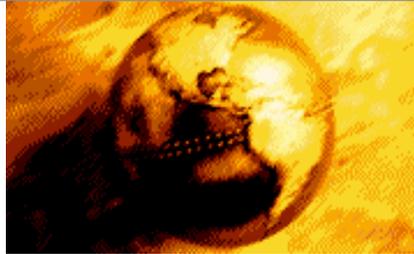


# Installation Guide

1010101010101011  
1010101010101011

Release 9.0.2  
October 2000

011  
1001



# CA-Disk™

2000 Computer Associates International, Inc.  
All Rights Reserved

Computer Associates International, Inc.  
Storage Management Division  
625 East Carnegie Drive, Suite 200  
San Bernardino, California 92408-3510

01DS100P009-902

## **Proprietary/Trade Secret Notice and Warranty Disclaimer**

This documentation and the software it describes, constitute the proprietary, confidential, and valuable trade secret information of Computer Associates International, Inc., and is licensed either "AS IS" or with a limited warranty, as set forth in the applicable license agreement. NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE.

## **Right to Revise**

CA-Disk is undergoing constant revision, refinement, and expansion to include additional features. The current documentation is believed to be accurate and reflects the current version of the software. Important information not available when this book was published is contained in the Cover Letter that accompanied the distribution tape.

Computer Associates International, Inc., reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes. No part of this Installation Guide can be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, for any purpose, without the express written permission of Computer Associates, Inc.

Use, duplication or disclosure by the Government is subject to restrictions as set forth in Sections 52.227-14 and 52.227-19 of the FAR Commercial Computer Software Restricted Rights Clause and/or the Rights in Technical Data and Computer Software Clause at DFAR 252.227.7013 as applicable.

© 2000 Computer Associates International, Inc., All Rights Reserved.

Printed in the USA

## **Customer Support**

In the United States, contact:  
Computer Associates International, Inc.  
OS/390 Storage  
10811 International Drive  
Rancho Cordova, California 95670  
(800) 889-0226

Outside the United States, contact your local Computer Associates customer support representative.

---

# Contents

---

## CHAPTER 1

<i>Preinstallation</i> .....	7
CA-Disk System Installation Summary .....	7
<i>Installation</i> .....	8
<i>Customization</i> .....	9
Installation Planning .....	10
<i>Item 1: Disk Space and Naming Conventions</i> .....	10
<i>Item 2: APF-Authorized Libraries</i> .....	10
<i>Item 3: User SVC</i> .....	11
<i>Item 4: Auto-Restore</i> .....	11
<i>Item 5: Linklist Library</i> .....	11
<i>Item 6: LPA List Library</i> .....	11
<i>Item 7: Help Library</i> .....	11
<i>Item 8: Parameter Library</i> .....	12
<i>Item 9: Tailoring and Other Considerations</i> .....	12

## CHAPTER 2

<i>Installation</i> .....	13
Preparing the Environment .....	14
<i>Step 1. Distribution Tape Structure</i> .....	14
<i>Step 2. Preparing for Multiple Releases of CA-Disk</i> .....	14
<i>Step 3. Download the Related Installation Materials Library</i> .....	14
<i>Step 4. Prepare the SMP/E Environment</i> .....	15
<i>Step 5. Allocate CA-Disk data sets</i> .....	15
<i>Step 6. Set DDDEF Entries</i> .....	15
<i>Step 7. Receive CA-Disk</i> .....	15
<i>Step 8. Apply CA-Disk</i> .....	16
<i>Step 9. Accept CA-Disk</i> .....	16
<i>Step 10. Apply Maintenance</i> .....	16
Activating the CA-Disk System .....	17
The CA-Disk Parameter Library (Parmlib) .....	17

Step 1. Identify Files, Parmlib, and Activation Codes .....	18
SET Command Syntax .....	18
The CA-Disk JCL Procedures .....	20
Step 2. Modify Symbolic Parameters in JCL Procedures .....	22
Step 3. Customize JCL Procedures .....	22
Step 4. Create a Files Definition Member in Parmlib .....	23
Step 5. Initialize the Files data set .....	23
Step 6. Verify CA-Disk Enqueue Usage .....	24
Activating CA-Disk Features .....	25
Activating CA-Disk Security Features and Interfaces .....	25
Installing CA-Disk Under CA-ACF2 .....	26
Installing the CA-ACF2 Security Interface .....	27
Installing CA-Disk Under CA-Top Secret .....	28
Installing the CA-Top Secret Security Interface .....	29
Installing CA-Disk Under RACF .....	30
Installing the RACF Security Interface .....	30
Activating Miscellaneous Security Features .....	33
Parmlib Security — PARMAUTH .....	33
USERMOD5 .....	35
System Parameter Override Security — SYSPAUTH .....	36
Other Usermods .....	40
Activating MTFDEBUG DD Statement Security Feature .....	40
Activating VSAM Support .....	41
Activating ISPF Support .....	43
Dynamic Menu-Formatting Feature .....	43
Formatting Used for On-line Reporting .....	43
Connecting the ISPF Libraries to Your System .....	45
Customizing the CA-Disk ISPF Support .....	45
Activating TSO Support .....	46
Step 1. Make the Command Processor Available to TSO .....	46
Step 2. TSO Help Text .....	47
Step 3. The TSO Dynamic Restore Command Processor .....	47
Activating DASD Billing .....	48
Activating the DBRC Interface .....	48
To Activate the DBRC Support .....	49

## CHAPTER 3

## *Data Set Customization..... 51*

The CA-Disk SVC .....	52
CA-Disk SVC Modes .....	53
Tailoring the CA-Disk SVC .....	54
Installing the CA-Disk SVC .....	54
The CA-Disk VSAM Date Stamp .....	57
Installing the IDATMSTP Module .....	58
Testing the Installation of the IDATMSTP Module .....	58
The Auto-Restore Function .....	58
The Two Auto—Restore Methods .....	59
Conceptual Diagram of CA-Disk Auto-Restore Methods .....	60
Installing the Auto-Restore Function .....	61

Test Auto-Restore Hook Interface .....	64
Overview .....	64
How it Works .....	65
Components of the Test Auto-Restore Hook Interface .....	66
Preparing to Use the Test Auto-Restore Hook Interface .....	66
Installing the Test Auto-Restore Hook Interface .....	67
Operating the Test Auto-Restore Hook Interface .....	68
Setting Up DASD Pools for Auto-Restore .....	70
Customizing the TSO/ISPF Auto-Restore Environment .....	77
USERMOD8 .....	79
Auto-Restore Implementation Guidelines .....	80
VSAM Alternate Indexes .....	81
Restrictions .....	81
Auto-Restore Delayed by Merge .....	81
Propagating Enqueues .....	81
Additional Considerations .....	82
Installation of IGGDASU2 User Exit .....	84
Customizing the CA-Disk Tape Management Support .....	85
Method 1 — Controlling Tapes Via the EDM .....	86
Method 2 — Controlling Tapes by Expiration Date .....	88
Method 3 — Controlling Tapes by Catalog Status .....	88
Method 4 — Controlling Tapes Using TLMS/EDM .....	89
CA-Disk/CAI Interfacing Considerations .....	89
Capabilities of the ISPF Interface .....	89
Defining ISPF and DSCL User Options .....	89
Customizing the CA-Disk TSO Support .....	94
Tape Units Allocated Concurrently .....	94
Tape Unit Name .....	94
Limiting TSO User Access .....	94
TSO Screening Exits for Archive and Restore .....	95
TSO Command and ISPF/PDF Task Library Override - TASKLIB .....	95

**CHAPTER 4**

**Evaluation..... 96**

Testing the CA-Disk System .....	96
Step 1. Run Some CA-Disk Reports .....	97
Step 2. Archive Some data sets .....	99
Step 3. Restore the data sets .....	100
Step 4. Submit a Backup Job .....	101
Step 5. Produce a LISTD/LISTV Report .....	102
Step 6. Run a MERGE Job .....	102
Step 7. Recover a DASD Volume .....	104
Step 8. Compress a PDS .....	104
Step 9. Release Idle Space .....	106
Special Testing Environment .....	106
Conclusion of Evaluation .....	107
Testing the CA-Disk Auto-Restore Feature .....	107
Forcing an Auto-Restore to be Initiated .....	108

---

<b>APPENDIX A</b>	<b><i>DSCB Update Utility</i></b> .....	<b>109</b>
	Overview .....	<b>109</b>
	<i>Caution</i> .....	<b>109</b>
	<i>Condition Codes</i> .....	<b>110</b>
	JCL — DSCB Command .....	<b>110</b>
	SYNTAX .....	<b>110</b>
	User Exits .....	<b>114</b>
<b>APPENDIX B</b>	<b><i>IXCATLG Utility</i></b> .....	<b>115</b>
	IXCATLG Function Description .....	<b>115</b>
	<i>IXCATLG Command and Parameters</i> .....	<b>116</b>
	<i>IXCTLGEX - IXCATLG User Screening Exit</i> .....	<b>117</b>
<b>APPENDIX C</b>	<b><i>Files Data Set Conversion</i></b> .....	<b>118</b>
<b>APPENDIX D</b>	<b><i>CA-Disk SVC</i></b> .....	<b>121</b>
	Conversion for the CA-Disk SVC .....	<b>122</b>
	<i>Conversion within the CA-Disk SVC</i> .....	<b>123</b>
	<i>Conversion within Normal CA-Disk Processing</i> .....	<b>124</b>
	<i>Conversion Using the DSCB Update Utility</i> .....	<b>124</b>
	<i>Running Release 9.0.2 with a Previous Release's CA-Disk SVC</i> .....	<b>125</b>
<b>APPENDIX E</b>	<b><i>S213 Abend Exit</i></b> .....	<b>126</b>
	“F1-DSCB Not Found” S213 Abend Exit (IFG0EX0A Exit) .....	<b>126</b>
	<i>Installing the S213 Abend Exit</i> .....	<b>127</b>
<b>APPENDIX F</b>	<b><i>Message Check Utility</i></b> .....	<b>129</b>
	Installing CA-Disk Message Check .....	<b>129</b>
	Implementation .....	<b>136</b>

---

CA-Disk executes on IBM (or compatible) processors running under MVS operating systems. Other operating systems are formally supported only if they are fully compatible with MVS.

This guide covers installation, customization, evaluation and maintenance of the Computer Associates Data Storage Management System, CA-Disk. An overview that outlines the steps is presented below. Each step contains detailed instructions.

You will probably follow most of the steps, but some of them will be skipped depending on your particular data center. For example, you can install the CA-Disk interface to the CA-ACF2 security system or to the RACF security system, but not to both. So you will not use at least one of those installation steps.

---

## *CA-Disk System Installation Summary*

Installation Planning

**Planning Item 1:** - Disk Space and Naming Conventions

**Planning Item 2:** - APF-Authorized Libraries

**Planning Item 3:** - User SVC

**Planning Item 4:** - Auto-Restore

**Planning Item 5:** - Linklist Library

**Planning Item 6:** - LPA List Library

**Planning Item 7:** - Help Library

**Planning Item 8:** - Parameter Library

**Planning Item 9:** - Tailoring and Other Considerations

## Installation

### Preparing the Environment

- Step 1.** - Distribution Tape Structure
- Step 2.** - Preparing for Multiple Releases of CA-Disk
- Step 3.** - Download the Related Installation Materials Library
- Step 4.** - Prepare the SMP/E Environment
- Step 5.** - Allocate CA-Disk data sets
- Step 6.** - Set DDDEF Entries
- Step 7.** - Receive CA-Disk
- Step 8.** - Apply CA-Disk
- Step 9.** - Accept CA-Disk
- Step 10.** - Apply Maintenance

### Activating the CA-Disk System

- Step 1.** - Identify Files, Parmlib, and Activation Codes
- Step 2.** - Modify Symbolic Parameters in JCL Procedures
- Step 3.** - Customize JCL Procedures
- Step 4.** - Create a Files Definition Member in Parmlib
- Step 5.** - Initialize the Files data set
- Step 6.** - Verify CA-Disk Enqueue Usage

### Activating CA-Disk Features

- Step 1.** - Activating CA-Disk Security Features and Interfaces
  - Select Password-Indicated data set Support Options
  - Activate CA-Disk System Security
  - Prepare Security Package to Work With CA-Disk
  - Install CA-Disk Security Interface
- Step 2.** - Activating VSAM Support
- Step 3.** - Activating ISPF Support
- Step 4.** - Activating TSO Support
- Step 5.** - Activating DASD Billing

**Step 6.** - Activating the DBRC Interface

## **Customization**

### **The CA-Disk SVC**

**Step 1.** - CA-Disk SVC Modes

**Step 2.** - Tailoring the CA-Disk SVC

**Step 3.** - Installing the CA-Disk SVC

### **The VSAM Date Stamp**

#### **The Auto-Restore Function**

**Step 1.** - The Two Auto-Restore Methods

**Step 2.** - Setting up Auto-Restore DASD Pools

- VSAM data sets
- Non-VSAM data sets

**Step 3.** - Customizing the TSO/ISPF Auto-Restore Environment

**Step 4.** - Auto-Restore Implementation Guidelines

**Step 5.** - Auto-Restore Via VSAM Alternate Indexes and Path Names

**Step 6.** - Auto-Restore Restrictions

**Step 7.** - Additional Auto-Restore Considerations

#### **Customizing the CA-Disk Tape Management Support**

**Step 1.** - Controlling Tapes Via the External Data Manager Interface of CA1

**Step 2.** - Controlling Tapes By Expiration Dates

**Step 3.** - Controlling Tapes By Catalog Status

#### **Customizing the ISPF Support (ISPF and DSCL User Options)**

#### **Customizing the TSO Support**

**Step 1.** - Tape Units Concurrently Allocated

**Step 2.** - Tape Unit Name

**Step 3.** - Limiting TSO User Access

**Step 4.** - TSO Screening Exits for Archive and Restore

## Evaluation

**Step 1.** - Testing the CA-Disk System

**Step 2.** - Testing the Auto-Restore Feature

---

## *Installation Planning*

The installation process proceeds much more smoothly with a little advance preparation. Consider the following items before starting the actual installation.

