process.
- These routines replace the hexadecimal routine 8 that was available in previous releases. User @CFDE macros must use routine 11.
- This processing routine uses the CA ACF2 for z/VM Data Encryption Routine ACF00ENC. This is an XDES, two-step, irreversible encryption routine. The eight-byte key that creates the cipher is the external @CFDE name used for the FDE.
- These routines process fields with acceptable values of PREVENT, LOG, and ALLOW. CA ACF2 for z/VM stores them internally as P, L, and A.
- The CA ACF2 for z/VM system configuration supports processing and reconstruction routines with IDs from one to 98. If you add your own routines, use the IDs 17 through 30 to avoid conflicts with future CA ACF2 for z/VM development.

GROUP=0|nn

Identifies the display group where CA ACF2 for z/VM is to format the output. See the @GROUP-Group Names for Logonid Display section for the GROUP names.

MVFLAGS=0

Identifies an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

MVMIN=0|nn

Specifies the minimum number of values the field can hold for a multivalued field.

MVMAX=1|max

Specifies the maximum number of values thefield can hold for a multivalued field. This is usually the same as the value specified for MAX on the ANULTFLD macro for the field.

VRTN1=0|num

Identifies an CA ACF2 for z/VM validation routine that verifies that data entered in a record field is appropriate for that field. Validation routines obtain control before the processing routine described previously. This operand is optional. You do not have to specify it for user-defined fields. The VRTN1 validation routines supplied with CA ACF2 for z/VM include:

0

Requires no additional validation. This is the default.

1

Validates the source for a character-field replacement. The source field must be a valid program or data set index-level name that is one- to eight-characters long. The first character cannot be numeric; the remaining ones can be alphanumeric or national. A blank string is valid.

2

Validates a source character field. It checks for a valid OS/VS data set name. It does not support GDG relative version numbers and PDS member names.

3

Matches a character value to a list value. The @VALUES macro defines this list.

4

Validates a binary value range.

5

Ensures that a user cannot perform a logonid record PASSWORD update until the MINDAYS interval contained in the logonid record expired.

6

Validates the key mapping operand.

8

Validates the full-field specification.

11

Validates character fields. The services available include:

- Validation of minimum and maximum data length
- Rejecting data with embedded blanks
- Validating authority fields against the ACFFDR.

16

Validates bit set fields.

31

Validates storage size fields and converts them to a binary value that CA ACF2 for z/VM passes to ACSPR31P to store into the record.

32

Validates and right-justifies a hexadecimal value. Validation options include:

- The minimum and maximum number of hexadecimal digits
- Null field processing options
- Valid ranges of the data.

CA ACF2 for z/VM passes the resulting value to the ACSPR32P processing routine.

33

Validates multistring fields and converts these strings to fixed length subfields. Validation includes:

- Validates the length of each substring
- Validates the minimum length of each substring
- Ensures that a specified portion of the field does not match any entry in an exclude list. CA ACF2 for z/VM passes the resulting value to the ACSPR32P processing routine.

34

Converts character input to a one-byte equivalent. CA ACF2 for z/VM converts the input character string to a one-byte equivalent. CA ACF2 for z/VM passes this converted value to the ACSPR32P processing routine.

See the *System Programmer's Guide* for information about service machine access of user-written routines.

VRTN2=0|num

Identifies an CA ACF2 for z/VM validation routine that verifies that the data entered into a record field is appropriate for that field. This operand is optional. You do not need to specify it for user-defined fields. VRTN2 validation routines supplied with CA ACF2 for z/VM include:

0

Requires no additional validation. This is the default.

14

Verifies that the value given for a two-byte or a four-byte binary field is not negative.

16

Validates the full field for bit set fields.

33

Sorts multivalued field entries and checks for duplicate entries. You can specify if CA ACF2 for z/VM should check the sort and duplication for only a portion of each entry.

See the *System Programmer's Guide* for information about supporting user-written validation routines for RSB functions.

VPRM1=0|addr

Specifies a parameter passed to the VRTN1 field validation routine. It points to a value list the @VALUES macro generates. Specify the label the @VALUES macro uses for *addr*.

VPRM2=0|addr

Specifies a parameter to pass to the VRTN2 field validation routine.

DFTAD=0|addr

Specifies the address of a remote default value the @CFDEDFT macro generates. Specify the label the @CFDEDFT macro uses for *addr*.

DFT=0

Specifies the default value for the field when you use the INSERT subcommand to insert the record. The default for each type of field is:

BINARY

0

BIT

RESET

CHAR

Blanks

CHEN

Blanks

HEX

0

PACKED

0

TIMEBIN

0

TOD

0

The DFT and DFTAD fields are mutually exclusive. Use DFTAD to specify the address of a remote default value that the @CFDEDFT macro generates.

STATUS=0

Indicates an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

INFOFLG=0

Indicates an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

INFOCLS=0

Indicates an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

CFDENME=0

Indicates an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

ZERO=NO|YES

The default of ZERO=NO indicates that CA ACF2 for z/VM copies the model record field when you issue the INSERT USING subcommand. When you indicate ZERO=YES, CA ACF2 for z/VM does not copy the field.

PROMPT=NO|YES

Indicates whether CA ACF2 for z/VM prompts for the value of a field and you enter it in a nondisplay protected area. This is useful for fields where you add or change sensitive information, such as a password. The default is PROMPT=NO. If you specify PROMPT=YES and the field name is PASSWORD, CA ACF2 for z/VM issues the following message to prompt you for the password:

```
ACFpgm256R Enter new CA-ACF2 password
```

You must then enter a new password for the logonid you specified in the CHANGE subcommand. CA ACF2 for z/VM then prompts you to reenter the password for verification. If you are changing a logonid, you cannot enter a value for PASSWORD. If you specify PROMPT=YES and the field name is not password, the following message prompts you for the value of this field:

```
ACFpgm799R Enter <fieldname> value:
```

TRIM=YES|NO

Indicates whether CA ACF2 for z/VM should remove trailing blanks from character fields or zeros from hex fields when it displays the fields. The default value, TRIM=YES, removes trailing blanks or zeros.

VER=0|fn

Provides a one-byte binary area to identify the version ID of an infostorage record.

XTYPE=0

Indicates an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

XSYMBOL=0

Indicates an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

COUPLE=0

Indicates an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

COUPTYP=0

Indicates an CA ACF2 for z/VM internal validation operand. Do not code a value for it.

CBPROC=NO|YES

Indicates whether authorized programs can bypass processing routines when updating the field. CA ACF2 for z/VM internal processing uses this operand. Do not code a value for it.

COUNTER=NO|YES

Specifies whether a binary field is a counter. You update counter fields by addition or subtraction, rather than replacement. This ensures that you do not lose counts when you make multiple, almost-concurrent updates. You cannot use the VRTN1 field validation routine three with counter fields.

CA ACF2 for z/VM requires the @CFDE macro for the supplied CA ACF2 for z/VM entries. It is optional for the addition of user fields. You can specify a total of 2048 @CFDE macros. If you add fields to the logonid record, you must also modify the USERLID or USERXLID COPY files to define the size and location of each additional field in the logonid record. The USERLID and USERXLID COPY files contain comments indicating where you should place these new fields. These COPY files are located in the ACF2USER MACLIB. You can also refer to installation step M9C0I021 for more information on modifying the USERLID and USERXLID COPY files.

The LIDREC DSECT contains all CA ACF2 for z/VM-defined fields in the logonid record.

@CFDEDFT-Generate Default Value for Database Record Field

The @CFDEDFT macro generates a default value or series of values for a database record field defined through a Field Definition Entry. Single valued fields have a single default value. This value can be generated inline in the FDE or in a remote area. Multivalued field default lists must be generated remotely.   The syntax for the @CFDEDFT macro is:

```
@CFDEDFT OPS,TYPE=,FLAGS=,MULTVAL=NO,MAXLEN=
```

OPS

Specifies one or more values appropriate for the field as defined by TYPE. Multiple values are permitted only when MULTVAL=YES and only if the list is generated out-of-line with respect to the FDE. If omitted, an appropriate 'zero value' is generated.

TYPE

Specifies a standard Field Definition Entry data type specification (for example, binary). This operand is required.

FLAGS

This value is coded on the @CFDE macro defining this field.

MULTVAL

Specifies whether a list of defaults is provided supporting a multivalued FDE. The default is MULTVAL=NO.

MAXLEN

For internal use only. Do not use.

@DDSN-Dynamic Data Set Allocation Macro

The @DDSN macro specifies the CMS filenames for each set of CA ACF2 for z/VM databases. With CA ACF2 for z/VM database sharing, this macro specifies the names of the VSE/VSAM databases.   This macro dynamically allocates the group of CA ACF2 for z/VM databases used at system startup time. To use this macro, enter the following command:

```
DDSN(name)
```

After the following message appears:

```
ACFpgm000R: Enter CA-ACF2 startup options or press enter
```

Specify this system startup option with the name of the CA ACF2 for z/VM database group, defined in the ACFFDR by the name operand. If you do not specify a group name at system startup time, CA ACF2 for z/VM uses the first group of CA ACF2 for z/VM databases specified in the @DDSN macro as the default.

The syntax of the @DDSN macro is:

```
@DDSN  name,                    DEFAULT GROUP INDEX
       RULE=filename,           RULES CLUSTER
       LID=filename,            LOGONID CLUSTER
       INFO=filename            GEN RESOURCE CLIST
```

A set of operands and corresponding values must exist for each set of defined CA ACF2 for z/VM databases

name

The one- to eight-character name of the CMS file or VSE/VSAM data set allocation group. The default is PRIMARY. We also supply a default group named BACKUP.

RULE

The one- to eight-character CMS filename (specified in single quotes) for the Rule database. The default is RULES for the PRIMARY group and BKRULES for the BACKUP group. If you are using VSE/VSAM data sets, specify the full data set name in single quotes (for example, RULE='ACF2.VSAM.RULES'). For VSAM databases, be sure to specify more than one data set name level.

LID

The one- to eight-character CMS filename (specified in single quotes) for the Logonid database. The default is LID for the PRIMARY group and BKLID for the BACKUP group. If you are using VSE/VSAM data sets, specify the full data set name in single quotes (for example, LID='ACF2.VSAM.LID'). For VSAM databases, be sure to specify more than one data set name level.

INFO

The one- to eight-character CMS filename (specified in single quotes) for the Infostorage database. The default is INFO for the PRIMARY group and BKINFO for the BACKUP group. If you are using VSE/VSAM data sets, specify the full data set name in single quotes (for example, INFO='ACF2.VSAM.INFO'). For VSAM databases, be sure to specify more than one data set name level.

The following ACF subcommand command displays the values specified in the @DDSN macro:

```
SHOW DDSNS
```

SHOW DDSNS also displays the set of files that are currently in use. You must specify at least one @DDSN specification. There is a maximum of 16 specifications per ACFFDR.

@GNVMFDR-Field Definition Record Generator Macro

This macro creates the ACFFDR CSECT. It must be last in the assembly, just before the END statement. The syntax of the @GNVMFDR macro is:

```
@GNVMFDR
```

@GROUP-Group Names for Logonid Display

This macro creates a name for a group of logonid record fields ACF command displays. The syntax of the @GROUP macro is:

```
@GROUP text
```

text

Any text string up to 30 characters enclosed in quotes.

The SHOW FIELDS subcommand of the ACF command displays the external names of the logonid record fields sorted under the appropriate group name. The @CFDE macro GROUP=operand specifies the group where a particular logonid record field belongs. Group 0 is the default group in the @CFDE macro. You cannot define Group 0 with an @GROUP macro. You must specify the twelve group entries supplied with the distributed ACFFDR. Additional ones are optional. You can define up to 50 site-defined group statements. For example, to create a group 13 category for logonid fields, place an additional @GROUP with the desired name after the supplied @GROUPs in the ACFFDR. Then, CA ACF2 for z/VM displays any field defined with GROUP=13 in its @CFDE macro under the group 13 title when listed. To define a group number of 15, you must define a dummy group 13 and 14, using dummy text.

The groups supplied with the default ACFFDR are:

Group ID	Group Name
1	CANCEL/SUSPEND
2	PRIVILEGES
3	ACCESS
4	PASSWORD
5	TSO
6	STATISTICS
7	CICS
8	IMS
9	IDMS
10	MUSASS
11	RESTRICTIONS
12	DFP

You cannot assign a group name to Group 0. The identification section of a logonid record display refers to Group 0. See the @HEADER macro section in this chapter for more information about the logonid record display and Group 0.

@HEADER-Heading Formatting for Logonid Display

The @HEADER macro specifies the content and formatting of the IDENTIFICATION section of a logonid record display. When SET TERSE is in effect, only those fields this macro defines are displayed. SET TERSE is in effect.   The syntax of the @HEADER macro is:

```
@HEADER field,...,field
```

field

The names of the logonid fields that are to become the first lines of a logonid record display. You can only specify character fields.

The contents of the first specified logonid field become the identification section's group name. The contents of the other specified fields are listed in the order specified and without their identifying field names in the identification section of the display. The Identification section can also contain fields with GROUP=0 in their defining @CFDE macro that are not also specified in the @HEADER macro. These additional fields follow those in the @HEADER macro, but are not displayed when SET TERSE is in effect. The default ACFFDR does not have any displayable fields in Group 0 that are not also specified in the @HEADER macro.   All fields specified in the @HEADER macro are only listed in the Identification section. They are not displayed in the section corresponding to the group specified in the @CFDE macro, even if the specified group is other than Group 0. The default ACFFDR only specifies Group 0 fields in the @HEADER macro.   You can only use one @HEADER macro. The default ACFFDR specifies LID, UID, NAME, and PHONE.

@MLID-Logonid Compression Macro

The @MLID macro specifies a logonid compression algorithm CA ACF2 for z/VM uses to conserve storage space. Logonid compression eliminates unused or unnecessary information from the resident copy of the logonid record.   There are three sections to a minilogonid:

- CA ACF2 for z/VM required fields (called the BASE section)
- CA ACF2 for z/VM required fields (called the VM section)
- Site-dependent fields.

Site-dependent minilogonids do not need to include those fields predefined under BASE and VM because the minilogonid construction routines include them. See the *System Programmer's Guide* for information about minilogonid requirements. The syntax for the @MLID macro is:

```
@MLID           id,base,tleng,(maxfield,minfield),...,(maxfield,minfield),
$LASTPRM
```

id

The ID assigned to this compression algorithm. This is a one to eight character valid assembler label. CA ACF2 for z/VM reserves the names BASE and VM. They define the minimum minilogonid requirements for CA ACF2 for z/VM system processing. However, you can add logonid record fields at the end of the VM minilogonid.

base

The label for a DSECT statement that defines the minilogonid section. The base and id operands cannot be the same.

Tleng

A symbol equated to the total length of the minilogonid section.

(maxfield,minfield)

Lets you can take any number of fields from the full logonid and include them in the minilogonid. You must define each field using two symbols; maxfield is the symbol operand of the @CFDE macro that specifies a field in the full logonid record and minfield defines the associated field name in the minilogonid record. These parameters are the assembler symbols in the associated DSECTs for each field. See the Default Field Definition Record appendix to see the shipped @MLID fields.

$LASTPRM

Checks the syntax for additional positional parms and missing commas. This operand is required and must always be the last operand of the @MLID macro.