### **Item 1: Disk Space and Naming Conventions**

Make available enough disk space to load and install the system, and decide on a naming convention for the CA-Disk data sets. Disk space estimates for all libraries are documented in the program directory file on the installation tape.

### **Item 2: APF-Authorized Libraries**

Establish two APF authorized libraries for CA-Disk.

1. Your CA-Disk Loadlib
2. Your CA-Disk LPA List Library

Doing this in advance permits you to test executions immediately after downloading the distribution tape. Otherwise, you have to wait for an IPL to authorize the libraries, or you can specify the following operator command to authorize the libraries dynamically:

```
T PROG=nn
```

### Item 3: User SVC

Make an entry for the SAMS Disk SVC in the IEASVCxx member of your system parmlib. It should be a type 3 or type 4 SVC, enabled for interrupts, Designating as either restricted or non restricted is optional. Detailed information for installing the CA-Disk SVC can be found in [“Installing the CA-Disk SVC” on page 54](#).

### Item 4: Auto-Restore

#### CSA Requirements

Ensure that approximately 41K bytes of ECSA (Extended Common Service Area) is available in your operating system if you are planning to install the CA-Disk Auto-Restore function. Detailed information for installing the CA-Disk Auto-Restore catalog management hook can be found in [“The Two Auto—Restore Methods” on page 59](#).

#### JES Awareness

Auto-Restores are automatically bypassed for all jobs starting with JESnnnnn. This is done to automatically exclude JES2 and JES3 from invoking an Auto-Restore that could not be done prior to starting JES, resulting in hanging the system.

### Item 5: Linklist Library

Add the DMSLINK library to the LNKLSTxx member of your system parmlib. If you plan to copy the DMSLINK library into an existing linklist library, make sure that approximately 45k bytes of disk storage are available in that library.

### Item 6: LPA List Library

Add the DMSLPA library to the LPALSTxx member of your system parmlib. If you plan to copy the DMSLPA library into an existing LPA list library, make sure that approximately 8k bytes of disk storage are available in that library. The DMSLPA library contains the CA-Disk SVC.

### Item 7: Help Library

If you are installing the TSO interface, either concatenate the DMSHELP library to the //SYSHELP DD in your TSO logon JCL, or if you plan to copy the DMSHELP library into your system TSO help library, make sure that approximately 6k bytes of disk storage are available for six members that must reside in that library. These members are required for the TSO interface.

## Item 8: Parameter Library

Save a copy of your PARMLIB so that you can copy your user-defined members after the CA-Disk installation process is complete.

## Item 9: Tailoring and Other Considerations

Review the options and suggestions in [“Tailoring Options” on page 52 in the Systems Guide](#) because there are some additional options and considerations that you will probably want to review. Some important topics include:

- [“Archive/Backup Considerations” on page 57 in the Systems Guide](#)
- [“Processing PDSs that Contain Anomalies” on page 75 in the Systems Guide](#)
- [“Implementing Support for StorageTek Redwood” on page 82 in the Systems Guide](#)
- [“Implementing Support for IBM's Magstar” on page 84 in the Systems Guide](#)
- [“ISPF Custom Reports” on page 85 in the Systems Guide](#)
- [“User-Specified Condition Codes” on page 89 in the Systems Guide](#)

Additionally, review sections [“Special Considerations” on page 9 in the User’s Guide](#) and [“General Restrictions” on page 19 in the User’s Guide](#), and implement any special processing controls indicated for your installation's environment.

---

This section of the CA-Disk Installation Guide provides step-by-step instructions for installing the basic CA-Disk system and activating various functions. CA-Disk installation is done with SMP/E using the Receive-Apply-Accept method. SMP/E invokes the linkage editor and IEBCOPY to download the distribution tape, and to install it into the target and distribution libraries.

We have provided sample jobs to set up a complete SMP/E environment. If you have already established an SMP/E environment, skip the steps you do not need to perform.

All of the data sets you will need in order to install SAMS products with SMP/E will be internally defined as DDDEF entries. There is no need for an SMP/E proc. All you need to do is execute the SMP program using "PARM=CSI=..." as documented in the SMP/E manual. You can also use the ISPF interface provided with SMP/E.

This process installs the full version of CA-Disk release 9.0.2 under the FMID of SDM1900.

---

*Note:* **Note:** CA-Disk v9.0.0 is also installed with the FMID of SDM1900. The full version of release 9.0.2 can not be installed into SMP/E Global, Target or Distribution zones that already contain CA-Disk 9.0.0.

---

The CA-Disk SVC Zaps will be installed under FMID SDZ1110, and the library of customer supplied Usermods under SDU1110. This permits you to install future releases of CA-Disk without reinstalling the SVC Zaps or Usermods.

---

*Note:* Do not install CA-Disk into the same SMP/E zones as any other vendor product on your operating system because naming conflicts can occur.

CA-Disk OS/390 Edition, (release 9.0.2 and above) can be applied into the same target and distribution zones as CA-Vantage (release 4.0 and above) as long as PTF SV01428 is applied to CA-Vantage. All CA-Disk and CA-Vantage sysmod dependencies will be indicated through IFREQ processing and dependency holds. The CA-Vantage RIMLIB and PGMDIR data sets have been updated to note the adjustments required.

CA-Disk can not be installed into the same zones as CA-Allocate. Naming conflicts still exist between CA-Disk and CA-Allocate.

---

## Preparing the Environment

### Step 1. Distribution Tape Structure

CA-Disk is distributed on a standard label tape. The VOLSER of the distribution tape and its format is documented in the current CA-Disk Cover Letter.

### Step 2. Preparing for Multiple Releases of CA-Disk

You can install this release of CA-Disk and continue to use an older release for your production environment. If you plan to run a previous release, consider the following:

1. Installing this release into the same zone as another release causes the previous release to be deleted. The target and distribution zones of the releases must differ if you are to keep multiple releases.
2. Be sure you do not install the new release modules into the same libraries as the old release modules. Doing so causes a compatibility problem with one or both releases. To avoid this problem, ensure the DDDEF entries in your new zones point to the new release libraries.
3. If you are installing CA-Disk 9.0.2 into an existing CA-Disk SMP/E environment, you need to create DDDEF entries for some new libraries that previous releases did not have. Review the DSK01ALC and DSK02DDD members in RIMLIB to identify the new job steps that are to be executed separately prior to RECEIVE processing.

### Step 3. Download the Related Installation Materials Library

The JCL needed to install CA-Disk is provided in the Related Installation Materials library (RIMLIB) on the distribution tape (file 2). Important information related to the installation process is provided in the Program Directory library (PGMDIR) on file 4. Run the following JCL to allocate both of these libraries and download them from tape.

```
//COPY      EXEC PGM=IEBCOPY,REGION=4096K
//SYSPRINT  DD  SYSOUT=*
//SYSUT1    DD  DSN=RIMLIB,DISP=OLD,UNIT=TAPE,VOL=SER=vvvvvvv,
//          LABEL=(2,SL)
//SYSUT2    DD  DSN=your.RIMLIB,DISP=(,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(15,3,6),RLSE)
//SYSUT3    DD  UNIT=SYSDA,SPACE=(CYL,1)
//SYSIN     DD  DUMMY
/*
//S1        EXEC PGM=IEBCOPY,REGION=512K
//SYSPRINT  DD  SYSOUT=*
//SYSUT3    DD  SPACE=(CYL,(2,2)),DISP=(NEW,DELETE),UNIT=SYSDA
//SYSUT4    DD  SPACE=(CYL,(2,2)),DISP=(NEW,DELETE),UNIT=SYSDA
//SYSUT1    DD  DSN=PGMDIR,DISP=SHR,LABEL=(4,SL),
//          VOL=SER=vvvvvvv,UNIT=3480
//SYSUT2    DD  DSN=YOUR.PGMDIR,
//          UNIT=SYSDA,DCB=(RECFM=FBA,BLKSIZE=3120,LRECL=80),
//          SPACE=(3120,(20,5,27),,ROUND),DISP=(NEW,CATLG)
//SYSIN     DD  DUMMY
```

---

*Note:* SMP/E is equipped with ISPF panels that can assist you in downloading the RIMLIB and PGMDIR libraries.

---

The RIMLIB data set contains jobs to install the base product with SMP/E. Customization jobs are still located in the installation library.

Full Product Distribution Tapes contain the base release and SMP/E format PTFs. If there are PTFs on the distribution tape, you can receive them at the same time that you receive the base product. However, you should apply and accept the base release before you apply any PTFs.

## Step 4. Prepare the SMP/E Environment

Customize and submit RIMLIB member SMP01GBL to define the global CSI data set, allocate the SMPPTS, and SMPLOG data sets, and initialize the global zone. If you are upgrading from release 8.2 and want to use the same global zone, skip job SMP01GBL and proceed to job SMP02TGT.

Customize and submit RIMLIB member SMP02TGT to define the target CSI data set, allocate the SMPMTS, SMPSCDS, and SMPSTS data sets, and initialize the target zone.

Customize and submit RIMLIB member SMP03DLB to define the distribution CSI data set, and initialize the distribution zone.

## Step 5. Allocate CA-Disk data sets

Customize and submit RIMLIB member DSK01ALC to create target and distribution data sets for CA-Disk.

## Step 6. Set DDDEF Entries

Customize and submit RIMLIB member DSK02DDD to create DDDEF entries for CA-Disk in the target and distribution zones.

## Step 7. Receive CA-Disk

Customize and submit RIMLIB member DSK03REC to receive CA-Disk. If there are PTFs on the distribution tape, they will also be received at this time. Do not apply the PTFs until after you have accepted the base function for CA-Disk.

You can use RIMLIB member RECEIVE to receive all sysmods and hold data on the distribution tape.

## Step 8. Apply CA-Disk

Customize and submit RIMLIB member DSK04APP to apply the base function for CA-Disk.

## Step 9. Accept CA-Disk

Customize and submit RIMLIB member DSK05ACC to accept the base function for CA-Disk.

## Step 10. Apply Maintenance

Maintenance to CA-Disk is shipped on your distribution tape and is received at the same time as the Base function. Apply this maintenance now.

Customize and submit RIMLIB member DSK06PTF to apply the PTFs. You can accept the PTFs according to your own installation's policy.

When you apply maintenance, you normally encounter SMP/E hold data. Computer Associates uses hold data to notify your SMP/E system of sysmods that have errors or special conditions.

There are two different types of hold data:

1. **Internal hold data** — Data that is an instream part of the sysmod instructing you of special conditions:

ACTION	You must perform special processing either before or after you apply this sysmod.
DEP	There is a dependency for this sysmod that you must externally verify.
DELETE	This sysmod deletes a load module. You cannot reverse this type of sysmod with the SMP/E RESTORE command.
DOC	There is a documentation change with this sysmod.
EC	This sysmod requires a hardware engineering change. An EC hold sysmod usually does not have an effect on the product unless the EC is present on the hardware device.
UCLIN	You need to perform a UCLIN either before or after you apply this sysmod.

Code a bypass operand on your APPLY command to install sysmods that have internal holds. Only code the bypass operand after you have performed the required action, or if you are performing the action after the apply, if that is appropriate.

2. **External hold data** — External hold data is not a part of the PTF. It resides in a separate file. On SAMS product tapes, there is a HOLDDATA file. External hold data is usually used for sysmods that have been distributed, and later are discovered to cause problems.

To take advantage of the external hold data, you must receive it into your SMP/E environment. If you use the jobs supplied by Computer Associates, SMP/E receives the hold data.

If a sysmod has an unresolved hold error, SMP/E does not install it unless you add a bypass to your apply command. You can bypass an error hold in situations that do not affect you. Error holds that do not apply can include a problem that only happens with a hardware device that you do not have, or in a product feature that you do not use.

When Computer Associates issues the sysmod that resolves the hold, the resolving sysmod supersedes the hold error. This allows you to apply the original sysmod in conjunction with the fixing sysmod.

There is a special hold data class called ERREL. This means that Computer Associates has determined that the problem fixed by the sysmod is more important than the one that it causes. Computer Associates recommends that you apply these sysmods.

The easiest and most reliable way to manage external hold data is to allow SMP/E to manage it automatically. When you allow SMP/E to manage the process, the only manual task you need to do is run a REPORT ERRSYSMODS. This report identifies any held sysmods that you have already applied to your system. If the resolving sysmod is in receive status, SMP/E identifies the sysmod that you need to apply to correct the situation.

---

## *Activating the CA-Disk System*

The instructions for activating the basic CA-Disk system are below. All steps are required; none are optional. The instructions include:

1. The CA-Disk Parameter Library (Parmlib) — Identify Files and Parmlib data sets to CA-Disk Functions
2. The CA-Disk JCL Procedures — Modify Symbolic Parameters in JCL Procedures and Customize JCL Procedures
3. The CA-Disk Files data set — Create a Files Definition Member in Parmlib and Initialize the Files data set

---

## *The CA-Disk Parameter Library (Parmlib)*

CA-Disk uses control parameters to provide many different kinds of information needed for processing. Some indicate how CA-Disk is to operate, some define the format of reports, some define data sets for which CA-Disk is to skip processing, and others provide user-dependent information to CA-Disk functions.

These parameter lists are specified as members of a partitioned data set known as the parameter library, or parmliib for short. The parmliib implementation allows CA-Disk to better tailor its operation to your requirements. The contents of a parmliib member can easily be changed. After they are changed, the next execution of CA-Disk uses them.

The parameter library is ready to use just as it was loaded from tape. The supplied members are regarded as parameter lists to be used for internal CA-Disk system functions and must not be changed; they are replaced with each new release. Parameter information needed by your installation and users can be placed in new members of the parameter library that you create.

The section “[PARMLIB](#)” on page 474 in the [Systems Guide](#), explains common CA-Disk parameter lists and the purpose of each. The sections for the other CA-Disk functions further defines the parmliib members that each uses.

For the remainder of the installation process, only two members of parmliib are of primary concern. These are members SYSPARMS and FILEDEFN.

Member FILEDEFN is used to define the internal attributes of the CA-Disk files data set (“catalog” or “index” data set). This is discussed in a later installation step.

Member SYSPARMS is used to indicate your selection of processing options to CA-Disk. The “[Sysparms](#)” on page 105 in the [Systems Guide](#) describes all of the system parameters (sysparms) that are available to modify the way CA-Disk executes. All have system defaults; do not try to review every one of them now. The remaining installation steps and the customizing instructions reference specific sysparms which familiarizes you with their use. As an

example, the default retention period for data sets being archived is 30 days. To change this to 120 days, create a member called SYSPARMS in the parmlib data set and add one line that contains ARCRETPD0120 and/or RETRETPD0120.

Because the CA-Disk parameter library is a partitioned data set, it can be updated like any other PDS. If you do not have an online editor, use the IBM utility IEBUPDTE for your updates.

## Step 1. Identify Files, Parmlib, and Activation Codes

Several functions of CA-Disk need to know the names of the files data set and parmlib data set to be used within the function. The names of these data sets are made available to CA-Disk through the SET command.

Additionally, indicators can be set to allow the use of preallocated files and parmlib data sets. Alternate files and parmlib data sets can then be used simply by allocating them prior to executing the desired function.

To provide the correct data set names and permit alternate data sets to be preallocated, use the following utility. A copy of this JCL can be found in member name DMSSET in the CA-Disk installation library.

```
//MODIFY EXEC PGM=ADSMI002, PARM=ADSDM338
//STEPLIB DD DISP=SHR, DSN=SAMS.DISK.LOADLIB
//SYSLIB DD DISP=SHR, DSN=SAMS.DISK.LOADLIB
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=A
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DISP=SHR, DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
SET .....
```

## SET Command Syntax

The SET command specifies the data set names and options to be used during some CA-Disk functions. An example format for the command and a description of its parameters follow. Only one command is accepted per execution:

```
SET FILESDSN=, PARMSDSN=, ALTFILES, ALTPARMS, PARMLIBD, INIT=, SVCNBR
```

**FILESDSN=** - The name of the CA-Disk files data set to use in the CA-Disk environment. This is a required parameter.

**PARMSDSN=** - The name of the CA-Disk parmlib data set to use in the CA-Disk environment. This is a required parameter.

**ALTFILES** - Specify this simple parameter to allow users to allocate DDNAME=FILES to their ISPF/TSO session prior to executing CA-Disk command processors. This allows users to use other than the default files data set.

**ALTPARMS** - Specify this simple parameter to allow users to allocate DDNAME=PARMLIB to their ISPF/TSO session prior to executing CA-Disk command processors. This allows users to use other than the default parmlib data set.

**PARMLIBD** - If your online environment already uses a ddname of PARMLIB (for something other than CA-Disk), specify this parameter to cause CA-Disk to use the alternate ddname of PARMLIBD.

**INIT=** - Specify this parameter to activate CA-Disk as indicated in your Installation Instructions letter.

**SVCNBR=** - This parameter can be used to indicate the CA-Disk SVC number, but it is no longer required for internal CA-Disk processing. This value is a 3-digit SVC number in decimal form. The default value is 244.

## The CA-Disk JCL Procedures

CA-Disk provides a set of PROCLIB JCL members for all functions in order to save time in the initial JCL preparation, reduce execution failures due to improper JCL, and greatly enhance the installation, testing and continued use of the CA-Disk functions. You may need to modify some of these procedures to increase the size of the temporary and sortwork files, depending on the size of your FILES data set. The procedure library contains the following:

**TABLE 1. Proclib JCL Members**

PROC	Function	PROC	Function
ARCHIVE	explicit archive	LISTD	list archive index
BILLING	non-VSAM DASD billing (select, accumulate)	LISTREQ	list deferred archive/restore requests
COMPRES	PDS compression	MERGE	merge unexpired archive data sets
DEFVOLS	list tapes needed for deferred restores	MERGCOPY	copy new archives created by merge
DERASE	delete deferred archive/restore requests	MIGRATE	sequential migration to tape
DIM	Dynamic Installation Manager	PDS2SEQ	merge PDS members into a sequential file
DMS	DSCL-invoked functions	RACFCHK1	RACF profile synchronization utility
DMSAR	Auto-Restore	REBUILD	archive index rebuild
DMSGTF	diagnostic procedure used for problem resolution	RECOVER	implicit recovery
DMSLDSD	RACF profile list utility	RELOAD	files data set recovery
DMSLRAC	list alternate RACF profile cross-reference	REORG	files data set reorganization
DMSPOOL	tapepool update utility	REPORT	special purpose non-VSAM reports
DMSPROF	CA-Disk RACF profile maintenance utility	RESTART	restart after REORG shutdown
DMSUTIL	miscellaneous diagnostics	RESTORE	explicit restore
EXTEND	DASD billing (extend totals and clear)	SMFRPT	data set report based on SMF data
FMS	DSCL Recover	THRSHMGR	volume threshold manager
FRECOVER	Files data set forward recovery	TSTAR	Test a new/updated Auto-Restore function
IXCATLG	catalog archived DSNs for Auto-Restore	UNLOAD	files data set backup
IXMAINT	index maintenance utilities	VSAMBILL	VSAM DASD billing (select, accumulate)
IXUPDATE	index update utilities	XCOPY	Disaster recovery extract utility

All of these procedures make consistent use of the following symbolic parameters and their recommended defaults.

**TABLE 2. Symbolic Parameter Names and Defaults**

SYMBOLIC	DEFAULT	EXPLANATION
P =	SAMS.DISK.INSTALL	library for control statements
Q =	SAMS.DISK	qualifier for CA-Disk data sets
S =	*	SYSOUT class default

**TABLE 2. Symbolic Parameter Names and Defaults**

SYMBOLIC	DEFAULT	EXPLANATION
W =	SYSDA	WORKFILE default unit name (must have a storage volume)
U =	null	TSO userid

The procedures generate the following data set names if the recommended P and Q symbolic parameters are used:

**TABLE 3. data set Names Generated by Procedures**

SYMBOLIC	DEFAULT
&Q..PARMLIB	SAMS.DISK.PARMLIB (contains the system parameters and installation options)
&Q..LOADLIB	SAMS.DISK.LOADLIB (contains the executable load modules)
&Q..FILES	SAMS.DISK.FILES (contains the index of archived or backed up data sets)
&Q..ARCHPRIM	SAMS.DISK.ARCHPRIM (data set name used for the primary archive/backup tapes)
&Q..ARCHCOPY	SAMS.DISK.ARCHCOPY (data set name used for the duplicate copy archive/backup tapes)
&Q..MERGPRIM	SAMS.DISK.MERGPRIM (data set name used for the primary tapes created by merge)
&Q..MERGCOPY	SAMS.DISK.MERGCOPY (data set names used for the duplicate copy tapes from merge)
&U.&Q..WORKFILE	SAMS.DISK.WORKFILE (data set names used for the PDS compression workfile)
&P	SAMS.DISK.INSTALL (data set names used for the library containing needed control statements)

The procedures also use the following control statements that by default are retrieved directly from the CA-Disk installation library. They can be copied to any source image library, but the symbolic parameter P must then be modified to name the appropriate library.

**TABLE 4. Procedures and Associated DD Statements**

DD STATEMENT	PROCEDURE(S) THAT USE THEM
DMSORT1	ARCHIVE, DMS, DMSLRAC, MERGE, MIGRATE, RECOVER, REPORT, RETAIN
DMSORT2	MERGE
DMSORT3	EXTEND (DASD Billing)
DMSORT4	COMPRES
DMSORT5	DMS
DMSORT6	sort for non-DSNINDEX subfile records
DMSORT7	sort for DSNINDEX subfile records only
DMSORT8	DMS
DMSPASS1	REPORT
DMUNLOAD	MERGE
DMUNLOD2	RACFCHK1 and to unload all subfiles
GTFPOPT	DMSGTF
RELOAD	REORG, RESTART and to reload subfiles
SMFSORT	SMFRPT

## Step 2. Modify Symbolic Parameters in JCL Procedures

Each JCL procedure can be updated manually using ISPF edit or any other editor you prefer. Alternatively, two utilities are provided in the Installation library that allow you to globally edit the JCL procedures:

1. **PROCUNLD** - Use this JCL utility against your procedure library to unload the procedures into a sequential file in IEBUPDTE format. Use ISPF edit or another editor to globally modify and change the procedures. For example: To change the first and second level qualifiers of your CA-Disk target libraries, you can use the following ISPF edit command:

```
CHANGE SAMS.DISK TO SYS2.SAMS.DISK ALL
```

2. **PROCRELD** - Use this JCL utility against your procedure library to reload the tailored procedures into your target procedure library.

## Step 3. Customize JCL Procedures

The PDS Compress function uses a workfile to perform the compression. It is allocated by the //COMPWORK dd statement in the distributed COMPRES procedure, and must be big enough to hold the largest PDS selected for compression. The supplied primary and secondary space allocation quantities are adequate for processing most PDSs, even when limiting the workfile to a single volume. However, to compress a very large PDS, the space available for the workfile on one volume may not be sufficient. If you view this as a potential problem, update the COMPRES proc directly and change the value of the WRKVOLS= parameter, which defaults to 1. A larger value permits the workfile to extend to multiple volumes as needed. Set it to the maximum number of volumes you want to make available to the workfile. Your installation's configuration limits the value you can assign to this parameter.

The distributed procedures for ARCHIVE, DMS, RECOVER, RESTORE and RETAIN all include dd statements for tape drives that have been commented out. Although these dd statements can be “uncommented” to cause system allocation routines to allocate the proper device when the job is initiated, CA-Disk default processing dynamically allocates them when and as needed. Dynamic allocation is recommended because the actual need for the device depends upon the CA-Disk parameters that are specified for the job. Simulated functions probably do not need the drives, but when submitted in live mode, they do. Similarly, the CA-Disk ARCHIVE and RESTORE facilities have options to queue the requests rather than executing them immediately. Queuing the requests does not require the drives, but later processing of the queues does.

“TAPE” is used for the tape unit name and is not included as a symbolic parameter. You can update it to the appropriate value for your installation.

The region size in most of the procedures has been set at 5120K. Monitoring the actual memory use during the evaluation period can permit further reductions, based on the specific processing at your installation.

## The CA-Disk Files data set

CA-Disk uses a single direct access data set, called the files data set, to record information related to the various functions you are executing. The information stored in the files data set is divided up into sections called subfiles. The following table lists each subfile and describes its function.

**TABLE 5. Files data set Subfiles**

Subfile	Description
DSNINDEX	an index of all data sets in the archives
ARCHVOLS	an index of tape or disk volumes that contain archived data sets
DMSPOOLS	volser names of tapes (assigned to pools) that can be used during archive or backup runs
ARCHCMDS	a queue for user requests to archive or backup specific data sets
RESTCMDS	a queue for user requests to restore specific data sets
RETEXCLD	re-archive grace periods for restored data sets
DASDSPCB	DASD space billing records
MIGRECAT	re-catalog information for sequential migration backup tapes
DMSPARMS	TSO dynamic (immediate) restore information
RACFEND	RACF profile name cross-reference

The next two steps are required for all new users of CA-Disk. If you already have a files data set from an 8.1 release or above, you can continue to use that files data set; however, you can create and initialize a new files data set for testing purposes before you install this release into your production environment. If you have a files data set from an 8.0D release or below, see [“Files Data Set Conversion” on page 118](#).

## Step 4. Create a Files Definition Member in Parmlib

To create the files data set, first define its subfile characteristics to CA-Disk. A sample set of file definition entries is supplied in the parameter library. The only variable in each entry is the number of records (capacity) you expect to need in each subfile. The minimum is one. For initial use, the supplied capacities should be more than sufficient, and also provide the best performance. Copy the sample definitions (member FDSAMPLE in parmli) into a new member of parmli named FILEDEFN. This new member is used to initialize your files data set.

## Step 5. Initialize the Files data set

After your file definition member FILEDEFN has been created as outlined above, run the following JCL to allocate and initialize the files data set. Your output consists of a status report based upon your definitions. You can find a sample in [Figure 1, “Sample Files Data Set STATUS Report,” on page 18 in the Systems Guide](#).

Special note for RACF users: By default, CA-Disk bypasses a data set if the VTOC entry for it indicates it is RACF-indicated. Details on instructing CA-Disk to process these data sets is presented in the topic [“Installing the RACF Security Interface” on page 30](#). Therefore, if the files data set becomes RACF-indicated as soon as you allocate it, you can receive message 0725 saying that the data set has been bypassed (and therefore not initialized). Either “unprotect” the files data set for the duration of the initialization run, or specify sysparm RACFSUPP with a value of Y, which allows the function to complete.

The following JCL can also be found as member FILEINIT in the installation library.