You must supply BASE and VM definitions. Site @MLID entries are optional in the ACFFDR. For each @MLID specification, you must provide an assembler DSECT to map the storage area offset and symbol length. You do not need to specify field information in a specific order in the @MLID. CA ACF2 for z/VM constructs the minilogonid based on the assembler DSECT specification, so changing the @MLID order has no effect.

The length of the associated fields, as defined in the DSECT definition, must match exactly between the full logonid and the minilogonid record. The total length of all the fields in the minilogonid must be equal to the *tleng* specification. The @MLID specification must completely define all padding space. You must associate it with an equivalent area in the logonid record. There is no need for a one-to-one correspondence between fields coded as @MLID operands and DSECT labels. You only need to code the DSECT labels that define the field length attributes and displacements in the DSECT. You do not need to redefine those areas in the @MLID operands.

The minilogonid record includes all fields that CA ACF2 for z/VM processing requires. The BASE and VM minilogonid definitions include fields that are minimally required to control a user's activities in regard to CA ACF2 for z/VM system processing requirements. You must define fields in the logonid record that the installation exit processing in each individual @MLID specification needs. Failure to properly define and maintain these definitions can impact the integrity of the CA ACF2 for z/VM system. If you modify the BASE or VM minilogonid definitions to define additional fields, be sure to reassemble HCPAC0 after you copy the new MLAVM macro into the ACF2VM LOADLIB.

The default ACFFDR defines three @MLID specifications. They are BASE, VM, and VSE. VM and VSE sites require the BASE @MLID specifications. The BASE and VM @MLID macros include all the logonid record fields that VM sites require for processing. Do not omit the BASE and VM @MLID macros from your ACFFDR. The BASE and VSE @MLID provide support for CA ACF2 for z/VM VSE.

@SETUP-DSECT Map Initialization Macro

The @SETUP macro expands the DSECTs and necessary equates for an CA ACF2 for z/VM Field Definition Record Generation. The syntax of this macro is:

```
@SETUP
```

@SMF-System Management Facility Macro

The @SMF macro specifies SMF processing options, SMF record number performance, and integrity options. You can format SMF record numbers into reports with the CA ACF2 for z/VM report generators. See the *Reports and Utilities Guide* for information about SMF records. Do not change SMF record numbers after installation. Do not specify an SMF record number greater than 255.

If you are upgrading from a previous CA ACF2 for z/VM release, **do not use** the numbers defined in the PSWD, DSN, LID, RULE, INFO, RSRC, ACFSERVE, CMDLIM, and DIRM operands to write SMF records. CA ACF2 for z/VM post-release 3.0 systems writes all SMF records using the number specified in the ACF2= operand.

For detailed information on the SMF options, refer to the explanations of each operand on the following pages. You must specify one @SMF macro per ACFFDR. This macro is required. You must maintain it.

The syntax of the @SMF macro is:

```
@SMF     DESTID=MAINT|userid,               DESTINATION ID DUMP|NOTIFY
         TIME=(00:00|t1,...,t16),            AUTOMATIC SWITCH TOD LIST
         SWITCH=DUMP|NOTIFY,               DUMP|NOTIFY AT FILE SWITCH
         DISKLST=(200,201,202|cuu,...,cuu),   LIST OF SMF DISKS
         CHECKPT=20,                         UPDATE HEADER EVERY N WRITES
         TIMER=15,                           UPDATE HEADER EVERY N MINUTES
         MAXQUE=10,                          MAX QUEUED BLOCKIO WRITES
         PRIOR=NO|YES,                      SMF WRITE HIGH PRIORITY
         DUMPINT=1|nnn|FILE,                 SMF DUMP PUNCH INTERVAL
         SMFID=string|null,                   COMBINED SMF RECORD
         PSWD=0|nnn,                         (Default was 220 for pre-3.3 releases)
         DSN=0|nnn,                          (Default was 221 for pre-3.3 releases)
         LID=0|nnn,                          (Default was 222 for pre-3.3 releases)
         RULE=0|nnn,                         (Default was 223 for pre-3.3 releases)
         INFO=0|nnn,                         (Default was 226 for pre-3.3 releases)
         RSRC=0|nnn,                         (Default was 227 for pre-3.3 releases)
         ACFSERVE=0|nnn,                     (Default was 229 for pre-3.3 releases)
         CMDLIM=0|nnn,                       (Default was 231 for pre-3.3 releases)
         DIRM=0|nnn,                         (Default was 232 for pre-3.3 releases)
         ACF2=230|nnn,                       COMBINED SMF RECORD
         LASTPRM=YES                         CHECK SYNTAX
```

DESTID=MAINT|userid

Specifies a valid VM user ID set up to receive SMF data or notification that an SMF file is ready to process. CA ACF2 for z/VM sends data to this ID in SENDFILE format. The SWITCH operand of this macro determines if the DESTID receives an entire SMF file or a notification file.

MAINT

Specifies MAINT is the user ID that receives SMF data or notification. This is the default value for this operand.

userid

Specifies a site-defined virtual machine designated to receive SMF data.

TIME=00:00|t1,...,t16)

Specifies the times of day that CA ACF2 for z/VM automatically initiates an SMF switch. You can specify up to sixteen time of day values in 24-hour format. Separate these times with commas.

00:00

Specifies that SMF files are switched automatically at the first SMF write request after midnight. This is the default value for this operand. Specifying only TIME=00:00 prevents an automatic switch at any other time.

t1,...,t16

Specifies a site-defined time. You can specify up to 16 time of day values for this operand. Separate these times with commas.

SMF also switches automatically at midnight if you do not indicate you want to close out the day's activity. You must reload the ACFFDR to make this change effective. See the *Administrator Guide* for information on reloading the ACFFDR.

SWITCH=<u>DUMP</u>|NOTIFY

Specifies the disposition of SWITCH processing of a SMF file. The SMF SWITCH occurs at file full time, automatically (specified in the TIME= operand) or through the following command:

```
ACFSERVE SWITCH SMF
```

DUMP

Specifies that CA ACF2 for z/VM sends a SMF dump file to the DESTID when it switches SMF files. DUMP is the default.

NOTIFY

Specifies that CA ACF2 for z/VM sends a notification file to the DESTID when it switches SMF files. Using the NOTIFY option improves the throughput of the CA-ACF2 service machine when there are a large number of SMF records. For information regarding the format of the NOTIFY file and processing of SMF data, see the *Report and Utilties Guide*.

DISKLST(<u>200,201,202</u>|cuu,...,cuu)

Specifies a list of minidisk addresses for recording SMF data. You must CMS reserve these minidisks. This list is a circular chain that allows rotation between listed multiple disks. The more minidisks listed, the greater the amount of historical datan CA ACF2 for z/VM maintains. If CA ACF2 for z/VM cannot find the minidisk or a device error occurs, it issues an error message and tries to use the next minidisk in the list.

200,201,202

Specifies that CA ACF2 for z/VM records SMF data on the CA-ACF2 service machine 200, 201, and 202 minidisks. This is the default value.

cuu,...,cuu

Specifies that CA ACF2 for z/VM records SMF data using site-defined minidisk addresses. You can specify a maximum of 32 minidisks. You must specify at least two minidisks, but we recommend you specify three. Designate one disk as the active disk, where CA ACF2 for z/VM is currently recording data. CA ACF2 for z/VM reserves any other disks listed for historical SMF data, unless otherwise specified. Disk address 191 is not valid. See the *Administrator Guide* for additional information about historical SMF disks.

CHECKPT=20

Specifies the number of SMF data blocks CA ACF2 for z/VM writes between each update of the SMF header block. This header block contains checkpoint information, such as the block number of the last data block written. You can specify any number from 0 to 255. The default value is 20. This operand applies only when you specify the PRIOR=NO value; it does not apply when you specify PRIOR=YES. This option affects the performance of your system.

TIMER=15

Specifies the maximum amount of time between each update of the SMF header block and write of the current SMF block. This operand ensures CA ACF2 for z/VM only writes SMF records when the system is not busy. Specify the time interval in minutes, from 0 to 255. The default value is 15. This operand applies only when you specify the PRIOR=NO value; it does not apply when you specify PRIOR=YES. This option affects the performance of your system.

MAXQUE=10

Specifies the maximum number of outstanding SMF write requests CA ACF2 for z/VM can queue at any time. When it reaches MAXQUE, SMF writes records through CMS I/O until the BLOCKIO requests are below MAXQUE. You can specify any number from 0 to 255. The default value is 10. This operand applies only when you specify the PRIOR=NO value; it does not apply when you specify PRIOR=YES. This option affects the performance of your system.

PRIOR=NO|YES

Specifies whether to give SMF recording priority over all other CA ACF2 for z/VM activity. If you change the setting of PRIOR while the CA-ACF2 service machine is active, you must restart the service machine to activate the change.

NO

Specifies SMF recording does not have priority. If you specify NO, SMF records through BLOCKIO. NO is the default.

YES

Writes all SMF I/O through CMS I/O. Recording SMF data has the highest priority. It also restricts the number of SMF records queued up for processing, eliminating the loss of audit data in the event of a system failure. When you specify PRIOR=YES, there may be a decrease in throughput of CA ACF2 for z/VM validations, especially if SMF is creating large numbers of records. All sites running in a C2 configuration must run CA ACF2 for z/VM with this value set to YES.

DUMPINT=1|nnn|FILE

Specifies the number of SMF records punched in one interval by SMF DUMP processing. Specifying a low number maximizes CA-ACF2 service machine throughput, but increases the amount of time needed to perform the dump. A high number reduces throughput, but decreases the amount of time needed to perform the dump. If you set SWITCH=NOTIFY in this macro, this operand does not apply.

1

Punches one SMF record in one interval.

nnn

Specifies a site-defined value (up to 255) for the number of punches of SMF records in one interval.

FILE

Punches an entire SMF file in a single interval. In a PRIORITY DUMP situation, it punches the entire SMF file in a single interval.

SMFID=string|null

Defines a four-character string that CA ACF2 for z/VM inserts in all SMF records it writes. You can use this in multi-CPU environments to identify the specific CPU CA ACF2 for z/VM wrote the record for. The specified string appears on most standard CA ACF2 for z/VM reports. The default is NULL. If you specify NULL, the first four characters of the System_Identifier_Default value from the SYSTEM CONFIG file is used. If a value is not specified for System_Identifier_Default, the first four characters of the HCPSYS SYSID macro DEFAULT= operand is used for the SMFID. You must reload the FDR to make this change effective.

Important! If you are upgrading from a pre-3.3 release, CA ACF2 for z/VM automatically accepts and processes SMF data for those records. If you specify 0 (the default) for the PSWD, DSN, LID, RULE, INFO, RSRC, ACFSERVE, CMDLIM, and DIRM operands, pre-release 3.3 CA ACF2 for z/VM uses the default. You must specify a number here only if you did not use the pre-release 3.3 defaults.

If you are installing CA ACF2 for z/VM for the first time, use the default of zero (0) for these operands.

PSWD=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses to log password violations and other system logon events. See the *Caution!* in the SMFID=string|null operand for information about specifying this operand. The default value for this operand is 0.

DSN=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses to log or deny access to a minidisk, CMS file, DOS file accessed through CMS, or OS data set. See the *Caution!* in the SMFID=string|null operand for information about specifying this operand. The default value for this operand is 0. You must reload the FDR to make this change effective.

LID=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses to process a request to insert, delete, or change a logonid record. See the *Caution!* in the SMFID=string|null operand for information about on specifying this operand. The default value for this operand is 0.

RULE=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses to process a request to store, change, or delete an access rule. See the *Caution!* above for information about specifying this operand. The default value for this operand is 0.

INFO=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses for the Infostorage database updates, including information change journal records. See the *Caution!* in the SMFID=string|null operand for information about specifying this operand. The default value for this operand is 0.

RSRC=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses for resource database updates, including resource journal records. See the *Caution!* in the SMFID=string|null operand for information about specifying this operand. The default value for this operand is 0.

ACFSERVE=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses for ACFSERVE command tracking, including the ACFSERVE command journal records. See the *Caution!* in the SMFID=string|null operand for information about specifying this operand. The default value for this operand is 0.

CMDLIM=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses for command limiting violations, including command and diagnose limiting journal records. See the *Caution!* in the SMFID=string|null operand for information about specifying this operand. The default value for this operand is 0.

DIRM=0|nnn

Specifies the SMF record number CA ACF2 for z/VM uses for DirMaint violations. See the *Important!* in the SMFID=string|null operand for information about specifying this operand. The default value for this operand is 0.

ACF2=230|nnn

Defines the SMF record number selected for CA ACF2 for z/VM use. CA ACF2 for z/VM uses this number to write all security-related logging records. This includes password violations, denied system access loggings, trace records, data set and program loggings, resource loggings, rule modification journal records, and infostorage and logonid modification journal records. The default is 230.

LASTPRM=YES

> Checks syntax for additional positional parms, missing commas, or parenthesis. When a syntax error occurs, CA ACF2 for z/VM displays an error message. This operand is required and must always be the last operand of the @SMF macro.

Although the @SMF macro includes the DISKFUL and FILEFUL operands, they are obsolete. They remain in this macro for compatibility only. Using these operands results in the following warning Mnote (macro note):

```
Parameter obsolete ignored
```

The following ACF subcommand displays the record numbers and the various configuration options SMF recording uses:

```
SHOW SMF
```

ACFSERVE Subcommands

The following command notifies the CA-ACF2 service machine that CA ACF2 for z/VM processed an SMF file in unloading status and it is ready to be flagged as a history file:

```
ACFSERVE ARCHIVE SMF
```

The following command tells the CA-ACF2 service machine to switch SMF disks:

```
ACFSERVE SWITCH SMF
```

The following command forces an SMF checkpoint:

```
ACFSERVE CKPT SMF
```

The following command displays the SMF minidisk addresses and status of SMF files:

```
ACFSERVE QUERY SMF
```

The following command reloads the ACFFDR:

```
ACFSERVE RELOAD
```

@SRF-System Request Facility Macro

The @SRF macro defines virtual machines authorized to use the System Request Facility (SRF). The System Request Facility lets site-written applications request access validation and database maintenance services from the CA ACF2 service machine.

Sites that use CA ACF2 for z/VM with CA Top Secret for z/VSE should be aware that the default @SRF macro definitions include VSEIPO and CICSCVT. The VSEIPO definition names the VSE guest machine (as defined in the VM directory) that operates under CA Top Secret for VSE control. CICSCVT names the CICS/VSE system that operates under CA Top Secret for z/VSE control. Review and modify these two definitions when installing CA ACF2 for z/VM or CA Top Secret for z/VSE. You must define an @SRF macro for each VSE guest machine and each CICS partition CA Top Secret for z/VSE is to control.   An example of using the @SRF macro to define the VM directory name of a multiuser SRF who is authorized to use the SRF is @SRF VM,MLID=VM. You can only define a multiuser SRF, not a usercall SRF. You can define a maximum of 256 @SRF specifications.

The syntax for the @SRF macro is:

```
@SRF id,                                          SRF IDENTIFIER
   MLID=name,                        NAME OF MINI-LID DEFINITION
   MODE=ABORT|WARN|LOG|QUIET|(RULE,no-rule,no$mode)OPERATIONAL MODE
   OPTNAME=vseoptmod              NAME OF THE VSE OPTIONS MODULE
```

id

Identifies a SRF-authorized virtual machine. This ID must be the same as the one defined in the VM directory entry for the virtual machine or the CICS partition. Whenever you IPL a virtual machine, CA ACF2 for z/VM compares its VM name to the @SRF IDs. If CA ACF2 for z/VM does not find a match, it assumes no special processing. When it does find a match, the associated options control CA ACF2 for z/VM processing while the guest machine is running. We supply sample @SRF macros for both CA ACF2 for z/VM and CA Top Secret for z/VSE environments. We also provide a sample @SRF for sites that choose the CA Top Secret for z/VSE CICS option.