FIGURE 1. Sample FILEINIT Member in INSTALL Library

```

EDIT          .DMS.R900.INSTALL(FILEINIT) - 01.01          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOBNAME  JOB (ACCT INFO)
000200 //*****
000300 //* SAMPLE JCL TO INITIALIZE (FORMAT) THE SAMS:DISK FILES DATA SET *
000400 //*
000500 //* NOTE: CHECK DSNAMES, ADD VOLSER IF DESIRED *
000600 //*****
000700 //FILE1  EXEC PGM=ADSMI000, PARM=ADSDM100, REGION=2048K
000800 //STEPLIB DD DISP=SHR, DSN=SAMS.DISK.LOADLIB
000900 //ABNLDUMP DD DUMMY
001000 //CMDPRINT DD SYSOUT=*
001100 //FILES   DD DISP=(,CATLG,DELETE), DSN=SAMS.DISK.FILES,
001200 //          VOL=SER=, <=== SUPPLY VOLUME
001300 //          UNIT=SYSALLDA, DCB=(DSORG=DA),
001400 //          SPACE=(CYL,10,,CONTIG)
001500 //MSGPRINT DD SYSOUT=*
001600 //PARMLIB DD DISP=SHR, DSN=SAMS.DISK.PARMLIB
001700 //SYSPRINT DD SYSOUT=*
001800 //SYSUDUMP DD SYSOUT=*
***** ***** Bottom of Data *****
Aa A TCPIP R 4 C 15 17:00 4/01/98
  
```

## Step 6. Verify CA-Disk Enqueue Usage

CA-Disk issues several types of enqueues during processing. Some enqueues are the result of normal system services such as dynamic allocation. Others are issued for internal CA-Disk functions. Turn to [“ENQ/DEQ/Reserve Usage” on page 506 in the Systems Guide](#) for more information about the enqueues, dequeues, and reserves issued by CA-Disk. If you use CA-Disk in a multi-system environment, make sure that all enqueues are propagated across all systems. Failing to do so can result in problems later.

*Note:* When a reserve against a CA-Disk files data set is converted to an enqueue, both SCOPE=SYSTEMS and sysparm RSUPPRES set to a Y need to be in place for a GRS environment to function properly.

---

## Activating CA-Disk Features

Below you will find instructions for activating the following CA-Disk features:

- CA-Disk System Security Features and Interfaces
- VSAM Support
- ISPF Support
- TSO Support
- DASD Billing
- DBRC Interface
- Stand-Alone Restore

### Activating CA-Disk Security Features and Interfaces

#### Step 1. Select Password-Indicated data set Support Options

In IBM's standard password support, when a password-indicated non-VSAM data set is opened, the operating system prompts the TSO user (for online applications) or the master console operator (for batch jobs) for the password to the data set in question. Authority to continue is based on the response to that prompt. For non-VSAM data sets, the password indicator bit is the DS1IND10 bit set (bit x'10' at offset 93 x'5D') in the format-1 DSCB.

CA-Disk has special code to avoid prompting the operator for the password on OPEN and SCRATCH of non-VSAM user data sets. This special code is used by CA-Disk only when sysparm PASSWORD is specified with a value of Y, when the PASSWORD parameter is included on the command (if it has an optional PASSWORD parameter), and when CA-Disk is running as an APF-authorized task. Most TSO and ISPF sessions run non-APF-authorized, so they are not exempted by CA-Disk from authority checking.

The security package named SECURE replaces the non-VSAM password-checking feature of the operating system. SECURE does not result in prompting of the operator for open of data sets, so this special CA-Disk code is not needed.

There are two substeps for installing the password-indicated data set support options. The substeps are:

1. If your installation does not use the security package SECURE, review the setting of sysparm PASSWORD. If you want CA-Disk to process password-indicated data sets, specify the sysparm value as Y.  
If you do not want CA-Disk to process password-indicated data sets, it can default to N.  
If your installation uses the security package SECURE, specify sysparm SECUSUPP with a value of Y. With SECURE, the value of the sysparm PASSWORD does not matter.
2. Review the setting of sysparm PASSNEWN. If you want CA-Disk to allow the renaming of password-indicated data sets, specify the sysparm value as Y. If you do not want CA-Disk to allow the renaming of password-indicated data sets, let it default to N.

### Step 2. Activate CA-Disk System Security

Security administrators should review CA-Disk sysparms “SPFUSRIDn” on page 180 in the [Systems Guide](#) and “TSOUSRIDn” on page 185 in the [Systems Guide](#), which apply to the ISPF and TSO functions.

Refer to “[Security Processing](#)” on page 311 in the [Systems Guide](#). This documentation describes several security features along with the CA-Disk security philosophy. Your installation's security administrators should determine the features applicable to your environment. You can activate the features now, or you can add them at a later time.

### Step 3. Prepare Security Packages to Work With CA-Disk

If you are planning to run CA-Disk and a security package on your system without purchasing a CA-Disk Selectable Unit for the security package, follow the tasks outlined below. The tasks show the minor changes needed to let CA-Disk work properly with your security package.

### Step 4. Install CA-Disk Security Interface

If you have purchased one of the CA-Disk security package Selectable Units, or if you are planning to run CA-Disk under one of the security packages CA-ACF2, CA-Top Secret or RACF, follow one of the portions of this installation step as appropriate.

Review the setting of sysparm SECURVOL in your installation. Of special importance is its use in the volume-level functions of VBACKUP and VRECOVER. If it is left at its default value of Y, volume-level checking will be performed. This applies to all three CA-Disk security interfaces.

---

## Installing CA-Disk Under CA-ACF2

Regardless of whether or not the CA-Disk Security Interface for CA-ACF2 is installed, there are several items that must be done in order for CA-Disk to run effectively on a system using CA-ACF2.

Set up your production CA-Disk runs to have sufficient access to process those data sets managed by CA-Disk. The simplest way to accomplish this is to run these CA-Disk tasks with a Logonid that has the non-CNCL attribute.

Many users with ACF2 select instead to set up a Logonid as a data administrator:

1. Create a Logonid that has the MAINT attribute.
2. Create an ACF2 GSO options record to allow that Logonid access via program ADSMI002 from your SAMS.DISK.LOADLIB. This looks like:

```
MAINT  LIBRARY(SAMS.DISK.LOADLIB)
       LID(logonid)
       PGM(ADSMI002)
```

If you already have a MAINT record, you need to append a qualifier to the record name to generate a unique RECID. This looks like:

```
MAINT.DMS          LIBRARY(SAMS.DISK.LOADLIB)
                   LID(logonid)
                   PGM(ADSMI002)
```

With the exception of the TSO command processors, most of the time CA-Disk runs as the program name ADSMI002. The remaining usage is either with the program name of ADSMI000 or ADSMI302.

Do not use ADSMI302 unless Customer Support requests a PET trace for debugging a problem. The other program (ADSMI000) is used in situations where CA-Disk does not control the retention of the tape and therefore the tape created should not be under EDM control. This is the case in Sequential Migrate or Files data set Unload processing.

Because of this, both ADSMI000 and ADSMI302 also need an ACF2 GSO option.

3. Run these CA-Disk production jobs with that Logonid.

Other alternatives are also possible.

Your individual users can process (archive, restore, migrate, and so on) those data sets for which they are authorized.

Also, to accomplish its function, CA-Disk needs to be able to directly update format-1 DSCBs in volume VTOCs. For instance, backups must be able to turn off the “change bit” DS1IND02. Restore and Move/Copy processing must update certain fields maintained by CA-Disk. See [“The CA-Disk SVC” on page 52](#) for a list of those fields. To avoid S913-38 abends as CA-Disk tries to do this function, you must allow CA-Disk to open VTOCs for INPUT and OUTPUT.

If you plan to run any CA-Disk task with a Logonid that does not have the non-CNCL attribute, allow CA-Disk programs to update your VTOCs by adding the ACF2 access rule:

```
$KEY(SYSVTOC)
- UID(*) LIB('SAMS.DISK.LOADLIB') PGM(ADSMI002)
READ(A) WRITE(A)
```

In addition, if you plan to use the CA-Disk TSO command RESTORE, add the ACF2 access rule:

```
$KEY(SYSVTOC)
- UID(*) LIB('SAMS.DISK.LOADLIB') PGM(RESTORE)
READ(A) WRITE(A)
```

---

*Note:* READ(A) also implies EXEC(A). If the CA-Disk load library is in the Linklist, replace LIB('SAMS.DISK.LOADLIB') with LIB('SYS1.LINKLIB'). WRITE(A) can be replaced with WRITE(L).

---

With the exception of the TSO command processors, CA-Disk always runs as the program name ADSMI002.

If you plan to run all of your CA-Disk tasks with Logonids that have the non-CNCL attribute, you do not need to worry about ACF2 protection of VTOCs.

## Installing the CA-ACF2 Security Interface

1. Activate the CA-Disk CA-ACF2 Security Interface by specifying sysparm ACF2SUPP with a value of Y in the SYSPARMS member of the parmlib data set.
2. Review access to data set names VTOC.volser and DMSOS.Vvolser. If you use the “SELECT VTOCS” DSCL statement, CA-Disk backs up the VTOC of each volume processed, tracking this information by the esoteric data set name “VTOC.volser”, where “volser” is the volume on which the VTOC resides.

If you create volume-level backups with the VBACKUP command, CA-Disk backs up each volume, tracking this information by the esoteric data set name “DMSOS.Vvolser”, where “volser” is the volume being backed up.

BACKUP, VBACKUP, and IXMAINT functions each query any CA-Disk Security Interfaces for authority to process these names.

If you plan to use the “SELECT VTOCS” DSCL statement, or the VBACKUP command, make sure that your BACKUP, VBACKUP, and IXMAINT functions can each access this fictitious data set name.

3. Examine the following special considerations for implementation of CA-ACF2 in multiple-CPU shops.

If the ACF2SUPP sysparm is specified with a value of Y and CA-ACF2 is not active on the CPU on which CA-Disk is running, CA-Disk issues an appropriate message and abends. This prevents CA-Disk from processing data sets that are normally protected by the security system. This situation sometimes occurs in installations that have CA-ACF2 installed only on some of the CPUs that CA-Disk is running on.

To avoid this problem, specify sysparm ACF2FORC with a value of Y to indicate that CA-Disk continues processing even if CA-ACF2 is not installed on the system on which it is running. If ACF2FORC is specified as Y and CA-ACF2 is not active on the system, no security checking is performed.

---

## Installing CA-Disk Under CA-Top Secret

Regardless of whether or not the CA-Disk Security Interface for CA-Top Secret is installed, there are several items that must be done in order for CA-Disk to run effectively on a system using CA-Top Secret.

1. Set up your production CA-Disk runs to have sufficient access to process all data sets managed by CA-Disk. The simplest way to accomplish this is to run these CA-Disk tasks with a user ACID that has all authority to all data sets.

Other alternatives are also possible. Individual users can process (archive, restore, etc.) for only those data sets they are authorized to.

2. Decide if you must set sysparm ARSECURE. If you do not intend to set CA-Disk sysparm ARSECURE, proceed to step 4.

If you intend to set CA-Disk sysparm ARSECURE, you must set up CA-Disk as a CA-Top Secret Facility. Follow the steps below to do so.

- Choose an 8 character Facility Name, a unique Facility Code for the TSSUTIL report, and a 7 character user ACID name.
- Create the Facility. If you select to call the facility DMSOS and you are operating under CA-Top Secret 4.2 or below, you can use the following CA-Top Secret Control Options:

```
FAC ( USERx=NAME=DMSOS )
FAC ( DMSOS=PGM=ADS )
FAC ( DMSOS=ACITVE , NOABEND , NOASUBM , NOAUDIT , AUTHINIT )
FAC ( DMSOS=ID=facilitycode )
FAC ( DMSOS=NOINSTDATA , KEY=8 , LCFEED , LOCKTIME=0 )
FAC ( DMSOS=NOLUMSG , LOG ( VIOL ) )
FAC ( DMSOS=NOMRO , SUAS , NOPSEUDO , NORNDPW , RES , SIGN ( M ) )
FAC ( DMSOS=NOSHRPRF , NOSTMSG , NOTENV , NOTSOC , WARNPW )
FAC ( DMSOS=NOXDEF )
```

---

*Note:* If the LOG(VIOL) parameter in the Control Option FAC(DMSOS=NOLUMSG,LOG(VIOL)) is obsolete on your release of CA-Top Secret, remove it.

---

If you call the facility DMSOS and you are operating under CA-Top Secret 4.3 or above, you can use the following CA-Top Secret Control Options:

```
FAC ( USERx=NAME=DMSOS )
FAC ( DMSOS=PGM=ADS )
FAC ( DMSOS=ACITVE , NOABEND , NOASUBM , NOAUDIT , AUTHINIT )
FAC ( DMSOS=ID=facilitycode )
FAC ( DMSOS=NOINSTDATA , KEY=8 , LFCMD , LOCKTIME=0 )
FAC ( DMSOS=NOLUMSG , LOG ( MSG ) )
FAC ( DMSOS=SUAS , NORNDPW , RES , SIGN ( M ) )
FAC ( DMSOS=NOSHRPRF , NOSTMSG , NOTSOC , WARNPW )
FAC ( DMSOS=NOXDEF )
```

- Build an ACID for the CA-Disk Auto-Restore started task. If you call the ACID DMSAR, you can use the following TSO command:

```
TSS CREATE ( DMSAR ) NAME ( ' DMS AUTO-RESTORE ' ) -
FAC ( STC ) TYPE ( USER ) PASS ( NOPW ) -
DEPT ( deptname ) MASTFAC ( DMSOS )
```

- Connect the user ACID to the CA-Disk Facility in the CA-Top Secret STC record. If you call the ACID DMSAR, you can use the following TSO command:

```
TSS ADDTO ( STC ) PROC ( DMSAR ) ACID ( DMSAR )
```

- Connect each of your end-user ACIDs that might use Auto-Restore to the CA-Disk Facility in the CA-Top Secret STC record, or if you select to call the facility DMSOS, you can use the following TSO command:

```
TSS ADD ( DMSAR ) FAC ( DMSOS )
```

## Installing the CA-Top Secret Security Interface

1. Activate the CA-Disk CA-Top Secret Security Interface by specifying sysparm TOPSSUPP with a value of Y in the SYSPARMS member of the parmlib data set.
2. Review access to data set names VTOC.volser and DMSOS.Vvolser. If you use the "SELECT VTOCS" DSCL statement, CA-Disk will back up the VTOC of each volume processed, tracking this information by the esoteric data set name "VTOC.volser", where "volser" is the volume on which the VTOC resides.

If you create volume-level backups with the VBACKUP command, CA-Disk backs up each volume, tracking this information by the esoteric data set name "DMSOS.Vvolser", where "volser" is the volume being backed up.

BACKUP, VBACKUP, and IXMAINT functions each queries any CA-Disk Security Interfaces for authority to process these names.

If you plan to use the "SELECT VTOCS" DSCL statement, or the VBACKUP command, make sure that BACKUP, VBACKUP, and IXMAINT functions can each access this fictitious data set name.

3. Examine the following special consideration for implementation of CA-Top Secret.

Under most versions of IBM's operating systems, OPEN, SCRATCH and RENAME processing will query CA-Top Secret for authorization, regardless of the setting of the RACF-indicator bit. This feature is called "always call". data sets cataloged in ICF catalogs also cause a query of CA-Top Secret for authorization, regardless of the setting of the RACF-indicator bit.

Under some operating systems, data sets not cataloged in ICF catalogs query CA-Top Secret only if the RACF-indicator bit is on. For non-VSAM data sets, the RACF-indicator bit is the DS1IND40 bit (bit x'40' at offset 93 x'5D') located in the data set's format-1 DSCB.

CA-Disk security processing normally queries CA-Top Secret for authorization, regardless of the setting of the RACF-indicator bit. If you do not have the “always call” feature of the operating system and you do not use ICF catalogs, specify sysparm TOPSALWZ with a value of N in the SYSPARMS member of the parmlib data set.

---

## *Installing CA-Disk Under RACF*

Regardless of whether or not you intend to install the CA-Disk Security Interface for RACF, there are several items that you must care for in order for CA-Disk to run effectively on a system using RACF.

You will need to set up your production CA-Disk runs to have sufficient access to process all data sets managed by CA-Disk. The simplest way to accomplish this is to run these CA-Disk tasks with a user ID that has the OPERATIONS attribute. Other alternatives are also possible.

Your individual users can process (archive, restore, migrate, and so on) those data sets for which they are authorized.

At this time, RACF does not provide the ability to give different DASDVOL authorities based on the program name being run. With the exception of the TSO command processors, CA-Disk always runs as the program name ADSMI002.

If you plan to run the DSCB Update function with a user ID that has the OPERATIONS attribute, or do not plan to run the utility, you do not need to worry about RACF DASDVOL rules.

## **Installing the RACF Security Interface**

There are nine steps for installing the CA-Disk RACF Security Interface. Steps 1 through 4 are required only for users who have discrete RACF profiles at their shop. Due to RACF's continuing support of discrete profiles, we recommend that all users follow each step.

### 1. Storing CA-Disk-Saved Discrete Profiles

Prepare for CA-Disk-saved discrete profiles to be kept in IBM's RACF data set. This step is optional but review it for possible applicability.

CA-Disk-saved discrete profiles are created and maintained through standard RACF macros (see [“Security Processing” on page 311 in the Systems Guide](#)). RACF places the new profiles in the RACF data set selected by the user. While most users elect to keep all data set profiles in a single RACF data set, it can be useful (for example, to improve RACF performance by reducing contention between CA-Disk RACF and other RACF requests) to separate CA-Disk profiles from standard RACF data set profiles. This can be done by using the CA-Disk-saved discrete profile data set name prefix as an identifier to CA-Disk profiles (see the next step). Then specify the CA-Disk prefix in the RACF RANGE TABLE (ICHRRNG) to indicate to RACF where to place CA-Disk profiles.

### 2. Specifying Name for CA-Disk-Saved Discrete Profiles

Provide a prefix (first qualifier) for the data set name of CA-Disk saved discrete profiles by specifying sysparm RACFUSID with a 1- to 8-byte name. This prefix identifies the CA-Disk profiles and allow them to be placed on a RACF data set apart from the standard RACF data set. It is a RACF restriction that this RACFUSID value represent a user ID or group ID. We recommend using a user ID, not a group ID, for the value for this sysparm. It is meaningful to you in identifying CA-Disk profiles. We recommend using “DMSOS”.

To avoid having RACF RACDEF processing update the PERMIT list, ensure that the user ID does not have the GRPACC attribute.

This RACF user ID is able to restore any data set for which CA-Disk has saved a discrete profile. To prevent this exposure of unauthorized use of this RACF user ID, it can be revoked using the TSO command:

```
ALTUSER racfusid REVOKE
```

Revoking the RACF user ID in this manner does not prevent its use for CA-Disk discrete profile support.

### 3. Specifying Volume for CA-Disk-Saved Discrete Profiles

Provide a volume name to be associated with CA-Disk-saved discrete profiles by specifying sysparm RACFDVOL with a character volume serial number. Due to RACF restrictions, this volume must be a real DASD volume. Once specified, it must not change. Therefore, select a volume that is always online.

### 4. Review RACF Utility Function

Review the description of the utility documented under the heading [“Management of CA-Disk-Saved Profiles” on page 354 in the Systems Guide](#). You do not need to use the utility at CA-Disk installation time, but be aware of the utility's existence. After the CA-Disk RACF Security Interface has been implemented, the utility can be run periodically, prior to running the CA-Disk IXMAINT function. See [“IXMAINT Utility” on page 320 in the User's Guide](#).

### 5. Set Sysparm RACFSUPP and RACFPROC

Activate the CA-Disk RACF Security Interface by specifying sysparm RACFSUPP and RACFPROC with a value of Y in the SYSPARMS member of the parmlib data set.

### 6. Review Access to Special CA-Disk data set Names

Review access to data set names VTOC.volser and DMSOS.Vvolser. If you use the “SELECT VTOCS” DSCL statement, CA-Disk backs up the VTOC of each volume processed, tracking this information by the esoteric data set name “VTOC.volser”, where “volser” is the volume on which the VTOC resides.

If you create volume-level backups with the VBACKUP command, CA-Disk backs up each volume, tracking this information by the esoteric data set name “DMSOS.Vvolser”, where “volser” is the volume being backed up.

BACKUP, VBACKUP, and IXMAINT functions each queries any CA-Disk Security Interfaces for authority to process these names.

If you plan to use the “SELECT VTOCS” DSCL statement, or the VBACKUP command, make sure that BACKUP, VBACKUP, and IXMAINT functions can each access this fictitious data set name.

### 7. Special RACF Considerations

Examine the following special consideration for implementation.

Under most versions of IBM's operating systems, OPEN, SCRATCH and RENAME processing query RACF for authorization regardless of the setting of the RACF-indicator bit. This feature is called “always call”. data sets cataloged in ICF catalogs also cause a query of RACF for authorization, regardless of the setting of the RACF-indicator bit.

Under some operating systems, data sets not cataloged in ICF catalogs will query RACF only if the RACF-indicator bit is on. For non-VSAM data sets, the RACF-indicator bit is the DS1IND40 bit (bit x'40' at offset 93 x'5D') located in the data set's format-1 DSCB.

CA-Disk security processing normally queries RACF for authorization, regardless of the setting of the RACF-indicator bit. If you do not have the “always call” feature of the operating system and you do not use ICF catalogs, specify sysparm RACFALWZ with a value of N in the SYSPARMS member of the parmlib data set.

### 8. Reviewing Applicable Sysparms

Review the following sysparm descriptions for possible use in your installation. They are located beginning on [page 160](#) of the *Systems Guide*.

RACFALLO	RACFMODL	RACFPRED
RACFBKUP	RACFMVOL	RACFSEQM
RACFDVL2	RACFNEWN	RACFVCAV
RACFMDSN	RACFPDSW	

## Activating Miscellaneous Security Features

Certain CA-Disk parmlib members contain information that you do not want your users to override. The SYSPARMS, SPFOPTNS, and DMCOPTNS members are good examples. This section describes how to restrict CA-Disk to a list of authorized parmlib data sets. You can activate this protection now or at a later date.

### Parmlib Security — PARMAUTH

If you want to restrict your users to a list of authorized CA-Disk parmlibs, activate the Parmlib Security Feature by installing user exit "USERMOD5" as follows:

1. Locate the source for the Parmlib Security Feature in member PARMAUTH, located in the library associated with the DMSASM DDDEF. Below is a sample of that source:

FIGURE 2. Sample source for PARMAUTH

```

File Edit Transfer Options Connection Macro Window Help
-----
Menu Utilities Compilers Help
-----
BROWSE .DMSASM(PARMAUTH) - 01.00 Line 00000000 Col 001 080
Command ==> Scroll ==> PAGE
***** Top of Data *****
PARMAUTH TITLE 'SAMS:Disk System parameter Data set security'
*****
          COMPILE ASEM=RENT,LKED=RENT
*
* DESCRIPTION:
* This is a sample usermod used to tailor SYSPARM data set
* specification security.
*
* Read the PARMAUTH-macro prolog for specifications of options
* you may override.
*
*****
PARMAUTH PARMAUTH SECURITY=NO, X
          SECURLIB=SYS1.PARMLIB, X
          SECURTL=ZDMSPARM
          END
***** Bottom of Data *****
Aa A TCPIP R 4 C 15 10:53 11/24/97

```

2. In order to ensure that the changes you make to PARMAUTH are protected during future CA-Disk installs or maintenance, copy this member into the source library associated with the //USERASM dd statement in USERMOD5.

3. Customize PARMAUTH as follows:
  - Activate the Parmlib Security Feature by specifying “YES” to the SECURITY= parameter. The default is “NO”, which deactivates the feature.
  - Alter the SECURLIB= parameter as required. The value you specify is the name of the data set that stores the list of authorized parmlibs. The default is “SYS1.PARMLIB”. However, this can be any highly protected library available to CA-Disk, TSO, and ISPF users with an LRECL of 80.
  - Alter the SECURTLB= parameter as required. The value you specify is the name of the member that stores the actual list of authorized parmlibs. The default is “ZDMSPARMS”.
  - Save your work by issuing SAVE at the TSO command line.
4. Create member to store your authorized parmlibs. The data set must match that specified for “SECURLIB=”, and the member must match that specified for “SECURTLB=”. If you allowed SECURLIB= to remain at its default value, you must create the member “ZDMSPARM in your cataloged SYS1.PARMLIB data set.

To create a list of authorized parmlibs, use the sample SAMPZDMS in PARMLIB.

To prevent a security exposure (and in keeping with IBM's strategic directions), only cataloged CA-Disk parmlibs can be authorized. If a user creates an uncataloged parmlib (or a data set with the same name but on another pack), then tries to use the parmlib using the VOL=SER= parameter, CA-Disk notes that the parmlib is uncataloged or incorrectly cataloged, issues a descriptive message and abends.

---

*Note:* If you are installing a new release and have created a test parmlib, you must include this test parmlib in the list. If you want to keep your production users from using your test parmlib, instruct your security package to give access to the test parmlib only to yourself.

---

## USERMOD5

USERMOD5 must be installed using SMP/E. Sample JCL is provided for you in member USERMOD5 of the install library. Install USERMOD5 as follows:

1. Locate the source for USERMOD5 in the install library. Below is a sample of that source:

FIGURE 3. Sample SMP/E JCL for USERMOD5

```

File Edit Transfer Options Connection Macro Window Help
-----
EDIT                                     .INSTALL(USERMOD5) - 01.00      Columns 00001 00072
Command ==>                               Scroll ==> CSR
***** Top of Data *****
000001 //JOBNAME  JOB (ACCT INFO)
000002 //SMP      EXEC PGM=GIMSMP,REGION=5120K,
000003 //          PARM='CSI=SAMSSMPE.GLOBAL.CSI'
000004 //SYSPRINT DD SYSOUT=*
000005 //SMPRPT  DD SYSOUT=*
000006 //SMPOUT  DD SYSOUT=*
000007 //SMPHOLD DD DUMMY
000008 //USERASM  DD DISP=SHR,DSN=USERID.DISK.ASM
000009 //SMPCNTL DD *
000010 SET BOUNDARY(GLOBAL).
000011 RECEIVE S(SDU1815) .
000012 SET BOUNDARY(DMSTGT) .          /* <--- SAMS:DISK TARGET */
000013 APPLY  S(SDU1815).
000014 //SMPPTFIN DD *
000015 ++USERMOD(SDU1815).
000016 ++UER(2038) FMID(SDM1900) .
000017 ++SRC(PARMAUTH) DISTLIB(ADMSASM) DISTMOD(ADMSLOAD) TXLIB(USERASM) .
000018 ++SRC(SYSPAETH) DISTLIB(ADMSASM) DISTMOD(ADMSLOAD) TXLIB(USERASM) .
000019 /*
Aa  A TCPIP          R 4  C 15          11:08  11/24/97

```

2. Customize USERMOD5 by supplying the appropriate information for the following:
  - jobcard information
  - Global and Target CSI names
  - USERASM data set name
3. Submit USERMOD5.

After a successful APPLY run, CA-Disk attempts to read your SECURLIB= data set. If you allowed SECURLIB= to remain at its default value and your security package protects SYS1.PARMLIB, you must perform one additional step to make the list of authorized parmlibs available to CA-Disk. To accomplish this, do the following:

- Grant READ access to each of your CA-Disk users through all programs from your CA-Disk load library.

---

*Note:* If you have users running CA-Disk TSO or ISPF functions under a TSO session that is not APF-authorized, these functions must obtain a shared enqueue to read SYS1.PARMLIB. If the shared enqueue can be obtained quickly, this should not be a problem as the read is very brief. However, if a shared enqueue cannot be obtained, the job goes into a wait state until the enqueue is available. CA-Disk tasks running APF-authorized do not use an enqueue on SYS1.PARMLIB, and are not affected.