MLID

Specifies a minilogonid compression algorithm used for this virtual machine. The minilogonid facility (the ID specified in the @MLID macro) omits unnecessary portions of the logonid record from resident storage to conserve space.

MODE

Specifies the mode of the guest machine as it relates to data access. It has an effect on the SRF environment. Set MODE to one of the following:

ABORT

Log attempted violations, issue violation messages, and deny the accesses. This is the default value.

WARN

Log access violations and issue warning messages, but let accesses continue.

LOG

Log data access violations but let access continue.

QUIET

Disable CA ACF2 for z/VM data access rule validations for the guest machine. CA ACF2 for z/VM logonid record and similar user and system access validations still take place.

RULE

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. The $MODE control statement applies only when the (RULE,no-rule,no-$mode) option is in effect and CA ACF2 for z/VM determines that a data access request violates security. The two positional parameters, no-rule and no-$mode, are defined as:

- **no-rule**

  Specifies the action CA ACF2 for z/VM takes if it does not find an access rule when RULE mode is in effect. The value for this parameter can be QUIET, LOG, WARN, or ABORT, as defined above. Be aware that if you run in rule QUIET mode, CA ACF2 for z/VM might handle read errors the same as not found conditions, resulting in allowed access.

- **no-$mode**

  Specifies the action CA ACF2 for z/VM takes if it does not find a $MODE control statement in the applicable access rule set when RULE mode is in effect. The value for this parameter can be QUIET, LOG, WARN, or ABORT, as defined above.

  For example, if user TLCJJS tries to access user TLCVLL's file named XYZ WORKFILE A for write access, but the TLCVLL rule set does not grant user TLCJJS this access, CA ACF2 for z/VM checks the $MODE control statement in the access rule set and bases the access permission decision on the $MODE value. If you specified $MODE(LOG) in the access rule set, CA ACF2 for z/VM allows user TLCJJS write access to XYZ WORKFILE A and creates an CA ACF2 for z/VM logging record. If you specified $MODE(ABORT), CA ACF2 for z/VM denies user TLCJJS access and creates an CA ACF2 for z/VM logging record detailing the access violation attempt.

OPTNAME

Defines the phase name of the guest machine options module for this system. You must code this operand. The phase named must exist in the system core image library on the VSE system. You must add it to the SVA LOADLIST. This operand is applicable to CA Top Secret for z/VSE environments only. It has no effect in a CA ACF2 for z/VM only environment.

The following ACF subcommand displays the current @SRF macro definitions and the options selected for each @SRF macro:

```
SHOW SRF
```

@SYSID-System Identification Macro

The @SYSID macro defines the default system startup ID. CA ACF2 for z/VM maintains a current SYSID value that it uses to select VMO records when you define VM system options. This SYSID value is the default SYSID the ACF command uses when you enter the CONTROL(VMO) mode. CA ACF2 for z/VM also places the SYSID into SMF records so reports can select SMF records based on the SYSID. The syntax for the @SYSID macro is:

```
@SYSID sysid
```

sysid

Specifies the default system ID string CA ACF2 for z/VM uses during system startup to set the initial value of the current SYSID. If you specify null as the value for sysid (the default), the System_Identifier_Default value from the SYSTEM CONFIG file is used. If a value is not specified for System_ Identifier_Default, the HCPSYS SYSID macro DEFAULT= operand is used. The operator can specify SYSID(sysid) in response to the CA ACF2 for z/VM startup prompt to override this default startup SYSID during system IPL. After you complete system startup, you can use the following command to change the current SYSID value:

```
ACFSERVE SET SYSID
```

The following ACF subcommand displays the SYSID used at system IPL and the current SYSID:

```
SHOW SYSTEM
```

@UID-User Identification String Macro

The @UID macro specifies the logonid character fields CA ACF2 for z/VM concatenates (in the specified sequence) to form the UID. CA ACF2 for z/VM dynamically constructs the UID by concatenating these fields from the user's logonid record. It then uses the UID for comparisons during rule interpretation.   You can specify only one @UID macro in the ACFFDR. The maximum length of the total UID is 24 characters. CA ACF2 for z/VM considers the LID field to be eight characters, unless you specify only a partial field. The default ACFFDR defines the UID as the LID field of the logonid record.   The syntax of the @UID macro is:

```
@UID field,...,field|(lidname,length),...,(lidname,length)
```

field

Specifies the external name (specified in the name operand of the @CFDE macro) of the logonid character field to use (NAME). If the UID should not include the full field, you can specify the lidname,length format instead of the field name to select a partial field.

lidname,length

Specifies the internal name of the logonid character field CA ACF2 for z/VM uses (LIDNAME, the symbol operand of the @CFDE macro) and its length. The internal name is the name from the LIDREC DSECT and can be any valid equate, such as LIDLID or LIDLID+1. The length value can be a constant or any valid equate, such as L'LIDLID-1. For example, to specify a UID made up of the default account number (TSOACCT) plus only the second through fourth characters of the LID, the @UID entry is @UID TSOACCT,(LIDLID+1,3).

The following ACF subcommand displays the configuration of the UID as defined in the @UID macro:

```
SHOW UID
```

@VALUES-Generate Verification Values

The @VALUES macro is used in an RSB module to generate field verification values that the VRTN1 routines 3 or 4 use to validate fields of the structured infostorage record.

The syntax for the @VALUES macro is:

```
@VALUES label,operands,TYPE=type,LENGTH=len
```

label

Specify this label in the VPRM1 operand of any @CFDE that uses VRTN1 routine 3 or 4 to validate against these values.

operands

Specifies the valid values. For VRTN1 routine 3 that checks for a value in a list, TYPE can be BINARY, CHAR, or PACKED. You can specify a single value or a list of values. You can mask characters using the standard CA ACF2 for z/VM masking characters, asterisk and dash. Here are some examples:

```
PRIME    @VALUES 1,2,3,5,7,11,TYPE=BINARY,LENGTH=4
COLORS   @VALUES RED,GREEN,BLUE,YELLOW,TYPE=CHAR,LENGTH=6
PROGRAMS @VALUES IFD*****,INA-,IQADVM,TYPE=CHAR,LENGTH=8
EVEN     @VALUES 0,2,4,6,8,TYPE=PACKED,LENGTH=4
```

For VRTN1 routine 4 that checks for inclusion in a range, TYPE can be BINARY, PACKED, or TIMEBIN. You must specify one or more pairs of values, representing the low and high ends of the ranges, enclosed in parenthesis. If the TYPE is TIMEBIN, values are specified as hh:mm, where hh represents hours, in 24-hour format, and mm represents minutes. Here are some examples:

```
HALFWORD @VALUES (0,32767),TYPE=BINARY,LENGTH=4
NOLUNCH  @VALUES (08:00,12:00),(13:00,17:00),TYPE=TIMEBIN,LENGTH=4
```

TYPE=type

Specifies the type of verification values to generate:

BINARY

Four-byte binary

CHAR

Character, length determined by the LENGTH operand.

PACKED

Four-byte packed decimal.

TIMEBIN

Four-byte binary time, represented internally in units of 0.01 seconds past midnight.

LENGTH=len

Indicates the length of character values to generate. If the values given as operands are shorter than this, they are padded on the right with blanks. LENGTH is ignored for types other than CHAR.

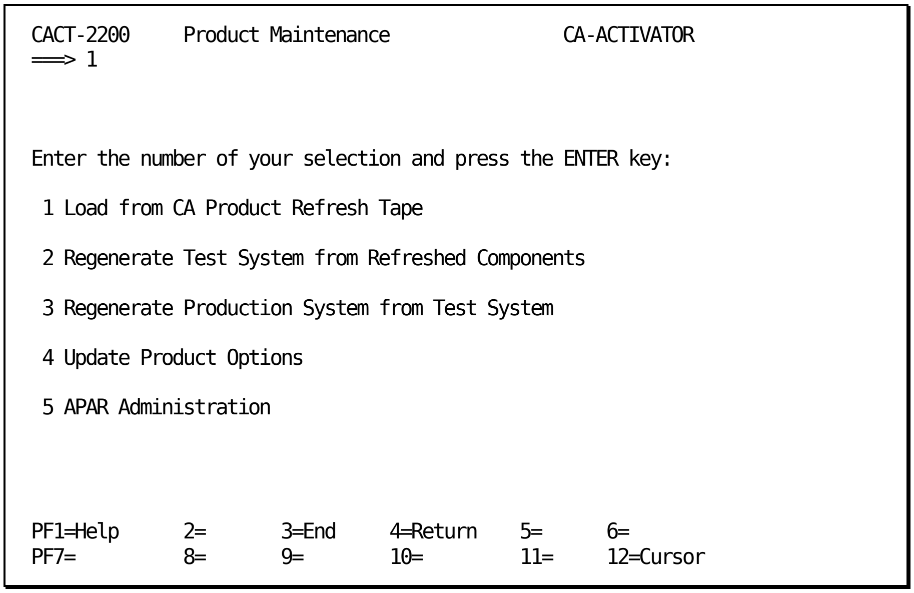
Appendix A: Applying Genlevel Updates

If you are installing a genlevel update, follow the directions in this section. You must make sure that your CAIMAINT test system disk (usually 291) has at least 3600 4 Kb blocks of free space (approximately 20 cylinders) before you can install a genlevel update tape on top of your previous CA ACF2 for z/VM r12 genlevel.

See the Preliminary Installation Steps section in the "Installing CA ACF2 for z/VM" chapter for more information about loading the tape and starting CA-ACTIVATOR.

1. Select option 2 on the CACT-2000 panel.

   The following panel displays:

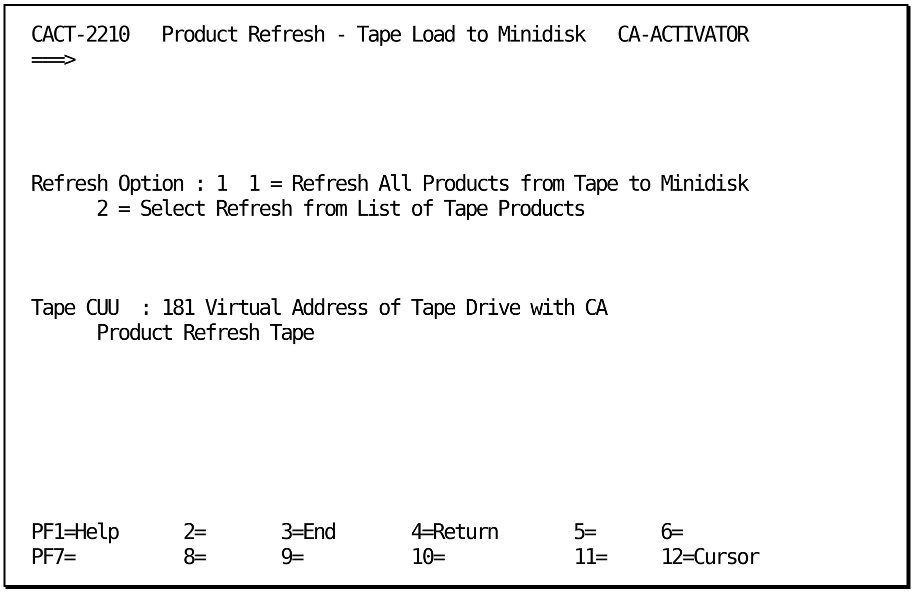


```
CACT-2200     Product Maintenance                CA-ACTIVATOR
===> 1


Enter the number of your selection and press the ENTER key:

 1 Load from CA Product Refresh Tape

 2 Regenerate Test System from Refreshed Components

 3 Regenerate Production System from Test System

 4 Update Product Options

 5 APAR Administration



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

2. Select option 1 to indicate that you want to load files from the refresh tape.

   The following panel displays:

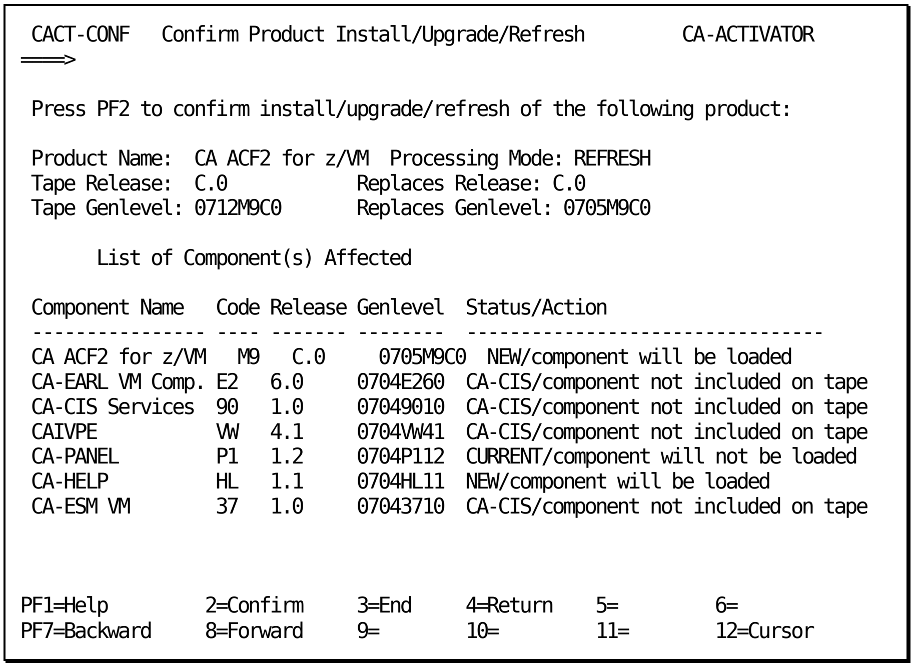
3. Select option 1 to indicate that you want to refresh all the products from the tape.