---

## System Parameter Override Security — SYSPAUTH

CA-Disk sysparm overrides are controlled through use of the //SYSPARMS dd statement. Use of this dd statement is described in [“Overriding Sysparms Instream” on page 106 in the Systems Guide](#). Users can override sysparms only if SYSPARMO is specified with a value of Y in the SYSPARMS member of your CA-Disk parmlib.

If you do not have a security package that is compatible with SAF, you cannot limit access to sysparm overrides for a subset of users, or for a subset of sysparms.

If your security package is SAF-compatible, and if you want to restrict your users to a subset of CA-Disk sysparms that they can override, or restrict sysparm overrides to a certain group of users, you can activate the System Parameter Override Security Feature by installing user exit "USERMOD5" as follows:

1. Locate the source for the System Parameter Override Security Feature in member SYSPAETH, located in the library associated with the DMSASM DDDEF. Below is a sample of that source:

FIGURE 4. Sample source for SYSPAETH

```

File Edit Transfer Options Connection Macro Window Help
-----
Menu Utilities Compilers Help
-----
BROWSE
Command ==> .DMSASM(SYSPAETH) - 01.00 Line 00000000 Col 001 080
Scroll ==> PAGE
***** Top of Data *****
SYSPAETH TITLE 'SAMS:Disk System parameter override security'
*****
      COMPILE ASEM=RENT,LKED=RENT
*
* DESCRIPTION:
* This is a sample usermod used to tailor SYSPARM override
* SAF security checking.
*
* Read the SYSPAETH-macro prolog for specifications of options
* you may override.
*
*****
SYSPAETH SYSPAETH SECURITY=NO,          SYSPARM-overide security (YES/NO) X
          RESOURCE=DISK.SYSPARMS,      Prefix of protected res.   X
          CLASS=FACILITY,              Resource class             X
          APPL=DMSOS                   Application of record
      END
***** Bottom of Data *****
Aa A TCP/IP R 4 C 15 11:16 11/24/97
    
```

2. In order to ensure that the changes you make to SYSPAETH are protected during future CA-Disk installs or maintenance, copy this member into the source library associated with the //USERASM dd statement in USERMOD5.
3. Customize SYSPAETH as follows:
  - Activate the System Parameter Override Security Feature by specifying "YES" to the SECURITY= parameter. The default is "NO", which deactivates the feature.
  - Alter the RESOURCE= parameter as required. The value you specify is the prefix of the resource. The default is "DISK.SYSPARMS".
  - Alter the CLASS= parameter as required. We recommend you specify FACILITY for ACF2 and RACF security packages, or IBMFAC for TOP-SECRET.
  - The value specified for the APPL= parameter is recorded by SMF. The default value is DMSOS.
4. Save your work by issuing SAVE at the TSO command line.

5. Activate the FACILITY (IBMFAC if you have CA-TOP SECRET) class of your security package. For example, if you have RACF, issue the command:

```
SETROPTS CLASSACT(FACILITY) GENERIC(FACILITY) +  
GENCMD(FACILITY)
```

When SYSPARMO is specified with a Y, CA-Disk uses the FACILITY (IBMFAC if you have CA-TOP SECRET) class of your security package to determine if the user has READ access to the resource “DISK.SYSPARMS.sysparmname” before allowing the override.

---

*Note:* The check for READ access is with the FACILITY (IBMFAC if you have CA-TOP SECRET) class, not the data set class. Therefore the name does not refer to the name of an actual data set—there could be other rules on the data set class that refer to data sets with that same name. RACF users must execute this command before creating the following generic profile.

---

6. Protect all sysparm overrides as a blanket. For example, if you have RACF, issue the command:

```
RDEFINE FACILITY DISK.SYSPARMS.* +  
UACC(NONE) NOTIFY(security administrator)
```

where “security administrator” is the userid of a person to whom optional violation messages are to be sent.

If you have CA-TOP SECRET, issue the command:

```
TSS ADDTO(deptacid) IBMFAC(DISK.SYSPARMS.*)
```

7. If you want to allow your storage management group of users to override most or all of the system parameters, you can instruct your security package to allow that. For example, if you have RACF, issue the command:

```
PERMIT DISK.SYSPARMS.* CLASS(FACILITY) +  
ACCESS(READ) ID(storagemanagementgroup)
```

where “storagemanagementgroup” is the group name (or list of userids) of your storage administrators.

If you have CA-TOP SECRET, you can issue the command:

```
TSS PERMIT(acid) IBMFAC(DISK.SYSPARMS.*) ACC(READ)
```

where “acid” is the acid to whom you want to be able to use run-time sysparm overrides.

8. Then if you want to restrict your storage management group from overriding certain system parameters, you can individually protect them. For example, if you have RACF, issue the commands:

```
RDEFINE FACILITY DISK.SYSPARMS.sysparmname +  
UACC(NONE) NOTIFY(securityadministrator)  
PERMIT DISK.SYSPARMS.sysparmname CLASS(FACILITY) +  
ACCESS(NONE) ID(storagemanagementgroup)
```

If you have CA-TOP SECRET, issue the commands:

```
TSS ADDTO(deptacid) IBMFAC(DISK.SYSPARMS.sysparmname)  
TSS PERMIT(acid) IBMFAC(DISK.SYSPARMS.sysparmname) +  
ACC(READ)
```

where “acid” is the acid to whom you want to be able to use run-time sysparm overrides.

9. Finally, if you want to allow all users to override certain system parameters, such as ARCONAME and ARCCNAME, you can individually allow that. For example, if you have RACF, issue the commands:

```
RDEFINE FACILITY DISK.SYSPARMS.sysparmname +  
UACC(READ) NOTIFY(securityadministrator)
```

If you have CA-TOP SECRET, issue the command:

```
TSS ADDTO(deptacid) IBMFAC(DISK.SYSPARMS.sysparmname)
```

10. With this security feature activated, CA-Disk allows the override of a sysparm only if CA-Disk receives a return code of less than 8 from SAF using the macro:

```
RACROUTE REQUEST=AUTH,
  RELATED= 'DMS/OS AUTH CHECK' ,
  APPL='DMSOS ' ,
  ATTR=READ,
  CLASS='FACILITY' ,
  ENTITY=DISK.SYSPARMS.sysparmname
```

where “sysparmname” is the 8-character name of the system parameter. CA-Disk issues a message and ignores overrides of those sysparms to which the user does not have sufficient authority.

---

*Note:* CA-TOP SECRET translates this FACILITY class check to IBMFAC.

---

11. Submit USERMOD5 according to the instructions under [“USERMOD5” on page 35](#).

## Other Usermods

All other usermods can be installed using the installation instructions for “USERMOD5” on page 35. The following table is a list of such usermods:

**TABLE 6. List of Usermods**

USERMOD	SOURCE	Brief Description
USERMOD6	MTFLOCK	Used to tailor MTF function/command/parameter security. SAF is used for security checking. Read the MTFLOCK-macro prologue for specifications of options.
USERMOD6	PETLOCK	Used to tailor PET function/command/ parameter security. SAF is used for security checking. Read the PETLOCK-macro prologue for specifications of options.
USERMOD9	TSOEXMPT	TSO exclusion/exemption table
USERMOD9	DSNEXMPT	DSN exclusion/exemption table
USERMOD9	JOBEXMPT	JOB exclusion/exemption table
USERMOD9	PGMEXMPT	PGM exclusion/exemption table
USERMODB	ADSUMODB	When a data set is cataloged to the CA-Disk pseudo volume, the catalog management hook (ADSAR010) is designed to return the SVC26 requestor the volser of “MIGRAT”. If the program name or pattern is listed in this table, the CA-Disk pseudo volser will be returned instead. Read the SCRNLIST-macro prologue for specifications of options you can override.
USERMODC	ADSUMODC	Used to specify CA-Disk Auto-Restore DB2 recall delay time. Read the ADSUMODC-macro prologue for specifications of options you can override.
USERMODD	TASKLIB	Used to modify the CA-Disk DDNAME for programs running in the TSO/ISPF environment. Read the TASKLIB-macro prologue for specifications of options.
USERMODE	LINEMAX	Used to specify the page line-count, limit for the CA-Disk message processor. Read the LINEMAX-macro prologue for specifications of options you can override.
USERMODE	SYSOUTEX	Used to specify an exit routine for the CA-Disk message processor. Read the SYSOUTEX-macro prologue for specifications of options you can override.
USERMODF	ADSVS939	Used to activate CA-Disk catalog master password security. Read the CATLGPSW-macro prologue for specifications of options you can override.
USERMODG	ADSUMODG	Used to tailor MTF message diagnostics for MTF002I. Read the MTF002BY macro prologue for specifications of options you can override. In summary, use MTFMSG=YES to always issue the message; MTFMSG=NO to bypass the message unless MTF DD is active.
USERMODK	ADSUMODK	Used to specify whether CA-Disk should wait for RECON data set in case it is being used. Read the ADSUMODK module prologue for definition of values that can be specified.

## Activating MTFDEBUG DD Statement Security Feature

Using the //MTFDEBUG dd statement is not documented anywhere else in CA-Disk documentation, and is intended for use by CA-Disk technical staff only. The use of this dd statement can nevertheless be considered a security exposure.

CA-Disk allows the use of the //MTFDEBUG dd statement only if it receives a return code of less than 8 from SAF

using the macro:

```
RACROUTE REQUEST=AUTH,
  RELATED='DMS/OS AUTH CHECK',
  APPL='DMSOS ',
  ATTR=ALTER,
  CLASS='data set',
  ENTITY=DISK.USING.MTFDEBUG.DD
```

There is no CA-Disk control over this check. If you do nothing, the return code is always less than 8, and use of the //MTFDEBUG dd statement is allowed. If you do not have a security package that is compatible with SAF, you cannot control access to this dd statement.

To activate the CA-Disk //MTFDEBUG dd statement security feature, follow these steps:

1. Through your security package, grant ALTER access to the data set name "DISK.USING.MTFDEBUG.DD". A data set of this name does not need to exist; the name need only be significant to your security package.
2. If your SAF-compatible security package has protection-by-default and you want to allow certain users to use //MTFDEBUG dd statements, instruct your security package to grant these users ALTER access to "DISK.USING.MTFDEBUG.DD".

If your SAF-compatible security package does not have protection-by-default and you want to allow only certain users to use //MTFDEBUG dd statements, instruct your security package to deny access to "DISK.USING.MTFDEBUG.DD", (universal access of NONE) and then grant ALTER access to the authorized users.

CA-Disk issues a message and ignores //MTFDEBUG dd statements from users without sufficient authority to that resource.

---

## Activating VSAM Support

Follow these steps to activate the VSAM support.

1. Specify sysparm VSAMSUPP with a value of Y to activate the VSAM support.
2. A CA-Disk master password is available to allow processing of any password-protected cluster. For security reasons, it is supplied and documented in a separate enclosure. Review that documentation for use in your installation.
3. Review sysparm VSONLINE if your installation uses ICF catalogs and you have VSAM data sets defined on offline DASD volumes. For related information, Refer to the sysparm description for "[VSDSPACE](#)" on [page 195 in the Systems Guide](#).
4. IDCAMS uses the IDCUT1 and IDCUT2 dd statements in the RESTORE, RECOVER and DMSAR procedures to dynamically allocate work space when building a VSAM alternate index. Use one of two approaches to control where this space is allocated:
  - Modify the data set names to direct the allocation to a desired VSAM or ICF catalog, and modify the VOL=SER to point to volume(s) where work space can be allocated.
  - Supply the necessary sysparms to cause CA-Disk to dynamically allocate the needed dd statements prior to invoking BLDINDEX processing.

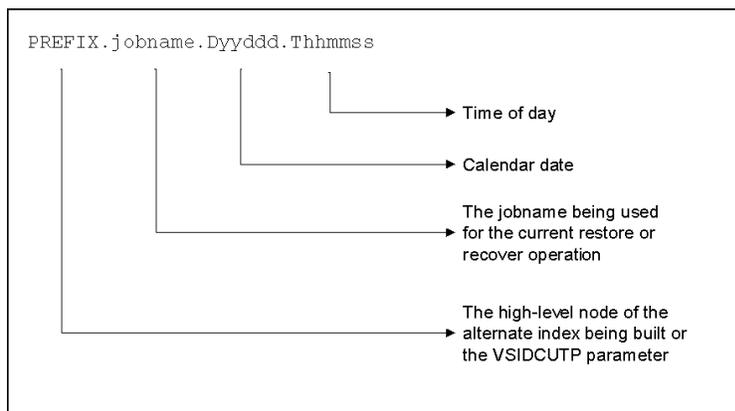
If you select the first method (supplying the JCL statements directly), review sysparms VSDEFCAT, VSBIXPSW and VSBIXCAT for consideration. The main drawback to this approach is that it forces all restore and recover jobs that use the same data set names for the work files to be single-threaded (that is, each job must wait until the previous restore has completed and released the exclusive enqueues on the work file data set names). This can cause major processing delays and frustration if there are many restore jobs waiting to be serviced. Also, the enqueues are left outstanding even if no VSAM data sets need to be restored.

If you select the second method (to leave the IDCUTx dd statements out of the JCL), simultaneous restore jobs can take place (as long as they don't require the same archive tape). When CA-Disk restores a VSAM alternate index and determines that BLDINDEX processing needs to be invoked, it checks to see if the IDCUTx dd statements are allocated. If they are not, it generates a unique name for each of the dd statements and allocates them accordingly. Two sysparms govern the allocation of these dd statements.

VSIDCUTPxxxxxxx—is used to specify the high-level node CA-Disk is to use for the work file name. For instance, if VSIDCUTPLABS is specified, the generated work file name begins with LABS. If a value is not specified, the high-level node of the first alternate index being built is used. This sysparm is most useful in installations that have not converted to ICF and therefore needs to have the work data set placed on specific volume(s) owned by the VSAM catalog pointed to by the high-level node of the data set name.