```
CACT-2210   Product Refresh - Tape Load to Minidisk   CA-ACTIVATOR
===>




Refresh Option : 1  1 = Refresh All Products from Tape to Minidisk
      2 = Select Refresh from List of Tape Products



Tape CUU  : 181 Virtual Address of Tape Drive with CA
      Product Refresh Tape







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

4. Specify the virtual address of the tape drive where the CA genlevel update tape is mounted and press Enter.

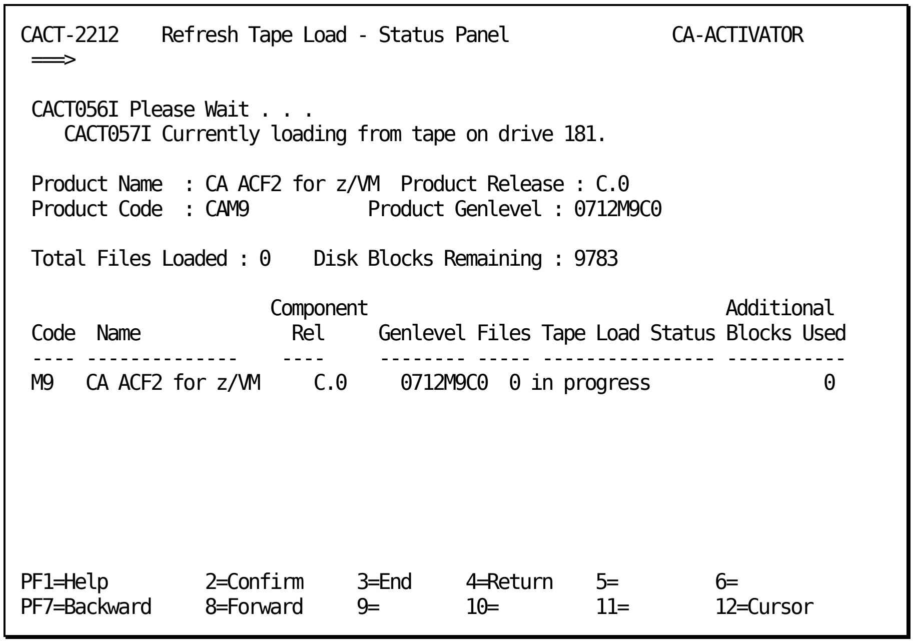
   The following panel displays:



```
CACT-CONF   Confirm Product Install/Upgrade/Refresh        CA-ACTIVATOR
 ===>

Press PF2 to confirm install/upgrade/refresh of the following product:

Product Name:  CA ACF2 for z/VM  Processing Mode: REFRESH
Tape Release:  C.0           Replaces Release: C.0
Tape Genlevel: 0712M9C0      Replaces Genlevel: 0705M9C0

      List of Component(s) Affected

Component Name   Code Release Genlevel  Status/Action
---------------- ---- ------- --------  ---------------------------------
CA ACF2 for z/VM   M9   C.0     0705M9C0  NEW/component will be loaded
CA-EARL VM Comp. E2   6.0     0704E260  CA-CIS/component not included on tape
CA-CIS Services  90   1.0     07049010  CA-CIS/component not included on tape
CAIVPE           VW   4.1     0704VW41  CA-CIS/component not included on tape
CA-PANEL         P1   1.2     0704P112  CURRENT/component will not be loaded
CA-HELP          HL   1.1     0704HL11  NEW/component will be loaded
CA-ESM VM        37   1.0     07043710  CA-CIS/component not included on tape



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

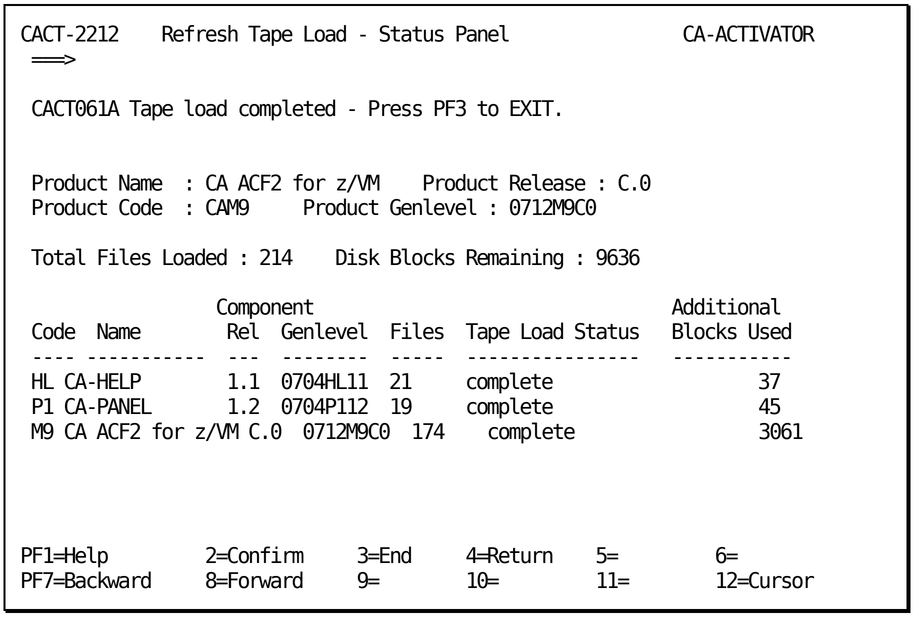
5. Press PF2 to confirm the refresh of CA ACF2 for z/VM.

   The following panel displays:



```
CACT-2212    Refresh Tape Load - Status Panel               CA-ACTIVATOR
 ===>

 CACT056I Please Wait . . .
    CACT057I Currently loading from tape on drive 181.

 Product Name  : CA ACF2 for z/VM  Product Release : C.0
 Product Code  : CAM9          Product Genlevel : 0712M9C0

 Total Files Loaded : 0    Disk Blocks Remaining : 9783

                      Component                                Additional
 Code  Name              Rel     Genlevel Files Tape Load Status Blocks Used
 ---- --------------    ----     -------- ----- ---------------- -----------
 M9   CA ACF2 for z/VM     C.0     0712M9C0  0 in progress                0




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

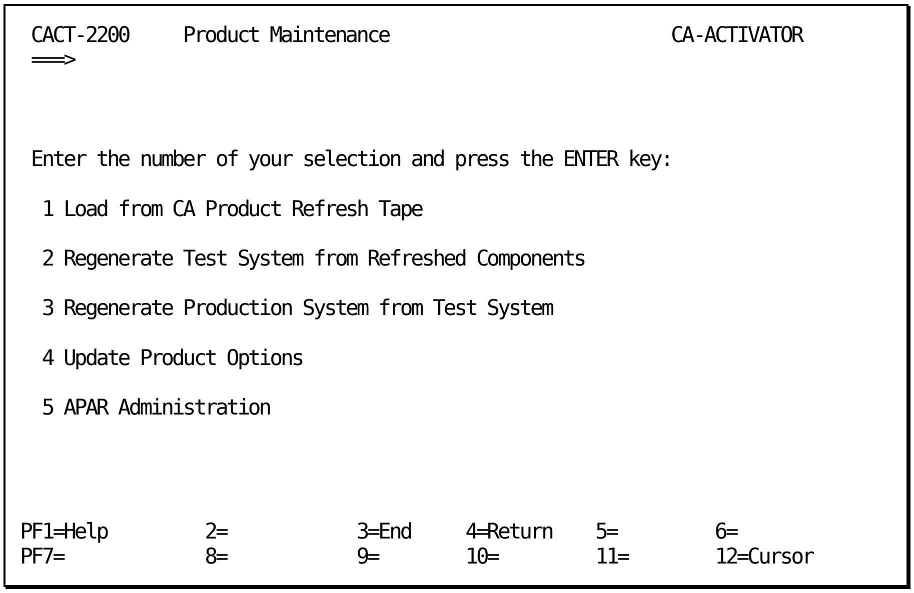
When the genlevel update tape is loaded, the following panel displays:



```
CACT-2212    Refresh Tape Load - Status Panel                CA-ACTIVATOR
 ===>

 CACT061A Tape load completed - Press PF3 to EXIT.


 Product Name  : CA ACF2 for z/VM    Product Release : C.0
 Product Code  : CAM9     Product Genlevel : 0712M9C0

 Total Files Loaded : 214    Disk Blocks Remaining : 9636

                  Component                                Additional
 Code  Name         Rel  Genlevel  Files  Tape Load Status    Blocks Used
 ---- -----------   ---  --------  -----  ----------------    -----------
 HL CA-HELP         1.1  0704HL11  21     complete                   37
 P1 CA-PANEL        1.2  0704P112  19     complete                   45
 M9 CA ACF2 for z/VM C.0  0712M9C0  174    complete                  3061



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

6. Press PF3.

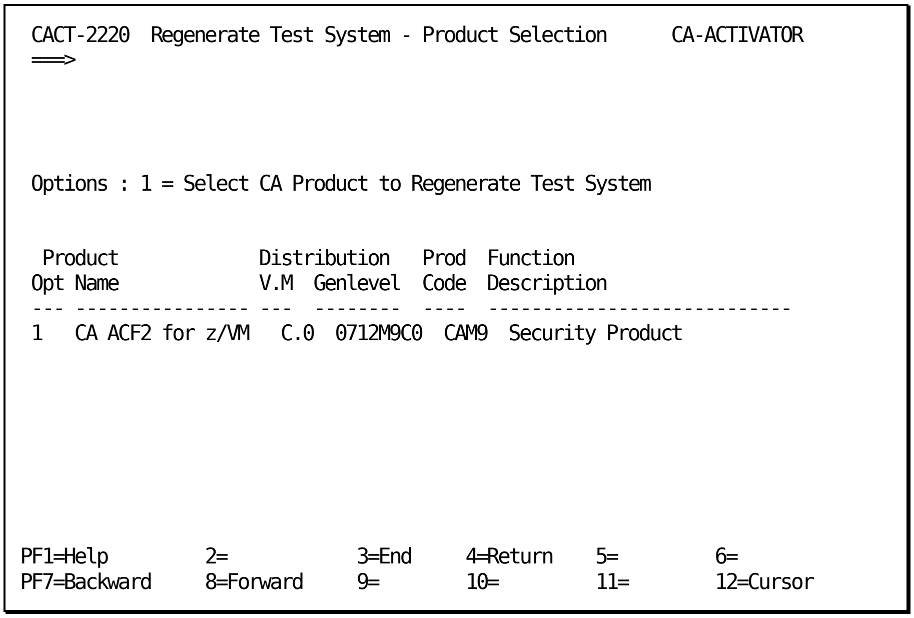
   The following panel displays:



```
CACT-2200     Product Maintenance                              CA-ACTIVATOR
===>



Enter the number of your selection and press the ENTER key:

 1 Load from CA Product Refresh Tape

 2 Regenerate Test System from Refreshed Components

 3 Regenerate Production System from Test System

 4 Update Product Options

 5 APAR Administration



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

7. Select option 2 and press Enter.

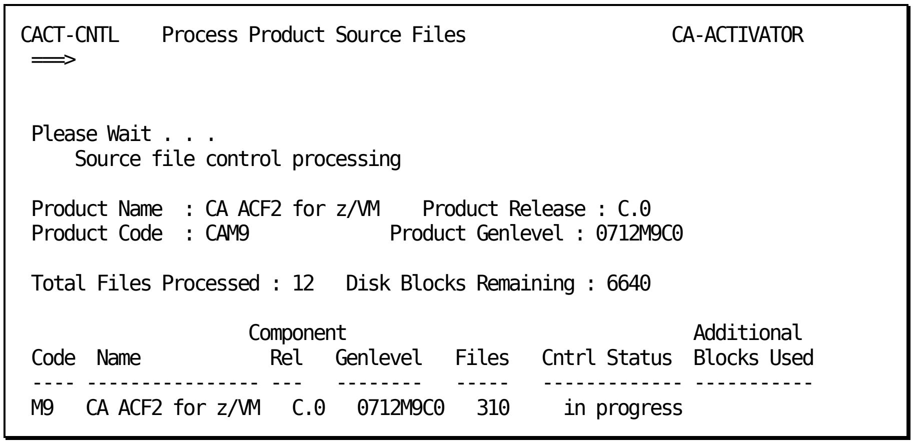
   The following panel displays:



```
CACT-2220  Regenerate Test System - Product Selection      CA-ACTIVATOR
===>




Options : 1 = Select CA Product to Regenerate Test System


 Product               Distribution   Prod  Function
Opt Name               V.M  Genlevel  Code  Description
--- ---------------- ---  --------  ----  ----------------------------
1   CA ACF2 for z/VM   C.0  0712M9C0  CAM9  Security Product






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

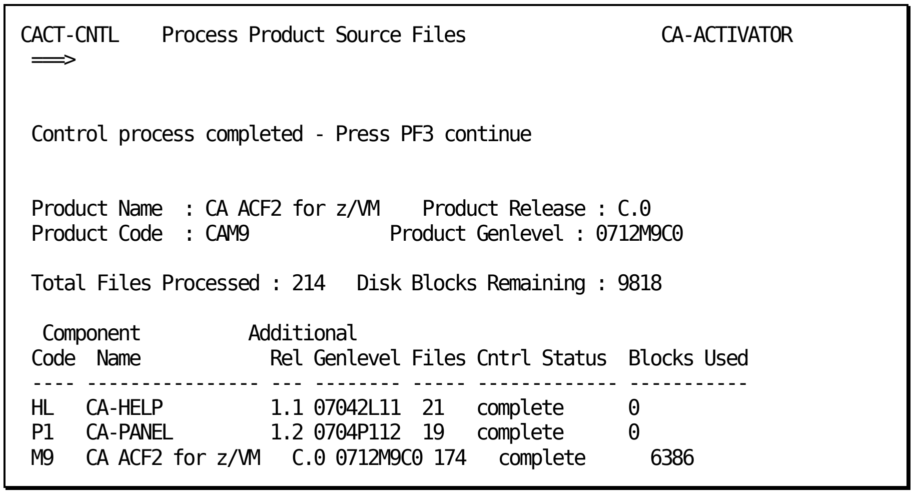
8. Select option 1 to indicate you want to regenerate the test system and press Enter.

   The following panel displays:



```
CACT-CNTL    Process Product Source Files                 CA-ACTIVATOR
 ==>


 Please Wait . . .
     Source file control processing

 Product Name  : CA ACF2 for z/VM    Product Release : C.0
 Product Code  : CAM9              Product Genlevel : 0712M9C0

 Total Files Processed : 12   Disk Blocks Remaining : 6640

                     Component                                Additional
 Code  Name            Rel   Genlevel   Files   Cntrl Status  Blocks Used
 ---- ---------------- ---   --------   -----   ------------- -----------
 M9   CA ACF2 for z/VM   C.0   0712M9C0   310     in progress
```

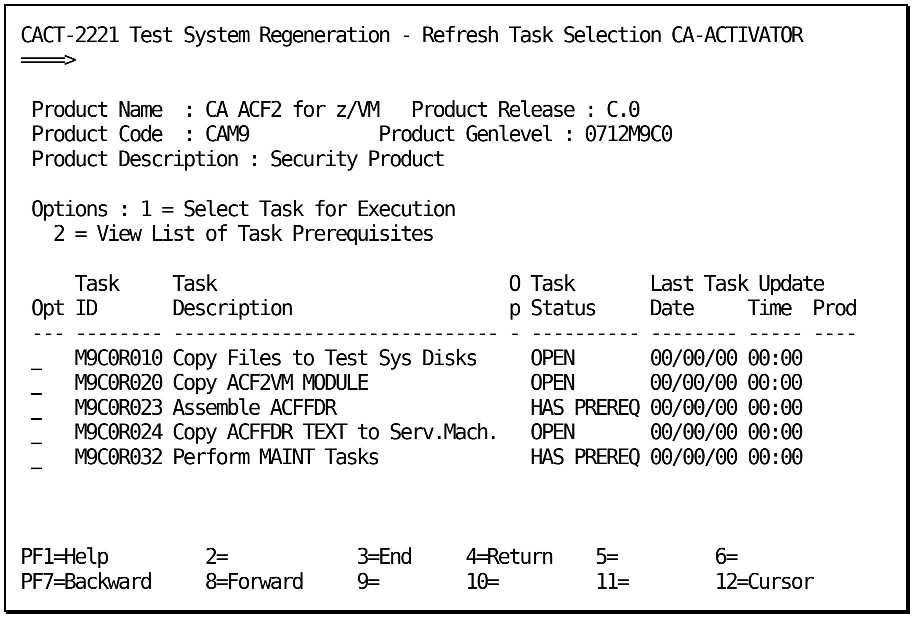
This processing can take quite a while to complete. When processing is complete, a message appears on the panel and CA-ACTIVATOR marks the Cntrl Status column as complete, as shown in the following panel:



```
CACT-CNTL    Process Product Source Files                CA-ACTIVATOR
 ==>


 Control process completed - Press PF3 continue


 Product Name  : CA ACF2 for z/VM    Product Release : C.0
 Product Code  : CAM9              Product Genlevel : 0712M9C0

 Total Files Processed : 214   Disk Blocks Remaining : 9818

  Component          Additional
 Code  Name            Rel Genlevel Files Cntrl Status  Blocks Used
 ---- ---------------- --- -------- ----- ------------- -----------
 HL   CA-HELP          1.1 07042L11  21   complete      0
 P1   CA-PANEL         1.2 0704P112  19   complete      0
 M9   CA ACF2 for z/VM   C.0 0712M9C0 174   complete      6386
```

9. Press PF3.

   The following panel displays:



```
CACT-2221 Test System Regeneration - Refresh Task Selection CA-ACTIVATOR
===>

 Product Name  : CA ACF2 for z/VM   Product Release : C.0
 Product Code  : CAM9            Product Genlevel : 0712M9C0
 Product Description : Security Product

 Options : 1 = Select Task for Execution
   2 = View List of Task Prerequisites

     Task     Task                           O Task       Last Task Update
 Opt ID       Description                    p Status     Date     Time  Prod
 --- -------- ------------------------------ - ---------- -------- ----- ----
 _   M9C0R010 Copy Files to Test Sys Disks     OPEN       00/00/00 00:00
 _   M9C0R020 Copy ACF2VM MODULE               OPEN       00/00/00 00:00
 _   M9C0R023 Assemble ACFFDR                  HAS PREREQ 00/00/00 00:00
 _   M9C0R024 Copy ACFFDR TEXT to Serv.Mach.   OPEN       00/00/00 00:00
 _   M9C0R032 Perform MAINT Tasks              HAS PREREQ 00/00/00 00:00



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

10. Select option 1 to indicate which task you want to perform. The next section contains information on these tasks.

11. When you have completed all the refresh tasks, press PF3.

Refresh Tasks

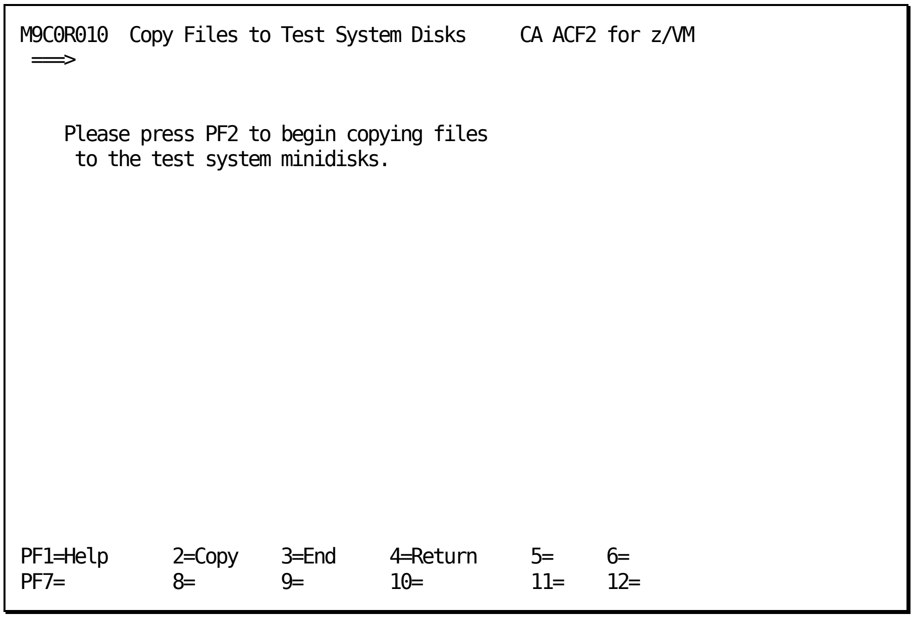
Following is a list of the tasks you must perform to install a new genlevel. See the individual tasks for detailed instructions.

Task #	Task Title	Description
M9C0R010	Copy files to test system disks	Copies the CA ACF2 for z/VM option file to the product generation disk.
M9C0R020	Replace ACF2VM MODULE	Lets you replace the ACF2VM MODULE.
M9C0R023	Reassemble ACFFDR	Lets you reassemble the ACFFDR.
M9C0R024	Copy ACFFDR TEXT to service machine	Copies the ACFFDR TEXT file from the local options disk to the service machine disk.
M9C0R032	Perform MAINT tasks	Processes the ACF2TASK EXEC.

Task M9C0R010: Copy Files to Test System Disks

This task copies the files to the test system minidisks. To complete this task:

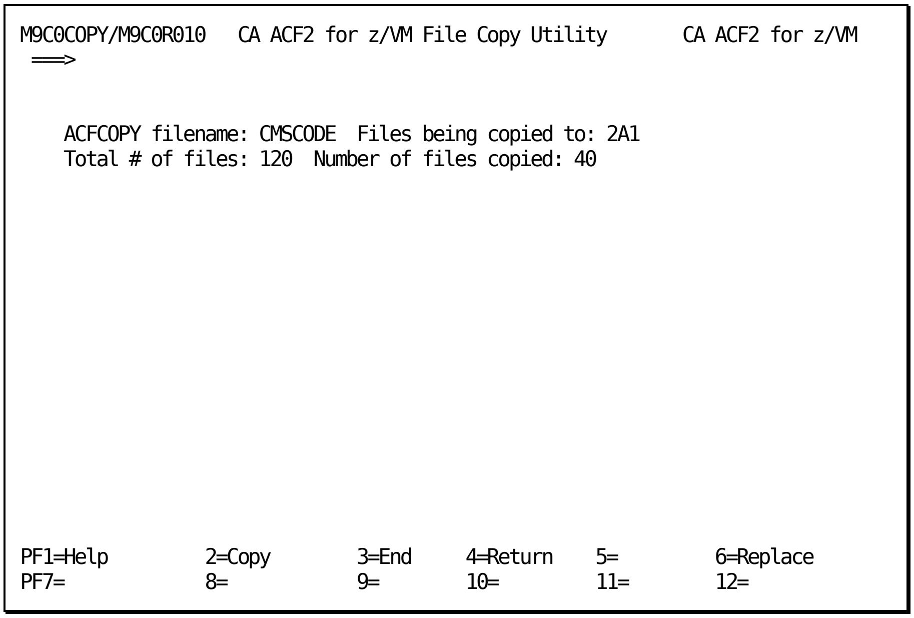
1. Select panel M9C0R010 from the Test System Regeneration - Refresh Task Selection menu (CACT-2221). You see the following panel that lets you copy files to your product generation minidisks.



```
M9C0R010  Copy Files to Test System Disks     CA ACF2 for z/VM
 ===>


    Please press PF2 to begin copying files
     to the test system minidisks.















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

2. Press PF2 to copy the files.

   The following panel displays information for each set of files you are copying. For example, if you specified Y for the CA ACF2 for z/VM full-screen file support and CA ACF2 for z/VM help files for the ACF command, this panel will display information for the full-screen file support, then the information for the help files.



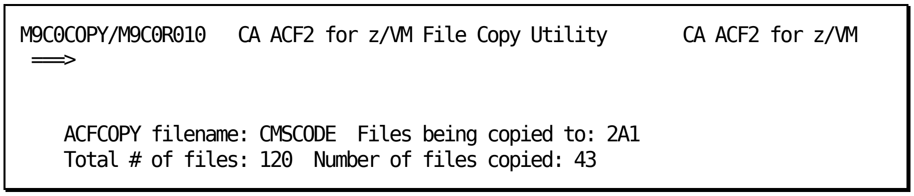
```
M9C0COPY/M9C0R010   CA ACF2 for z/VM File Copy Utility      CA ACF2 for z/VM
 ===>


    ACFCOPY filename: CMSCODE  Files being copied to: 2A1
    Total # of files: 120  Number of files copied: 40













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

   This panel tells you the following information:

   - What control file is used to control the copy
   - What disk the files are being copied to
   - How many files will be copied
   - How many files were copied so far.

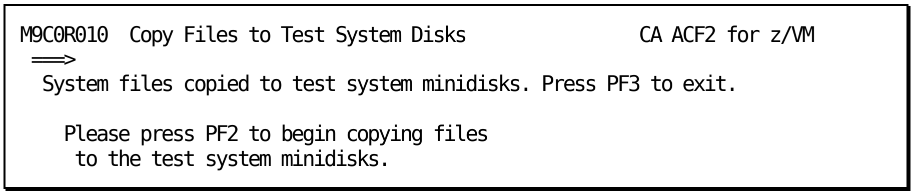
   The value for Number of files copied: changes as you watch the panel.

When each set of files is copied (for example, the full-screen file support), a message appears on the panel confirming that the full-screen files are copied:



```
M9C0COPY/M9C0R010   CA ACF2 for z/VM File Copy Utility      CA ACF2 for z/VM
 ===>


    ACFCOPY filename: CMSCODE  Files being copied to: 2A1
    Total # of files: 120  Number of files copied: 43
```

When all the files are copied, the following panel displays:



```
M9C0R010  Copy Files to Test System Disks              CA ACF2 for z/VM
 ===>
  System files copied to test system minidisks. Press PF3 to exit.

    Please press PF2 to begin copying files
     to the test system minidisks.
```

3. Press PF3 to return to the Task Selection Menu.

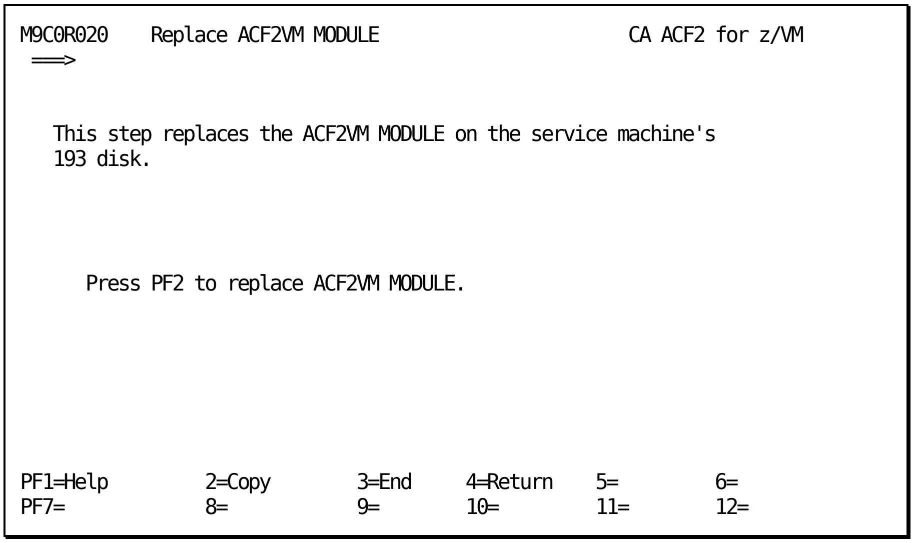
Task M9C0R020: Replace ACF2VM MODULE

This task lets you replace the ACF2VM MODULE on your service machine's disk.

To complete this task:

1. Select task M9C0R020 from the Test System Regeneration - Refresh Task Selection menu (CACT-2221).

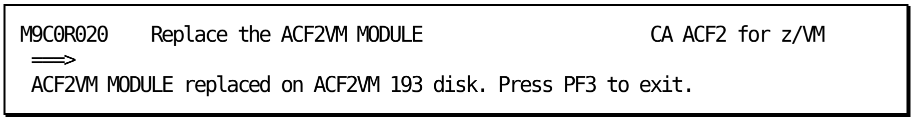
   The following panel display lets you replace the ACF2VM MODULE:



```
M9C0R020    Replace ACF2VM MODULE                          CA ACF2 for z/VM
 ===>


   This step replaces the ACF2VM MODULE on the service machine's
   193 disk.




      Press PF2 to replace ACF2VM MODULE.







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

2. Press PF2 to replace the ACF2VM MODULE.

   When processing is complete, a message appears on the panel confirming that you replaced the ACF2VM MODULE.



```
M9C0R020    Replace the ACF2VM MODULE                    CA ACF2 for z/VM
 ===>
 ACF2VM MODULE replaced on ACF2VM 193 disk. Press PF3 to exit.
```

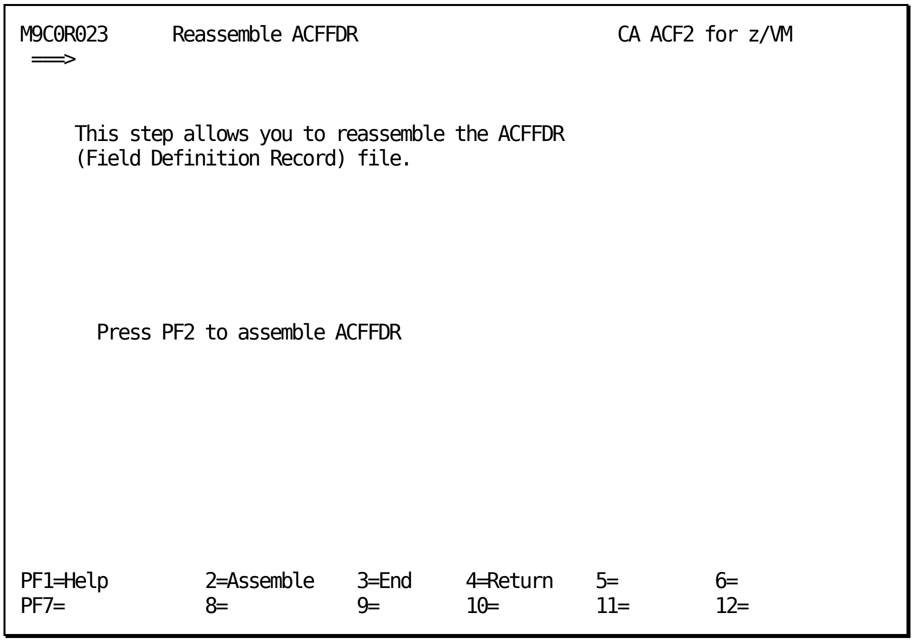
3. Press PF3 to return to the Test System Regeneration - Refresh Task Selection menu (CACT-2221).

Task M9C0R023: Reassemble ACFFDR

This task lets you reassemble the ACFFDR. To complete this task:

1. Select task M9C0R023 from the Test System Regeneration - Refresh Task Selection menu (CACT-2221).

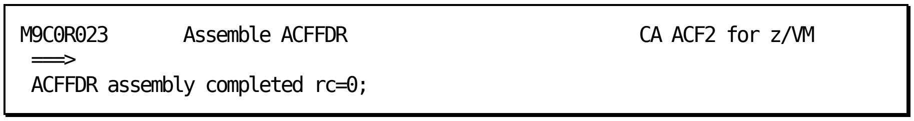
   The following panel displays:



```
M9C0R023      Reassemble ACFFDR                        CA ACF2 for z/VM
 ===>


     This step allows you to reassemble the ACFFDR
     (Field Definition Record) file.