VSIDCUTVxxxxx...yyyyyy—is used to specify the volume(s) to be used for the work data set. Up to ten volumes can be specified in the list. If you want to use more than one volume, make sure that each of the volumes is specified as exactly six characters (for example, 'VSIDCUTVWRK800WRK802WRK806'). The names that CA-Disk generates have the form:

**FIGURE 5. VSIDCUTP Generated DSNAME**



5. If your installation has vendor products--or internally developed application programs--that generate non-standard VSAM entry-sequenced data sets (ESDS) or linear data sets (LDS), review sysparms VSACCESS and VSARCFMT. Products known to have these non-standard formats are IBM's DB2 database management system and certain products from MSA. These data sets must be archived in a control interval image copy format rather than in logical record format.
6. If your installation is using ICF VSAM support and you want to make use of the last use date support, you must make certain updates to your system to activate the date stamping mechanism (if your installation has not already done so). Date stamping can be requested only for VSAM clusters that are defined in ICF catalogs. See [“The CA-Disk VSAM Date Stamp” on page 57](#).

---

## Activating ISPF Support

The CA-Disk ISPF support provides an interface from the IBM System Productivity Facility, Dialog Management Services, to the CA-Disk product. CA-Disk uses ISPF dialog management service panels to interface to the user in an interactive environment.

After it is implemented, the CA-Disk dialog management panels can be used to perform both foreground and background CA-Disk functions. Through a simple modification to the ISPF primary option panel, the ISPF user can select CA-Disk as a function in the same way other functions are selected. Once the CA-Disk selection menu is entered, the user has access to almost all functions of CA-Disk.

### Dynamic Menu-Formatting Feature

To protect against unauthorized use of a function, a dynamic menu-formatting feature is provided. This feature allows the CA-Disk installation coordinator to specify what functions of CA-Disk each user is authorized to use. When the CA-Disk selection menu is entered, users are presented only the menu items for the functions for which they are authorized. Other functions are not accessible. Each user can be given a customized list of authorized functions. These options are all specified by entering control statements in a member of parmlib. See [“Defining ISPF and DSCL User Options” on page 89](#).

The CA-Disk dialog management functions either guide the user through the process of generating JCL to execute batch CA-Disk facilities, or directly interface to the CA-Disk foreground applications. For foreground processing functions, no JCL is involved, as all requests are performed immediately in the TSO user's region.

The CA-Disk ISPF function FRESTOR interfaces to the TSO RESTORE command processor. Specifying this option, as well as all others to be given to end users, is documented later in this section.

All CA-Disk dialog management panels have associated HELP text panels. If at any time the user needs additional information on how to proceed or what to do, the HELP PF key can be pressed. The HELP text can be browsed until the user is ready to continue. The key takes them back to the panel they were originally processing.

### Formatting Used for On-line Reporting

Several dictionaries in the CA-Disk parmlib govern the formatting used for online reporting. One of the options is the capability of including SMS constructs and attributes in the ISPF online reports. Users who want to change these dictionaries should have a working knowledge of the ISPF online reporting feature of CA-Disk. Be familiar with the section [“ISPF” on page 417 in the User's Guide](#), covering the online report facility, and also have had hands-on experience with the product. You must also define a report definition library to store your user-defined reports. For details, refer to [“ISPF Online Reports” on page 447 in the User's Guide](#).

To run the ISPF reports interactively, we recommend that the TSO region size be at least 3000K. Without the foreground report generation, a region size of 2000K should suffice for the report definition process.

Although the online reporting facility of CA-Disk provides a lot of flexibility in defining user reports, it does not allow much tailoring as far as how individual fields are printed (column headings, numeric editing patterns, etc.). You can tailor these values on an installation-wide basis, however, by modifying the data dictionary in parmlib.

Actually two different dictionaries are used for each report type. There is a low-level dictionary that maps field names to data items in the F1 and F4 DSCBs. These dictionaries are called FMT1FLDS and FMT4FLDS and should not be modified by the user. They only provide a mapping mechanism between the high-level dictionary and

the low-level data and have no control over report formatting options. The dictionaries that do control formatting of data fields are called FMT1DICT and FMT4DICT. The FMT1DICT member maps the fields used on the F1-DSCB reports and the FMT4DICT member is used for the volume summary reports (F4 DSCBs).

The ISPF online reports can include SMS constructs and attributes. If you have SMS managed data sets and choose to include these new fields in a report, you must rename the following members within the CA-Disk parmilib data set:

- Rename member FMT1FLDS to FMTOFLDS
- Rename member FMT1DICT to FMTODICT
- Rename member FMTSFLDS to FMT1FLDS
- Rename member FMTSDICT to FMT1DICT

---

*Note:* If these renames were done while operating under a previous release of CA-Disk and there were reports generated from the SMS constructs, you must issue these renames again prior to generating any reports.

---

Be careful when changing any fields that modify the width required to print a field if report definitions have been generated that use these fields. The column positioning values are calculated at report definition time and a change in the length can cause the output report to be incorrect (overlapped fields, etc.). If this occurs, the problem can be corrected by going through the report modification panels for each report using the affected field(s). No updates need to be made -- the field adjustments are automatically made as a by-product of the update process.

For detailed information about each of the fields in the dictionary, refer to [“ISPF Custom Reports” on page 85 in the Systems Guide](#).

The ISPF support is shipped to the user on the CA-Disk distribution tape as four separate files. A description the file contents follows:

**TABLE 7. ISPF Libraries and Descriptions**

<b>Library</b>	<b>Description</b>
ISPF Dialog Manager Panel	Contains the panel definitions for all CA-Disk dialog manager panels and contains all HELP facility panels relating to the function and menu panels.
ISPF Dialog Manager Skeleton JCL	Contains skeleton JCL used for building JCL that is submitted for processing from CA-Disk ISPF applications.
ISPF Dialog Manager Message	Contains error messages to be printed by the dialog manager when the CA-Disk application detects an abnormal condition.
ISPF Dialog Manager Load Modules	Contains all CA-Disk load modules. It is used by the dialog manager to obtain all CA-Disk load modules.

## Connecting the ISPF Libraries to Your System

Allocate the CA-Disk dialog manager libraries to your TSO session using the following steps. This must be done before invoking ISPF.

1. Determine how your installation allocates the data sets referred to by the following ddnames to the TSO session prior to invoking the ISPF facility. They can be allocated by including them in the TSO LOGON JCL procedure or through a CLIST that is executed sometime before ISPF is invoked.

**TABLE 8. ISPF Clist DD Names**

DDNAME	DDNAME LIBRARY
ISPPLIB	PANEL
ISPSLIB	SKELETON
ISPMLIB	MESSAGE
ISPLLIB	LOADLIB

2. Update the allocation of these libraries to concatenate the CA-Disk ISPF libraries after them. The names of these libraries correspond to the ddnames to which they should be concatenated.
3. Check the blocksizes of the libraries being concatenated. The blocksize of the first library must be greater than or equal to the second, or errors occur. This can be corrected by either reblocking the libraries or simply putting DCB=BLKSIZE= a larger value on the dd statement for the first data set of the concatenation.

For example, JCL before libraries are concatenated:

```
//ISPPLIB DD DISP=SHR,DSN=ISP.vrm.ISPPLIB
//ISPSLIB DD DISP=SHR,DSN=ISP.vrm.ISPSLIB
//ISPMLIB DD DISP=SHR,DSN=ISP.vrm.ISPMLIB
//ISPLLIB DD DISP=SHR,DSN=ISP.vrm.ISPLLIB
```

JCL after CA-Disk libraries are concatenated:

```
//ISPPLIB DD DISP=SHR,DSN=ISP.vrm.ISPPLIB
// DD DISP=SHR,DSN=SAMS.DISK.ISPPLIB
//ISPSLIB DD DISP=SHR,DSN=ISP.vrm.ISPSLIB
// DD DISP=SHR,DSN=SAMS.DISK.ISPSLIB
//ISPMLIB DD DISP=SHR,DSN=ISP.vrm.ISPMLIB
// DD DISP=SHR,DSN=SAMS.DISK.ISPMLIB
//ISPLLIB DD DISP=SHR,DSN=ISP.vrm.ISPLLIB
// DD DISP=SHR,DSN=SAMS.DISK.LOADLIB
```

## Customizing the CA-Disk ISPF Support

Use one of the two methods below for installing the ISPF interface:

**Method #1:** This method has the advantage of reducing EXCPs associated with CA-Disk ISPF and thereby reducing execution time. It has the disadvantage of increasing the number of concatenated libraries to ISPLLIB and thereby increasing the number of directory blocks searched for non-CA-Disk load modules. This disadvantage can be minimized by placing the CA-Disk load library last in the ISPLLIB concatenation. To use this method, perform the following steps:

1. Alter all TSO/ISPF logon clists by placing the CA-Disk load library last in the ISPLLIB concatenation.
2. Add the following line to ISR@PRIM:

```
8, 'PGM(ADSMI002) PARM(ADSSP044) NEWAPPL(ISR)'
```

**Method #2:** This method has the advantage of decreasing the number of concatenated libraries to ISPLLIB and thereby decreasing the number of directory blocks searched for ISPF functions. It has the disadvantage of increasing the number of EXCPs associated with CA-Disk ISPF functions. This increase is incurred as CA-Disk modules are loaded. This load process requires a directory search for each individual module, which is not required in Method #1. To use this method, perform the following steps:

1. Alter all TSO/ISPF logon clists by adding an allocation for the CA-Disk load library to DMSOSLIB file.
2. Add the following line to ISR@PRIM:

```
8, 'PGM(ADSSP202) PARM(ADSSP044) NEWAPPL(ISR)'
```

---

*Note:* For override options, See [“TSO Command and ISPF/PDF Task Library Override - TASKLIB” on page 95.](#)

---

---

## *Activating TSO Support*

Follow these instructions to activate the support.

### **Step 1. Make the Command Processor Available to TSO**

Ensure that the dialog manager programs are available to the TSO session. This process has been partially completed, since you have already installed the TSO Command Processors, and control program ADSSP202 into one of the system linklist libraries.

To complete the process, modify your ISPF/TSO logon procedure to allocate a ddname of DMSOSLIB that points to the CA-Disk load library; that is:

```
//DMSOSLIB DD DISP=SHR,DSN=SAMS.DISK.LOADLIB
```

Also, add to your ISPF/TSO logon procedure a ddname of PARMLIB that points to your CA-Disk parameter library:

```
//PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
```

This step is required to give the TSO Command Processors access to other CA-Disk programs. The CA-Disk control module ADSMI002 always looks for the DMSOSLIB library first, and find all of the needed CA-Disk modules there without searching any other libraries. Your other ISPF functions never waste time searching the CA-Disk directory looking for one of their programs.

TSO command processors are available to perform the functions as listed below.

**TABLE 9. TSO Command Processor Available Functions**

Module	Function
DARCHIVE	Queue a request to archive a data set
DRESTORE	Queue a request to restore a data set
DERASE	Erase a queued archive or restore request
LISTDMS	List the CA-Disk index of archived data sets
LISTREQ	List the status of queued archive/restore requests
RESTORE	Restore a data set immediately (uses dynamic tape allocation). This command processor must also be designated as privileged.

*Note:* For override options, See [“TSO Command and ISPF/PDF Task Library Override - TASKLIB” on page 95.](#)

## Step 2. TSO Help Text

The Help text members were placed into the CA-Disk TSO Help library during the installation. Ensure that this library is allocated to your TSO session.

## Step 3. The TSO Dynamic Restore Command Processor

If you want to permit the use of the dynamic RESTORE command, follow the instructions below to activate this feature. For more information on the dynamic RESTORE command, refer to [“RESTORE — Immediate \(Dynamic\) Restore” on page 485 in the User’s Guide.](#)

The TSO dynamic RESTORE command processor uses a subfile called DMSPARMS, defined within the files data set. This subfile was described in the FDSAMPLE member in parmlib, and was created when you defined and initialized the files data set.

The dynamic restore command processor must be designated as privileged or authorized in the MVS environment, due to its use of the ENQ and ALLOCATE SVCs. Update your SYS1.PARMLIB(IKJTSOxx) member to include “RESTORE”. After the changed member has been activated by an IPL, any TSO user can do dynamic restores. TSO mount authority is required if the data set being restored is archived to tape.

*Note:* Standard ISPF does not allow authorized TSO commands to be entered from its menu option 6. Unless local ISPF modifications have been made to permit such executions, you must exit ISPF before using this command.

## Activating DASD Billing

Follow these instructions to activate the support.

1. A RESTART data set is required for DASD billing. It is used to clear the billing file and in proper recovery after an abend. It is referenced by the //RESTART dd statement in the EXTEND procedure, and defaults to DSN=&Q..RESTART.
2. Create the RESTART data set, and allocate it with the following attributes:

```
//RESTART DD DCB=(DSORG=PS,RECFM=F,LRECL=96,BLKSIZE=96),  
//          SPACE=(CYL,(2,2))
```

---

## Activating the DBRC Interface

Using the DBRC interface is optional. You can skip this step unless you use DBRC (Database Recovery Control) and are running MVS with IMS Release 4.1 or above. Even if you are, this step is still optional.

The performance, ease-of-use, data compression and other attributes of CA-Disk provide an alternative to using the IBM image-copy program for backing up IMS databases. The CA-Disk backups can be written to disk and/or tape and managed with the flexibility standard to all CA-Disk backups. If you still want to use IBM image-copy, benefits can be obtained by putting the image copies on disk, and then using the CA-Disk Sequential Migration to Tape function to free the disk space. The degree of operator intervention, primarily for tape handling, can be reduced significantly.