       Press PF2 to assemble ACFFDR










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

2. Press PF2 to reassemble the ACFFDR.

   A message appears on the panel confirming the assembly:



```
M9C0R023      Assemble ACFFDR                          CA ACF2 for z/VM
 ===>
 ACFFDR assembly completed rc=0;
```

3. Press PF3 to return to the Test System Regeneration - Refresh Task Selection menu (CACT-2221).

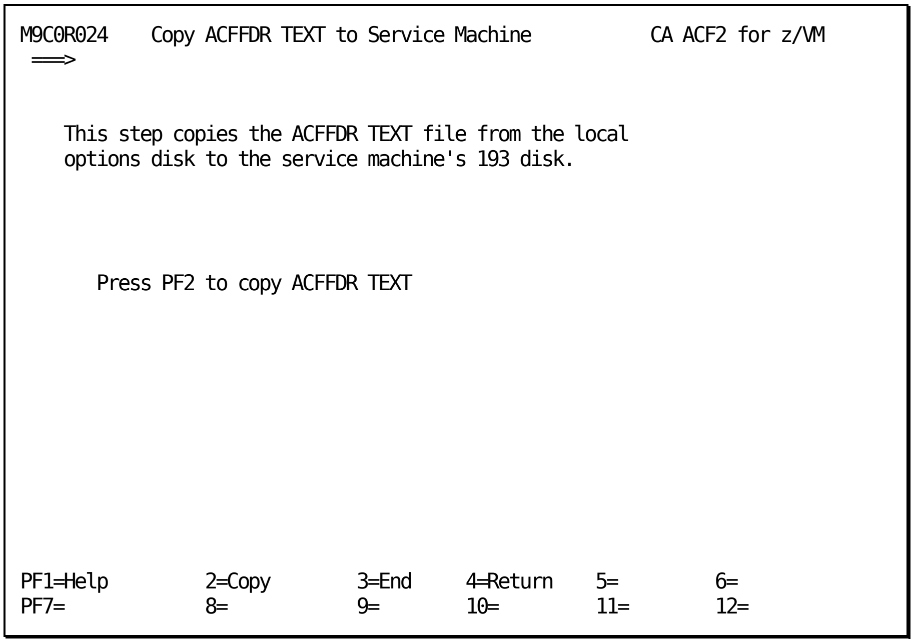
Task M9C0R024: Copy ACFFDR TEXT to Service Machine

This task copies the ACFFDR TEXT file to your service machine.

To complete this task:

1. Select task M9C0R024 from the Test System Regeneration - Refresh Task Selection menu (CACT-2221).

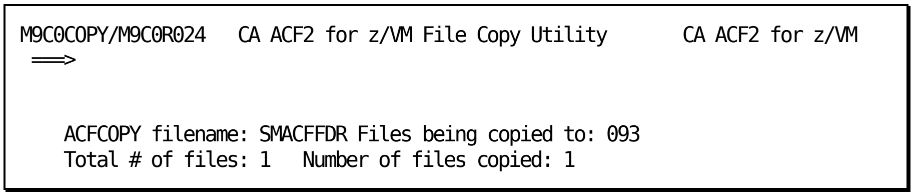
   The following panel displays:



```
M9C0R024    Copy ACFFDR TEXT to Service Machine          CA ACF2 for z/VM
 ===>


    This step copies the ACFFDR TEXT file from the local
    options disk to the service machine's 193 disk.




       Press PF2 to copy ACFFDR TEXT










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

2. Press PF2 to copy the exec.

   The following panel displays.



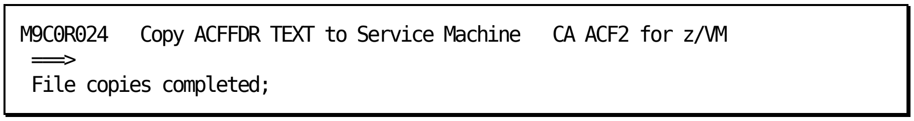
```
M9C0COPY/M9C0R024   CA ACF2 for z/VM File Copy Utility      CA ACF2 for z/VM
 ===>


    ACFCOPY filename: SMACFFDR Files being copied to: 093
    Total # of files: 1   Number of files copied: 1
```

   This panel tells you the following information:

   - What control file is used to control the copy
   - What disk the files are being copied to
   - How many files will be copied
   - How many files were copied so far.

   The value for Number of files copied: changes as you watch the panel.

   When processing is complete, a message appears on the panel confirming that all the files are copied.



```
M9C0R024   Copy ACFFDR TEXT to Service Machine   CA ACF2 for z/VM
 ===>
 File copies completed;
```

3. Press PF3 to return to the Test System Regeneration - Refresh Task Selection menu (CACT-2221).

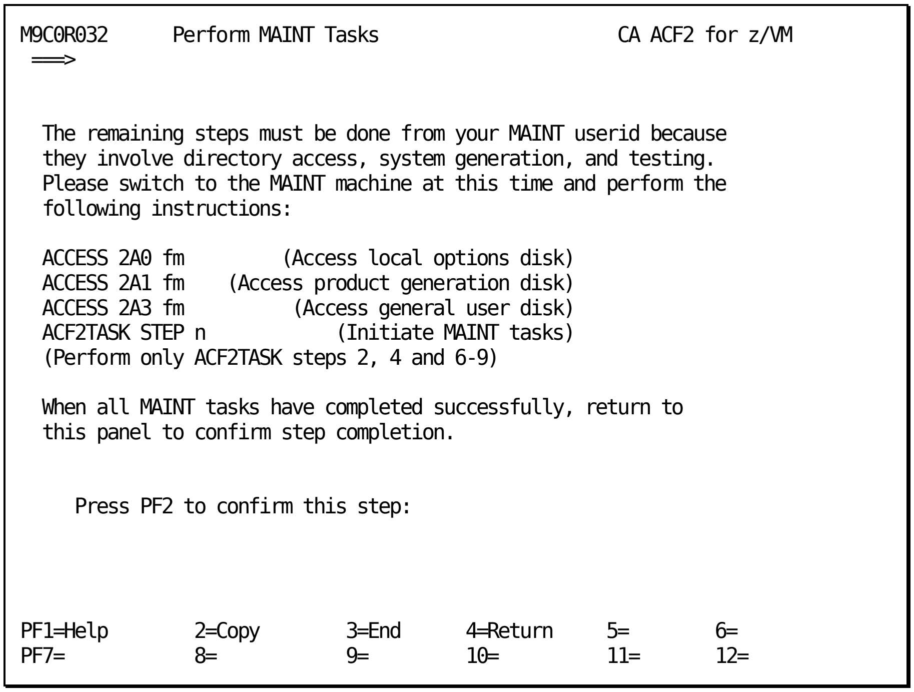
Task M9C0R032: Perform MAINT Tasks

You have completed all the steps that you can execute under CA-ACTIVATOR. You must perform the remaining installation tasks from the MAINT user ID. This task instructs you to switch to the MAINT service machine and initiate the ACF2TASK EXEC.

To complete this task:

1. Select task M9C0R032 from the Test System Regeneration - Refresh Task Selection menu (CACT-2221).

   The following panel displays:



```
M9C0R032      Perform MAINT Tasks                      CA ACF2 for z/VM
 ===>


  The remaining steps must be done from your MAINT userid because
  they involve directory access, system generation, and testing.
  Please switch to the MAINT machine at this time and perform the
  following instructions:

  ACCESS 2A0 fm         (Access local options disk)
  ACCESS 2A1 fm    (Access product generation disk)
  ACCESS 2A3 fm          (Access general user disk)
  ACF2TASK STEP n            (Initiate MAINT tasks)
  (Perform only ACF2TASK steps 2, 4 and 6-9)

  When all MAINT tasks have completed successfully, return to
  this panel to confirm step completion.


     Press PF2 to confirm this step:




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

2. Log onto the VM maintenance ID. Perform the links described on the above panel.

   You do not need to perform all the ACF2TASK EXEC tasks. The following table shows you which ACF2TASK tasks you need to perform. To use the ACF2TASK EXEC to apply genlevel updates, follow the steps below:

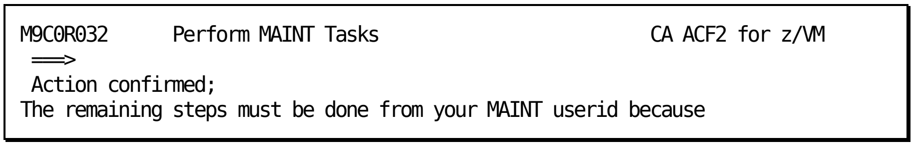
Step	Description
2	Enter **ACF2TASK STEP 2**. This task modifies the VMXAOPTS and assembles HCPAC0. See Step 2: Modify VMXAOPTS in the “Installing CA ACF2 for z/VM” chapter for detailed information about performing this step.

Step	Description
4	Enter **ACF2TASK STEP 4**. This task generates the CP nucleus, IPLs, and updates the VMO records. See Step 4: Generate the CP Nucleues, IPL, and Update VMO Records in the “Installing CA ACF2 for z/VM” chapter for detailed information about performing this step.
6	Enter **ACF2TASK STEP 6**. This task adds intercepts to CMS and CMS/DOS modules. See Step 6: Add CA ACF2 for z/VM Intercepts to CMS Modules in the “Installing CA ACF2 for z/VM” chapter for detailed information about performing this step.
7	Enter **ACF2TASK STEP 7**. This task generates the CMS nucleus with CA ACF2 for z/VM intercepts. See Step 7: Generate and Test the CMS Nucleus in the “Installing CA ACF2 for z/VM” chapter for detailed information about performing this step.
8	Enter **ACF2TASK STEP 8**. This task generates module files for various CMS commands. See Step 8 Generate Module Files for CMS Commands in the “Installing CA ACF2 for z/VM” chapter for detailed information about performing this step.
9	Enter **ACF2TASK STEP 9**. This task finishes the genlevel update application. See Step 9: The Final IPL with CA ACF2 for z/VM in the “Installing CA ACF2 for z/VM” chapter for detailed information about performing this step.

See Task M9C0I032: Perform MAINT Tasks in the “Installing CA ACF2 for z/VM” chapter for detailed information about running the ACF2TASK EXEC.

3. When you have successfully completed all ACF2TASK steps, return to M9C0R032 and press PF2 to confirm the step is complete.

   A message appears on M9C0R032 confirming the action:



```
M9C0R032      Perform MAINT Tasks                          CA ACF2 for z/VM
 ===>
 Action confirmed;
The remaining steps must be done from your MAINT userid because
```

4. Press PF3 to return to the Test System Regeneration - Refresh Task Selection menu (CACT-2221).

Index

$

$LASTPRM operand of @MLID macro • 277

@

@CFDE macro
  enabling • 261
  syntax • 262
@CFDE-Create Field Definition Entry Macro • 262
@CFDEDFT-Generate Default Value for Database Record Field • 273
@DDSN-Dynamic Data Set Allocation Macro • 274
@GNVMFDR macro, syntax • 275
@GNVMFDR-Field Definition Record Generator Macro • 275
@GROUP-Group Names for Logonid Display • 275
@HEADER macro
  enabling • 261
  syntax • 277
@HEADER-Heading Formatting for Logonid Display • 277
@MLID-Logonid Compression Macro • 277
@SETUP macro, syntax • 279
@SETUP-DSECT Map Initialization Macro • 279
@SMF macro
  enabling • 261
  syntax • 279
@SMF-System Management Facility Macro • 279
@SRF macro
  displaying macro definitions • 286
  enabling • 261
  syntax • 286
@SRF-System Request Facility Macro • 286
@SYSID macro, syntax • 288
@SYSID-System Identification Macro • 288
@UID macro, syntax • 289, 296
@UID-User Identification String Macro • 289
@VALUES-Generate Verification Values • 289

A

About the VMXAOPTS Macro • 259
access
  attempt SMF record number • 279
Accessing the Online Client Support System • 257
Account Validation • 155
ACF command
  ACFSERVE
    ARCHIVE SMF • 285
    CKPT SMF • 285
    QUERY SMF • 285
    SWITCH SMF • 285
  displayed header, defining • 260
  LIST subcommand format • 277
  SHOW
    DDSNS • 274
    FIELDS • 275
    SRF • 286
    STATE • 289
    SYSTEM • 279, 288
ACF2ASM EXEC, modifying • 93
ACF2DEFC AMSERV file sample • 233
ACF2DSC PARMS Parameter Files • 184
ACF2USER MACLIB • 100
ACF2VM MODULE, copying • 98
ACFDBVSM • 230
ACFFDR
  @CFDE macro • 261, 262
  @DDSN macro • 261, 274
  @GNVMFDR macro • 275
  @GROUP macro • 261, 275
  @HEADER macro • 261, 277
  @MLID macro • 261, 277
  @SETUP macro • 279
  @SMF macro • 261, 279
  @SRF macro • 261, 286
  @SYSID macro • 288
  @UID macro • 289, 296
  activating • 261
  assembling • 230
    during installation • 103
  copying • 105, 301
  CSECT, creating • 275
  defaults • 261
  defining
    ACF command display header • 260
    LID compression algorithm • 260
    logonid record fields • 260
  generating • 260
  initializing assembly equates • 260
  macros

@CFDE • 260
@CFDEDFT • 260
@DDSN • 260
@GNVMFDR • 260
@GROUP • 260
@HEADER • 260
@MLID • 260
@SETUP • 260
@SMF • 260
@SRF • 260
@SYSID • 260
@UID • 260
@VALUES • 260
summary • 260
mapping DSECTs • 260
Missing Object
@CFDE • 261
@DDSN • 261
@GROUP • 261
@HEADER • 261
@MLID • 261
@SMF • 261
@SRF • 261
modifying • 230
during installation • 103
reassembling • 261, 300
reloading • 230
ACFFDR Macro Summary • 260
ACFLIDGN utility • 115
ACFLINIT EXEC sample • 235
ACFMDEST macro • 202
ACFMSGEN macro • 203
ACFRPTCL report • 243
ACFSERVE operand of @SMF macro • 279
ACFSERVE subcommand
ARCHIVE SMF • 285
CKPT SMF • 285
QUERY SMF • 285
SWITCH SMF • 285
tracking SMF record number • 279
ACFSERVE Subcommands • 285
ACFSTART EXEC, copying • 98
Activating DSC • 185
Activating shared databases • 222
Activating Shared Databases on VM Systems • 222
Add and Relocate Modules • 165
Add CA ACF2 intercepts • 133
Adding user fields • 100
Advanced Program-to-Program Communication (APPC/VM) Support • 159
Advantage CA-Earl Reporting Service • 29
ALTER operand • 262
Altering Message Attributes through ACFCPMST • 198
AMSERV support, installing • 250
Applying Genlevel Updates • 291
Assemble
ACFFDR • 103, 300
Assembling ACFFDR • 230
Assembly equates, initializing • 260
ATTRIBUTE
VMSAF logonid • 193
AUTH operand • 262
AUTOLOG and XAUTOLOG Support • 155

B

Base Control Files • 171
BASE specification of ACFFDR @MLID macro • 277
BATEXIT2 exit of CMS Batch • 178
BITMAP operand • 262
BLOCKS
SMF data • 279
SMF header • 279
BookManager files • 62
Bundle Control Statement into Control Files • 170
BYPASS • 170
Bypass Intercept Processing • 170

C

CA
Commitment, Quality, Innovation • 12
CA ACF2 for z/VM Installation Task Selection Menu • 58
CA ACF2 for z/VM Virtual Machine Requirements • 21
CA certificate • 206
CA Education Services
Ready When You Are • 12
CA LMP
using • 24
CA Standard Security Facility • 177
CA Technologies Product References • 3
CA Technology Services
Delivering Business Value On Your Terms • 11
CA-ACTIVATOR • 27

CA-ACTIVATOR Installation Virtual Machine Requirements • 22
CA-ESM • 29
CAICCI • 28, 185
CAISSF • 28, 177
CAKEYS • 24
CA-REGISTER • 220
CBPROC operand • 262
CCIVM parameter • 185
CFDENME operand • 262
Change logonid SMF record number • 279
CHECK
  KEYWORD
    CAXABLD • 174
Check the CAXALOAD Control Statements • 174
Checkpoint, SMF • 285
CHECKPT operand of @SMF macro • 279
CLEAR
  INTERCPT • 167
  RENAMEXT • 169
CMDLIM operand of @SMF macro • 279
CMS
  BATCH
    BATEXIT2 exit • 178
    installing • 178, 179
    submitting jobs • 179
  CA-ACF2 filename • 274
  commands
    executing • 33
  databases
    converting • 111
    setting up • 109
  entry points • 167
  Missing Object
    customizing • 163
    generating • 134
    testing • 134
  modules • 133
  option, specifying • 72
  preliminary installation • 129
CMS Batch Facility • 178
Collecting Diagnostic Data • 256
command
  CMS • 33
  CP • 33
  files, copying • 76
  limiting violation SMF record number • 279
Command Notation • 34
commands
  Database Synchronization Component • 184
  DSC • 184
  notation • 34
  SMSG • 184
    DSC • 185
    DSCADMIN • 185
    DSCFILE • 185
    DSCLOCAL • 185
    DSCNODES • 185
    DSCOUT • 185
    DSCTRACE • 185
    RESETSTATS • 185
    ST • 185
    STATS • 185
    STATUS • 185
    SYSOUT • 185
Common Services
  CA LMP requirements • 24
Compile Access Rules • 153
Compile Command Limiting Rules • 154
Compile Command Syntax Models • 153
Compile Diagnose Limiting Rules • 154
Compile Resource Rules • 154
COMPLETE status description • 58
Complex Synchronizing Environments • 192
Component control files • 163, 171, 172
Component Control Files • 172
configuration file processing • 206
Configuring CA ACF2 for z/VM Security for Use With the PAM Server • 212
Configuring CA-REGISTER • 220
configuring PAM Server • 204
Configuring the PAM Server • 204
connection management • 206
Conserve storage space using @MLID • 277
Considerations • 204
Console Interface • 215
Contact CA Technologies • 3
Contacting Customer Support • 257
control files
  obtaining • 164
  types • 163, 171
Control options • 259
Control Statement Syntax • 164
control statements
  Missing Object
    adding modules • 165
    bundling • 170
    checking • 174

defining intercepts • 167
deleting modules • 166
relocating modules • 165
syntax • 164
Convert CMS databases • 111
Converting VM directory • 236
COPY
ACF2VM MODULE • 98
ACFFDR TE T • 105
ACFFDR TEXT • 301
ACFSTART EXEC • 98
books for language support • 198
eTrust CA-ACF2 command files • 76
files • 79
COUNTER operand • 262
COUPLE operand • 262
COUPTYP operand • 262
CP
commands • 33
directory, updating • 83
entry points • 167
generating nucleus • 123
Missing Object
customizing • 163
generating • 123
CP Free Storage • 22
CP or CMS Commands • 33
CP Resident Nucleus • 23
CPDEL • 166
CPIRES • 165
CPMOVIRS • 165
CPMOVPAG • 165
CPMOVRES • 165
CPPAG • 165
CPPAGMP • 165
CPRES • 165
CPRESMP • 165
creating
VSAM master catalog • 232
VSAM minidisk • 231
CSE/ISF • 195
CSECT of ACFFDR, creating • 275

D

DASD
Dump and Restore, installing support • 180
VM/ESA or z/VM systems • 179
VM/ESA systems • 179
VM/XA systems • 179
DASD Dump and Restore (DDR) Support • 179
DASD Dump and Restore support • 180
DASD Space Requirements • 16
Data blocks, SMF • 279
database
CMS
filename for eTrust CA-ACF2 databases • 274
setting up • 109
converting
to VSAM • 111
VM directory • 236
defining
CMS filenames • 274
VSAM • 233
groups, defining • 260
initializing VSAM • 235
merging • 230
SHARED
activating • 222
prerequisites • 222
VSAM
converting to • 111
defining • 233
initializing • 235
Database Synchronization Component • 180
Database Synchronization Component Processing • 190
databases
synchronizing • 180
Dataspace Support • 158
DDR support • 180
DDR support, installing • 180
VM/ESA or z/VM systems • 179
VM/ESA systems • 179
VM/XA systems • 179
DDSN • 123
debugging statements • 206
DEFAULT
ACFFDR • 261
logonid record groups • 275
SMF ID • 279
Define CA-ACF2 service machine • 81
Define Intercepts • 167
Defining Language Messages • 198
Defining User IDs through ACFMDEST • 202
Defining VSAM databases • 233
Delete Modules • 166
Descriptions, status • 58

DESTID operand of @SMF macro • 279
DFT operand • 262
DFTAD operand • 262
DIAG A0 • 193
DIAG 'A0' Subcode '04'
    CA ACF2 for z/VM Password Validation • 193
Diagnostic Procedures • 252
DIAL Support • 157
directory
    merging • 115
    modifying user • 223
    VM • 115
DIRM operand of @SMF macro • 279
DirMaint Consideration with ISF/CSE Complexes • 194
DLLs • 206
DMSFRE161T Message Issued • 255
DMSMA104S errors • 255
Documentation Changes • 4
DSC • 180
DSC parameter • 185
DSCADMIN parameter • 185
DSCFILE parameter • 185
DSCLOCAL parameter • 185
DSCNODES parameter • 185
DSCOUT parameter • 185
DSCTRACE parameter • 185
DSN operand of @SMF macro • 279
DUMPINT operand of @SMF macro • 279

E

Equates, expanding • 279
ERROR
    autologging • 255
    DMSMA104S • 255
    file not found • 255
    invalid password message • 254
    IUCV • 254
    messages • 33
Error Messages • 33
eTrust CA-ACF2
    files
        copying command • 76
    intercepts, adding • 133
    options, processing • 66
    setup • 212
eTrust CA-ACF2 database groups, defining • 260
eTrust CA-ACF2 for VSE guest machine phase names • 286
Execute CAXABLD • 173
Exiting CA-ACTIVATOR • 33
Expanded Rule Size • 236

F

field
    logonid, defining • 260
    names, defining • 260
    UID, specifying • 260
Field Definition Record • 259
Filenames, defining for eTrust CA-ACF2 databases • 274
files
    CNTRL • 67
    CONTROL
        base • 163, 171
        component • 163, 171
        update • 163, 171
    copying • 76, 79
    not found errors • 255
    STARTUP OPTIONS • 121
    USERLID COPY • 100
    USERXLID COPY • 100
Final Installation Tasks • 143, 152
FLAGS operand of @CFDE macro • 262
For More Information • 13
format
    ACF LIST subcommand display • 277
    minidisks • 86
Formatting Service Machine Disks Manually • 88
Full Pack Scenarios • 223
Full-screen files, copying • 76

G

Generate
    CMS nucleus • 134, 163
    CP nucelus • 123
    test system • 61
Generate a Nucleus Using a Base Control File • 173
Generate the Production System • 143
Generating a CP and CMS Nucleus • 163
Generating the Message Table through ACFMSGEN • 203
Genlevel Tapes • 24
Group Logon Support • 157
GROUP operand • 262

- GSO LINUX records • 213

H

- HCPAF0 • 202
- HCPRIO, modifying • 223
- Header
  - blocks, SMF • 279
  - for ACF command, defining • 260
- heavy trace debugging • 206
- History files, SMF • 285
- host • 206

I

- IBM LOADLIST • 163
- Implementation • 203
- Implementing CA-CIS • 27
- Implementing Unit Record Logging Support • 243
- INCOMPLETE status description • 58
- Incorrect Installation Symptoms • 254
- INFOCLS operand • 262
- INFOFLG operand • 262
- Infostorage
  - database, defining filename • 274
  - records, updating SMF record number • 279
- Initial program load • 123
- Initializing VSAM databases • 235
- Insert Logonids • 152
- Install the Postbackup Service Machine • 160
- installation
  - assembling ACFFDR • 103
  - CMS
    - commands • 33
    - databases • 109
    - installation option • 72
  - converting CMS databases • 111
  - copying
    - ACF2VM MODULE • 98
    - ACFFDR TEXT • 105
    - ACFSTART EXEC • 98
    - command files • 76
    - files • 79
    - full-screen files • 76
  - CP commands • 33
  - defining CA-ACF2 service machine • 81
  - documentation files, loading • 62
  - formatting minidisks • 86
  - generating a test system • 61
  - minidisks • 85
  - miscellaneous options • 76
  - Missing Object
    - adding CA ACF2 intercepts • 133
    - applying • 291, 297, 300, 301, 303
    - generating CMS nucleus • 134, 163
    - generating CP nucleus • 123
    - generating test system • 38
    - IPLing • 123
    - loading the CA-CIS files • 38
    - locating the file • 38
    - merge VM directory • 115
    - mounting the tape • 38
    - preliminary CMS installation • 129
    - reassembling ACFFDR • 300
    - setting up STARTUP OPTIONS • 121
    - tailoring CA-EARL • 38
    - testing CMS nucleus • 134, 163
    - updating VMO records • 123
  - modifying
    - ACF2ASM EXEC • 93
    - ACFFDR • 103
    - CNTRL file • 67
    - MLAVM • 100
    - PROFILE EXEC • 95
    - USERLID COPY • 100
    - USERXLID COPY • 100
  - on POSIX systems • 123
  - PF keys • 32
  - procedure conventions • 59
  - processing eTrust CA-ACF2 file options • 66
  - setting up
    - CA-ACF2 service machine • 81
    - SMF minidisks • 107
  - status descriptions • 58
  - Task Selection Menu • 58
  - task summary • 59
  - tasks • 59
  - updating CP directory • 83
- Installation Options • 177
- Installation Prerequisites • 31
- Installation Procedure • 37
- installing
  - Database Synchronization Component • 180
- Installing Command Limiting and Message Logging Support • 195
- Installing Password Validation support. • 195
- Installing the CMS Batch Interface • 178
- Installing the Database Synchronization Component • 181

Installing VM Batch Facility Support • 244
Intercept
  adding to CMS modules • 133
  processing, bypassing • 170
Intercept Processing • 168
interfaces
  DASD Dump and Restore • 179, 180
  DIRMAINT
    Version 1 Release 4 command limiting • 194
    with ISF/CSE • 194
  Missing Object
    BATEXIT2 exit • 178
    installing • 178, 179
    Release 3.1 • 219
    submitting jobs • 179
    Version 3 Release 1 • 219, 220
  PASSTHRU Release 1.3 • 218
  TCP/IP FTP • 237
  TCP/IP NFS • 240
  TCP/IP REXEC • 241
  VM Batch Facility • 244, 250
  VM Batch system • 243
  VSAM AMSERV • 250
Interpreting Diagnostic Data • 256
Inter-User Communication Vehicle (IUCV) Support • 158
Invalid Password Message • 254
Invalid password message errors • 254
IPL • 123
ISF/CSE • 194
IUCV
  error • 254

K

KEYS • 24
Keys, PF • 32
KEYWORD
  CHECK
    CAXABLD • 174
  Missing Object
    APPEND • 167
    BYPASS • 170
    CLEAR • 167, 169
    CMSNUC • 165
    CPDEL • 166
    CPIRES • 165
    CPMOVIRS • 165
    CPMOVPAG • 165
    CPMOVRES • 165
    CPPAG • 165
    CPPAGMP • 165
    CPRES • 165
    CPRESMP • 165
    INTERCPT • 167, 168
    RENAMEXT • 169
    REPLACE • 167

L

language
  support • 197, 198, 202, 203, 204
LINUX profile record • 213
LIST parameter • 262
LMP
  using • 24
LNX command • 214
LOADLIST, IBM • 163
Local Logonid fields • 100, 262
loglevel • 206
Logonid
  adding fields • 100, 262
  changing SMF record number • 279
  compression algorithm, defining • 260
  database, defining filename • 274
  deleting SMF record number • 279
  group name, creating • 275
  inserting SMF record number • 279
  Missing Object
    defining • 260, 262
    displaying • 275
  record groups, default • 275
  SRF calls, specifying • 260
  supplied • 117
LPAR • 214

M

macros
  @CFDE • 261, 262
  @DDSN • 274
    enabling • 261
  @GNVMFDR • 275
  @GROUP • 261, 275
  @HEADER • 261, 277
  @LID • 279
  @MLID • 261, 277
  @RULE • 279
  @SETUP • 279

@SMF • 261, 279
@SRF • 261, 286
@SYSID • 288
@UID • 289, 296
ACFFDR • 260
ACFMDEST • 202
ACFMSGEN • 203
changing ACFFDR • 261
MLAVM • 100
VMXAOPTS • 259
Maintenance ID • 31
Maintenance ID Installation Steps • 115
Maintenance Philosophy • 23
MAXQUE operand of @SMF macro • 279
Merge VM directory • 115
Merging databases • 230
messages
error • 33
language support • 198
system • 34
Minidisk Configuration • 16
Missing Object
@DDSN macro • 274
@SMF macro • 279
@VALUES macro • 289
ACFRPTCL report • 243
ACFSERVE
ARCHIVE SMF • 285
CKPT SMF • 285
QUERY SMF • 285
SWITCH SMF • 285
adding CA ACF2 intercepts • 133
APPEND keyword • 167
applying • 291, 297, 300, 301, 303
bundling control statements • 170
CA-ESM • 29
CAXABLD • 173
CAXALOAD • 169, 174
CAXALOAD keyword • 165, 167, 168, 169
CCIVM parameter • 185
changing
access rules • 279
logonids • 279
CHECK keyword • 174
CLEAR keyword • 167, 169
CMS
generating • 134, 163
testing • 134, 163
COMPLETE • 58
compression algorithm for guest machines • 286
configuring • 204
control files
CMSNUC keyword • 165
control statement • 169
checking • 174
syntax • 164
copying
ACFFDR TEXT • 301
files • 297
CP, generating • 123
creating • 277
CSE/ISF complex • 195
customizing • 163
DASD Dump and Restore support • 179
DDR support • 179
defining • 81, 279
deleting
access rules • 279
logonids • 279
description • 171
displaying • 279
displaying values • 274
DSC • 185
DSC parameter • 185
DSCADMIN • 185
DSCADMIN parameter • 185
DSCFILE • 185
DSCFILE parameter • 185
DSCLOCAL • 185
DSCLOCAL parameter • 185
DSCNODES • 185
DSCNODES parameter • 185
DSCOUT • 185
DSCOUT parameter • 185
DSCTRACE • 185
DSCTRACE parameter • 185
enabling • 261
executing CAVMLOAD module • 173
expanding • 279
file options • 66
formatting • 86
function • 163
generating
CMS nucleus • 134, 163
CP nucleus • 123
using eTrust CA-ACF2 base control files • 173
implementing • 237, 240, 241, 243
implementing language support • 203

INCOMPLETE • 58
inserting logonids • 279
installation and configuration • 204
installing • 180
intercept processing bypassing • 170
invalid access attempts • 279
IPLing • 123
language support
    execs • 204
    LANG operand • 198
    messages • 198
links • 85
loading from tape • 62
logonid defining • 260
mapping • 260
merging VM directory • 115
minilogonid compression algorithm • 286
minilogonid for guest machines • 286
Missing Object
    installing command limiting • 194
mode, defining • 286
modifying • 230
nucleus generation • 173
OPEN • 58
OPTIONAL • 58
password violations • 279
perform MAINT tasks • 112
phase name • 286
preliminary CMS installation • 129
PREREQUISITE • 58
reassembling ACFFDR • 300
Release 3.1 • 219
REPLACE keyword • 167
RESETSTATS • 185
RESETSTATS parameter • 185
service machine • 86
setting up • 81, 107
SHOW
    DDSNS • 274
    FIELDS • 275
    SRF • 286
    STATE • 289
    SYSTEM • 279, 288
SMF • 107
SMF records • 242
specifying • 85, 260
specifying options • 259
SRF, defining • 286
ST • 185
ST parameter • 185
starting • 242
STARTUP OPTIONS file • 121
STATS • 185
STATS parameter • 185
STATUS • 185
STATUS parameter • 185
storing access rules • 279
syntax • 274, 275, 277
SYSOUT • 185
SYSOUT parameter • 185
testing CMS nucleus • 134, 163
tracking
    ACFSERVE command • 279
    command limiting violations • 279
    diagnose limiting violations • 279
    DIRMAINT violations • 279
tracking violations SMF record number • 279
updating
    infostorage records • 279
    resource database • 279
Version 1 Release 5 • 194
    command limiting • 195
    message logging • 195
    password validation • 195
Version 3 Release 1 • 219, 220
with ISF/CSE • 194
MLAVM macro • 100
MLID operand of @SRF macro • 277, 286
Mode of guest machines, defining • 286
MODE operand of @SRF macro • 286
MODIFY
    ACF2ASM EXEC • 93
    ACFFDR • 103
    CNTRL file • 67
    files
        USERLID COPY • 100
        USERXLID COPY • 100
    MLAVM macro • 100
    service machine • 95
Modify and Activate ACFFDR Changes • 261
modifying
    ACFFDR • 230
    HCPRIO • 223
    service machine ACFSTART EXEC • 230
    user directory • 223
MODULES
    adding CA ACF2 intercepts • 133
    deleting • 166

HCPAF0 • 202
MVFLAGS operand • 262
MVMAX operand • 262
MVMIN operand • 262

N

Name field • 262
National language messages • 198
National Language Support • 197
NOAUTO • 123
NOBACKUP • 123
NODBSYNC • 123
Notification file of SMF data • 279
NULL • 123
NULL option • 262

O

Obtain Control Files • 164
OPEN status description • 38, 58
Operating System Configuration and Service Level • 21
OPTIONAL status description • 58
options
ACFFDR, enabling • 261
for POSIX systems • 123
IPL • 123
processing • 66
startup • 123
OPTNAME operand of @SRF macro • 286
Output • 243

P

PAM Server Support • 204
Parameters • 185
Partial Pack Scenarios • 227
PASSTHRU Release 1.3 Support • 218
PASSTHRU support • 218
Password violation SMF record number • 279
PEM format • 206
PF keys • 32
PF Keys • 32
Phase name for eTrust CA-ACF2 for VSE guest machines • 286
POSIX installation options • 123
Preliminary CMS installation • 129
Preliminary Installation Steps • 38
PREREQUISITE status description • 58
Prerequisites • 181
PRG005 Error • 255
print packets • 206
Privileges of supplied logonids • 117
Procedure conventions • 59
Procedure Conventions • 59
PROFILE EXEC • 95
Program Temporary Fix • 24
PROMPT operand • 262
PRTN operand • 262
pseudo-random number generator (PNG) • 206
PSWD operand of @SMF macro • 279

R

Reassemble ACFFDR • 300
Reassemble SRF Applications • 152
records
logonid • 275
VMO, updating • 123
Refresh eTrust CA-ACF2 • 291, 297, 300, 301, 303
Refresh Tasks • 296
REGISTER • 220
REINIT Command • 216
Reloading ACFFDR • 230
Rename External References • 169
REPLACE • 167
Required BATEXIT2 Exit • 178
Required CP Minidisks • 21
Requirements for Sharing on a Full Pack Minidisk • 223
Requirements for Sharing on a Partial Pack Minidisk • 227
RESETSTATS parameter • 185
resource
database, updating SMF record number • 279
resource rules • 214
Response at IPL • 123
Restart the CA ACF2 for z/VM Service Machine • 161
RRTN operand • 262
RSCS System Component • 219
RSRC operand of @SMF macro • 279
Rule database, defining filename • 274
rules
changing • 279
deleting • 279
storing • 279

S

sample

ACF2DEFC AMSERV file • 233
ACFLINIT EXEC • 235
UCENG component control file • 173
Sample CAXALOAD CHECK Summary Log for CP • 175
Sample CAXALOAD Summary Log for CP • 175
Sample CMS Base Control File • 172
Sample CP Base Control File • 172
Sample UCENG Optional Component File • 173
Secure Socket Layer (SSL) • 206
server certificate • 206
SERVICE
machine
defining • 81
minidisks • 85
modifying ACFSTART EXEC • 230
PROFILE EXEC • 95
setting up • 81
Service machine
DSC • 180
SET Command • 217
set up
CA-ACF2 service machine • 81
CMS databases • 109
SMF minidisks • 107
STARTUP OPTIONS file • 121
Shared Database Considerations for VM-Only Sites • 231
Shared Database Support • 222
shared databases
activating • 222
considerations • 231
full pack minidisk sharing • 223
modifying
HCPRIO • 223
user directory • 223
partial pack minidisk sharing • 223
prerequisites • 222
sharing
full pack minidisks • 223
partial pack minidisks • 223
with &gmaop. • 223
with guest machines • 223
with processors • 223
shared file system
CA-ESM • 29
Shared File System Prerequisites • 15
sharing
&gmaop. • 223
full pack minidisk • 223
partial pack minidisks • 223
with &gmaop. • 223
with other guest machines • 223
with other processors • 223
SHOW subcommand
DDSNS • 274
FIELDS • 275
SRF • 286
STATE • 289
SYSTEM • 279, 288
SHUTDOWN • 123
SMF
changing
access rules record number • 279
logonids record number • 279
data
blocks, specifying • 279
receiving • 279
defining user ID • 279
deleting
access rule record number • 279
logonid record number • 279
displaying record numbers • 279
files
flagging as history • 285
forcing a checkpoint • 285
switching • 285
IDs
default • 279
defining • 279
inserting logonid record number • 279
invalid access attempt record number • 279
Missing Object
addresses, displaying • 285
setting up • 107
status, displaying • 285
notification file • 279
outstanding write requests • 279
password violation record number • 279
receiving data • 279
record for unit record logging support • 242
record numbers
defining • 279
specifying • 260
specifying punched records • 279
storing access rule record number • 279
SWITCH
initiating • 279
processing • 279

tracking
    ACFSERVE command record number • 279
    command limiting violation record number • 279
    diagnose limiting violation record number • 279
    DIRMAINT violation record number • 279
updating
    infostorage record record number • 279
    resource database record number • 279
writing updates • 279
SMFID operand of @SMF macro • 279
SMSG Command • 184
Special Considerations • 156
specify
    CMS option • 72
    minidisks • 85
Spooling Initialization is Complete • 254
SRF
    calls for logonids, specifying • 260
    guest machines, defining • 286
SSL_Required option • 206
ST parameter • 185
Startup options • 123
STARTUP OPTIONS file, setting up • 121
STATS parameter • 185
STATUS Command • 217
STATUS operand • 262
STATUS parameter • 185
Step 1
    Create the PAM Server Service Machine • 205
    Create VSAM Minidisk • 231
    Defining the Linux for zSeries Hosts • 213
    Merge VM Directory into the Logonid Database • 115
    Modify the User Directory and HCPRIO • 223
Step 2
    Add the CA PAM Server to SYSTEM DTCPARMS • 206
    Create VSAM Master Catalog • 232
    Defining the Linux for zSeries Group Profile Records • 213
    Modify the Service Machine ACFSTART EXEC • 230
    Modify VMXAOPTS • 118
Step 3
    Define VSAM Databases • 233
    Defining the Linux for zSeries User Profile Records • 214
    Modify the TCPIP Configuration File • 206
    Modify, Assemble, and Reload the ACFFDR • 230
    Set Up STARTUP OPTIONS File • 121
Step 4
    Create the PAM Server Configuration File • 206
    Defining the Linux for zSeries Resource Rules • 214
    Generate the CP Nucleus, IPL, and Update VMO Records • 123
    Initialize VSAM Databases • 235
    Merge the Databases Using the ACFDBVSM Utility • 230
Step 5
    Convert VM Directory • 236
    Create the SRVRPAM EXEC • 211
    Define Secondary Group Membership • 215
    Preliminary CMS Installation Steps • 129
Step 6
    Add CA ACF2 for z/VM Intercepts to CMS Modules • 133
    Define the PAM Server ID to Security • 212
Step 7
    Generate and Test the CMS Nucleus • 134
Step 8
    Generate Module Files For CMS Commands • 136
Step 9
    The Final IPL with CA ACF2 for z/VM • 137
Stop Command • 215
Storage Requirements • 22
Subcode, DIAG A0 • 193
Submitting Jobs on Behalf of Another User • 179
Supplied CA ACF2 for z/VM Logonid Records • 117
Supplied logonids • 117
Support for DirMaint • 194
SWITCH operand of @SMF macro • 279
Switch SMF file • 279, 285
SYMBOL operand • 262
syntax
    @CFDE macro • 262
    @DDSN macro • 274
    @GNVMFDR macro • 275
    @GROUP macro • 275
    @HEADER macro • 277
    @MLID macro • 277
    @SETUP macro • 279
    @SMF macro • 279
    @SRF macro • 286
    @SYSID macro • 288
    @UID macro • 289, 296

  - BYPASS • 170
  - bypassing intercept processing • 170
  - Missing Object
    - adding modules • 165
    - APPEND keyword • 167
    - CLEAR keyword • 167
    - defining intercepts • 167
    - deleting modules • 166
    - relocating modules • 165
    - REPLACE keyword • 167
  - RENAMEXT, CLEAR keyword • 169
- SYSID • 123
  - defining • 260, 288
  - displaying • 288
- SYSOUT parameter • 185
- SYSTEM
  - messages • 34
- System Messages • 34
- System Prerequisites • 21
- System Requirements • 15

T

- Task M9C0I000
  - Specify If New Install or Upgrade • 63
- Task M9C0I001
  - CA-ACTIVATOR Product Disk Assignment • 64
- Task M9C0I002
  - Specify Maintenance Minidisks • 4, 65
- Task M9C0I003
  - Option File Processing • 66
- Task M9C0I004
  - Modify CA ACF2 for z/VM CNTRL File • 67
- Task M9C0I005
  - Specify CP Installation Option • 68
- Task M9C0I006
  - Modify CP CNTRL File • 69
- Task M9C0I007
  - Specify CMS Installation Option • 72
- Task M9C0I008
  - Modify CMS CNTRL File • 73
- Task M9C0I009
  - Specify Miscellaneous Installation Options • 76
- Task M9C0I010
  - Copy Files to Test System Disks • 79
- Task M9C0I011
  - Create CA ACF2 for z/VM Service Machine Directory Entry • 81
- Task M9C0I012
  - Update CP Directory • 83
- Task M9C0I013
  - Specify Service Machine Minidisk Links • 85
- Task M9C0I014
  - Format the Service Machine Minidisks • 86
- Task M9C0I015
  - Create/Edit ACFCP CAXALOAD File • 89
- Task M9C0I017
  - Edit CA ACF2 for z/VM $PPF File • 4, 91
- Task M9C0I018
  - Modify ACF2ASM EXEC • 93
- Task M9C0I019
  - Modify the Service Machine PROFILE EXEC • 95
- Task M9C0I020
  - Copy ACF2VM MODULE, ACFSTART EXEC, and CALMP KEYS • 98
- Task M9C0I021
  - Modify MLAVM, USERLID, and USERXLID COPY • 100
- Task M9C0I023
  - Modify and Assemble ACFFDR • 103
- Task M9C0I024
  - Copy ACFFDR TEXT to Service Machine • 105
- Task M9C0I025
  - Set Up SMF Minidisks • 107
- Task M9C0I026
  - Set Up CMS Databases • 109
- Task M9C0I027
  - Convert CMS Databases to VSAM Databases • 111
- Task M9C0I032
  - Perform MAINT Tasks • 112
- Task M9C0I033
  - Execute ACFCVSFS Utility • 141
- Task M9C0I034
  - Authorization of SFS Service Machines • 142
- Task M9C0I035
  - Activate SFS External Security • 143
- Task M9C0I90S
  - Selection of CA-CIS Services • 61
- Task M9C0IDOC
  - Load Documentation Files • 62
- Task M9C0P001
  - Copy CA ACF2 for z/VM Files to Production • 144
- Task M9C0P002
  - Perform Production MAINT Tasks • 151
- Task M9C0R010
  - Copy Files to Test System Disks • 297
- Task M9C0R020

Replace ACF2VM MODULE • 299
Task M9C0R023
Reassemble ACFFDR • 300
Task M9C0R024
Copy ACFFDR TEXT to Service Machine • 301
Task M9C0R032
Perform MAINT Tasks • 303
Task Selection Menu • 58
TCP/IP FTP Interface • 237
TCP/IP NFS Interface • 240
TCP/IP REXEC Interface • 241
Terminology • 16
Test
CMS nucleus • 134, 163
system installation tasks • 59
Test System Installation Tasks • 59
threads option • 206
TIME operand of @SMF macro • 279
TIMER operand of @SMF macro • 279
TLSCACertificateFile option • 206
TLSCertificateFile option • 206
TLSCertificateKeyFile option • 206
TLSRandomFile option • 206
TLSVerifyClient option • 206
trace all socket I/O function calls • 206
trace function calls • 206
Transport Layer Security (TLS) • 206
TRIM operand • 262
TRIVIA display mode • 262
troubleshooting
collecting diagnostic data • 256
Troubleshooting • 251
TYPE operand of @CFDE macro • 262
Types of CA ACF2 for z/VM Control Files • 163

U

UID
defining • 289, 296
displaying • 289
fields, specifying • 260
Undefined HCPAL0 and HCPAL0MC References • 255
Unit Record Logging Support • 242
UPDATE
CP directory • 83
VMO records • 123
User Diagnose Requirements • 23
User fields in logonid record • 100, 262
userid option • 206
USERLID COPY • 100, 262
USERXLID COPY • 100, 262
Using CA LMP • 24
Using the Database Synchronization Component • 184
Utilities, ACFLIDGN • 115

V

VER operand • 262
violations
command limiting SMF record number • 279
diagnose limiting SMF record number • 279
DIRMAINT SMF record number • 279
Virtual Machine Communications (VMCF) Support • 160
VM Batch Facility, installing • 244, 250
VM Batch System Support • 243
VM specification of ACFFDR @MLID macro • 277
VMDBK Field Requirements • 23
VMO records
updating • 123
VMSAF logonid attribute • 193
VPRM1 operand • 262
VPRM2 operand • 262
VRTN1 operand • 262
VRTN2 operand • 262
VSAM
installing AMSERV support • 250
VSAM AMSERV Support • 250
VSAM Database Configuration • 20
VSE specification of ACFFDR @MLID macro • 277

W

Welcome • 11
Write requests, outstanding SMF • 279

X

XSYMBOL operand • 262
XTYPE operand • 262

Z

z/VM systems
DASD Dump and Restore support • 179
DDR support • 179
ZERO operand • 262