The purpose of the DBRC interface in CA-Disk is to provide an automated means of notifying DBRC whenever any action is taken by CA-Disk against a DBRC-registered data set. This includes the direct processing of database data sets themselves, such as in backup functions, or in moving IBM image-copy data sets that have been placed on disk.

If you activate the DBRC interface, CA-Disk will extract the database names and their associated data sets from the DBRC RECON data set and keep the list in memory. When CA-Disk is taking a backup (or archive) copy of one of these data sets, a control statement for the IBM DBRC update utility (program DSPURX00) is created to notify DBRC that a "user image-copy" has been taken. Similarly, the restore (or recover) functions generate control statements to notify DBRC that the recovery has been performed.

If you take IBM image-copy backups and place them on disk instead of tape, the Move/Copy and Sequential Migration to Tape functions can encounter them (intentionally or otherwise) and move them to new locations. Instead of extracting the database data set names as described above, these functions extract the image-copy records from the DBRC RECON data set. When an image-copy data set is moved, a control statement to change the volume list in the DBRC image-copy record is produced.

The DBRC utility program itself can be executed as either the last step of the job that produces the control statements, or at some later time.

Since databases are often managed separately, it is desirable to supply the information as outlined below, but only for the specific jobs designed to need it. Instead of making changes directly to the CA-Disk JCL procedures and activating the support for all jobs, this information can be provided only for the specific jobs intended; for example, the database backup and recovery jobs.

The RECON data sets are dynamically allocated based upon the names provided in the IMS resident load library. CA-Disk determines the current RECON data set and uses it. Alternatively, you can provide them in the JCL through //RECON1, 2 and 3 dd statements.

The generated control statements and any messages relating to the DBRC processing are printed in a separate output listing for easier review. //DBRCPRT is dynamically allocated for this purpose, but can be supplied in the JCL if desired. If it appears that the control statements are not being generated, supply the //RECONDMP in the JCL to produce a listing of the data sets from the RECON data set. Match the listing against the data sets processed by CA-Disk. If the data set appears in both the listing and the CA-Disk job, verify member DBRCDSNS of parmlib.

The control statements produced are of the following general format. For fast path databases with replication, messages are issued to indicate that CA-Disk has processed an area data set. However, no control statements are produced. If the data set is given a new name when the backup copy is taken, the new name appears in the UDATA field.

For backup:

```
NOTIFY.UIC DBD(DBAHSP00) DDN(DBAHSP01) -
RUNTIME(862991635000) UDATA('*** DMS ***')
```

For recovery:

```
NOTIFY.RECOV DBD(DBAHSP00) DDN(DBAHSP01) -
RCVTIME(862991635000) CURRENT
```

For migration of image-copy data sets:

```
CHANGE.IC DBD(DBAHSP00) DDN(DBAHSI01) -
RECTIME(862671605243) VOLLIST(WRK800)
```

## To Activate the DBRC Support

1. Provide CA-Disk with the name of the IMS resident load library in sysparm DBRCRLIB. CA-Disk gets the names of the RECON data sets from here. The library is dynamically allocated when it is needed.

Alternatively, provide a //RESLIB dd statement to identify the library. The ARCHIVE, RETAIN, RESTORE, RECOVER, DMS, CONFIGR and MIGRATE procedures potentially needs the library.

2. Provide CA-Disk with the release number of IMS you are using by specifying sysparm DBRCRLSE. The default value is for Release 1.3.0. Enter the value without periods: 130.

The format of the RECON data sets changed in IMS Release 2.0.0., so for CA-Disk to determine the format, the release is needed.

3. Create member DBRCDSNS in the CA-Disk parmlib data set.
4. Enter the following in member DBRCDSNS: explicit and/or pattern names of IMS data sets and/or image-copy data sets for which CA-Disk DBRC processing is to be done.

A sample is provided in member SAMPDBRC of parmlib. If the name of the data set being processed does not have a matching entry in this member, no further DBRC processing takes place.

5. Ensure that the naming conventions in member DBRCDSNS provide for easy identification. When they do, this feature provides a more efficient method of determining that DBRC processing is not needed.

A pattern of "/" causes every data set to be examined against the list of names extracted from the RECON data set, while an empty member effectively turns off the DBRC support. Notice that by removing a pattern name or by supplying a pattern that never matches any data set names, the DBRC support is inactivated as well.

6. Provide a disk data set in which DBRC control statements can be written and identify this data set to CA-Disk with a //DBRCCARD dd statement in those functions where it is needed. The data set can be any sequential or partitioned data set that supports 80-byte fixed-length records.
7. Provide JCL for the DBRC update utility with the //SYSIN dd statement pointing to the same data set as in step 6 above.
8. Activate the support by specifying sysparm DBRCSUPP with a value of C.

---

The CA-Disk product provides the DASD administrator with a great deal of flexibility. In the pages that follow, several of the more commonly used customization options are discussed.

Whenever a CA-Disk customization option directly changes the operating system, it is highly advisable to apply the change as a usermod using SMP/E. At the same time, any change you make to CA-Disk itself should be considered as a usermod. You should also make these changes using SMP/E.

Each customization step discussed includes the proper JCL and instructions to install with SMP/E. After you have installed CA-Disk with SMP/E, make sure that you have run the ACCEPT step prior to installing any customizing options. This allows for the easy removal of these options, if problems should occur.

---

*Note:* Several SMP/E format usermods have sample JCL which should NOT be modified. This sample JCL is always preceded by a ++JCLIN control card. This sample JCL is used by SMP/E to indirectly determine the libraries to be updated. The sample JCL does not directly identify these libraries. Modification of this sample JCL can cause unpredictable results during installation of the usermod.

---

---

*Note:* Always verify the target and distribution zones into which you are installing each usermod. Usermods which modify your operating system--for example, the CA-Disk SVC, VSAM date stamp and so on--should be installed into the same zones as your operating system. Usermods which modify CA-Disk, such as auto restore zaps, etc., should be installed into the same zones as CA-Disk. Improper selection of target and distribution zones during the installation of user modifications can cause unpredictable results during installation.

---

As part of your planning for installing one or more customizing options, verify that the JCL provided conforms to your installation's JCL procedures. The Assembler JCL provided uses step names of "C" and "L". Your Assembler procedure can use "ASM" and "LKED". SMP/E JCL uses the DDNAME "SMPCNTL".

The following customizing options are discussed in this section:

- The CA-Disk SVC and applicable zap.
- The CA-Disk VSAM date stamp module.
- The auto-restore function.
- The CA-Disk tape management facility.
- Setting ISPF customization options.
- Defining ISPF user options.
- Defining TSO customization options.

---

## *The CA-Disk SVC*

The purpose of the CA-Disk SVC is to update a data set's VTOC entry, recording the last used date, for example. The CA-Disk SVC has the following advantage over the IBM SU 60 update code: exemption entries are provided such that the CA-Disk management tasks themselves don't cause data sets to appear used.

Zaps to one or more IBM modules must be applied to call the CA-Disk SVC to do the updating.

The CA-Disk SVC maintains the SU 60 change bit and last used date fields. It also maintains some extra CA-Disk-defined fields and provides the facilities to overcome the SU 60 deficiencies as described below.

The two fields maintained by IBM's SU 60 appear to provide the information that is needed to help manage disk storage, and CA-Disk functions can examine these fields and execute without the CA-Disk SVC being installed. However, from a practical point of view, the SVC is required for your implementation for the following reasons:

- The CA-Disk storage management tasks themselves open user data sets to perform the management functions; for example, backup, migration, compression, etc. IBM's SU 60 open hook by itself will mark these data sets as being used whenever these management functions are run, and therefore prevent other functions, such as archive, from finding and processing data sets that are truly inactive. That is, a potentially large number of data sets appear used and are kept on disk merely due to storage management jobs routinely being run. The CA-Disk SVC exempts the management jobs from causing these updates.
- The CA-Disk SVC can also record the date whenever the change bit is turned on (mod date). This can be useful in report information and, as explained in [“BACKUP and ARCHIVE Considerations” on page 195 in the User's Guide](#), it also provides a convenient and low-overhead means of truly eliminating redundant backup copies of data sets (that full pack dumps or other more simplistic techniques often create).
- The CA-Disk data set utilization (DSU) report loses much of its usefulness since it reports on the additional fields maintained by the CA-Disk SVC.

SVC Requirements - If the CA-Disk SVC is to be installed, a user SVC of type 3 or 4 must be provided by the installation. It should be enabled for interrupts and can be designated as either restricted or non-restricted. See "Item 3: User SVC" on page 11.

SVC Integrity and Security - Before any fields are modified in the DSCB, the SVC verifies that all INPUT parameters contain a F1-DSCB. It also checks to ensure that the caller is in key 0 and in supervisor state.

VTOC Fields Maintained by the CA-Disk SVC - If IBM modules are zapped to execute the CA-Disk SVC, the following fields are maintained in the F1-DSCB for each data set.

**TABLE 10. VTOC Fields Maintained by CA-Disk SVC**

Field #	DESCRIPTION	FORMAT	BYTES	OFFSET (DEC)
1	Last use date	ydd	3	75 SU 60
2	Last modify date	ydd	3	70
3	Job name or account code	char	8	62
4	Count of updates to VTOC entry	bin	2	73
5	Change bit	bit	x'02'	93 SU 60
6	SVC mode	char	1	103

Field 3 above is determined by macro settings within the CA-Disk SVC. You can choose to maintain an accounting code (from the first accounting field on the job card for the job or user accessing the data set) or job name. You can also select how this field is maintained--the first time the is updated, each time it is modified, or each time it is used. How you set the macro is discussed later.

## CA-Disk SVC Modes

The CA-Disk SVC is distributed with a control switch called SVCMODE that affects which fields are maintained and how often.

### SVCMODE=4

This mode of operation makes the minimum number of updates to the VTOC entry (F1-DSCB) for a data set, yet maintains all of the fields. The SVC source is distributed in this mode. It updates the F1-DSCB only if:

1. the last used date needs to be changed
2. the change bit needs to be set on (change bit turned on when data set opened for output), and MODDT is also updated

*Note:* (1) causes only one update per day, and (2) causes only one per day unless the change bit is being turned off more often than that (for example, running incremental backup two or more times per day).

Field 4 shows the number of times the F1-DSCB has been updated, not necessarily the number of times a data set has been opened.

### SVCMODE=5

This mode of operation causes updates under exactly the same conditions as mode 4, but only the SU 60 fields (1 and 5) plus the mode field itself (6) are maintained. This is an SU 60 “look-alike” mode, with the benefit of the exemption tables.

### SVCMODE=6

This mode of operation causes the F1-DSCB to be updated every time the data set is opened. Field 4 shows the number of times the data set has been opened.

This technique causes many more updates of the VTOC entry, and is known to cause occasional problems with partitioned data sets, made evident by duplicate TTR directory entries and loss (overlay) of a member update. This usually occurs only for PDSs that are accessed very heavily and by two or more concurrent users. Shared DASD environments without a “cross-system enqueue package” are especially susceptible.

## Tailoring the CA-Disk SVC

Source code for the CA-Disk SVC is supplied in member ADSMVS60 of the DMSASM library. The SVC has four tables that can be used to exclude from DSCB updating any combination of data sets, DASD volumes, jobnames and programs. The program exclusion capability is recommended for use when running CA-Disk jobs. This technique prevents a CA-Disk management job from causing a data set to appear to have been used. The SVC source code has each table initialized with one or more dummy entries to illustrate how entries are defined.

- The count halfword preceding each table must be incremented or decremented as necessary to reflect the deletion or addition of entries.
- The binary-length byte that immediately precedes each table entry must not exceed 44, 6, 8 or 8 for the data set, volume, jobname or program tables respectively.
- Generic entries can be specified to exclude a class of entities having a common prefix of less than the limit length for the particular entity type.

The program exclusion table is initialized to bypass DSCB updating for any program starting with “ADSMI”, as the CA-Disk control module does.

---

*Note:* The activating zap is such that the CA-Disk SVC “replaces” the MVS SU 60 update code (that is, if the SVC is activated, SU 60 decisions are deactivated).

---

## Installing the CA-Disk SVC

1. Based on the previous discussion, determine which SVC mode of operation you want to use. If you select mode 4, the default mode, skip the remainder of this substep and proceed to substep 2. If you select mode 5 or mode 6, you must change the supplied source code for the variable SVCMODE from C'4' to C'5' or C'6', corresponding to the mode you selected.
2. Decide which job name or accounting code will be maintained by the CA-Disk SVC. Then update the source code to reflect your selection. This update has been simplified so that it requires you to update only a macro. The default is to keep the creating JOBNAME. If this is appropriate, skip to substep 3.

This macro, \$JOBNM, has two parameters: “FLD” and “TYPE”. A “FIND” on the characters “\$JOBNM” positions you in the right place within the source.

The parameter “FLD” defines the field to maintain. Valid values are “ACCT” or “JOB”. To maintain a job name, specify this parameter with a value of “JOB”.

The parameter “TYPE” defines when this field is updated. Valid values are “CREATE”, “MOD”, or “USE”. To have the field updated each time the data set is accessed, set this parameter to a value of “USE”.

An example of the use of this macro is shown below:

```
$JOBNM FLD=ACCT,TYPE=CREATE
```

With the macro coded in this way, the CA-Disk SVC maintains the first eight bytes of the creating job's accounting for each data set.

- Next, determine which SVC number to use for the CA-Disk SVC. The CA-Disk SVC is normally installed as number 244, a number that it used by some IBM system products as well as other packages. Check if SVC 244 is already assigned in your system. If so, select a different number for the CA-Disk SVC.

If SVC 244 can be used by CA-Disk, proceed to substep 4.

Ensure that the SVC has been defined as a user SVC type 3 or 4, enabled for interrupts, and either restricted or non-restricted.

If you select an SVC number other than 244 for the CA-Disk SVC, you must modify the open zap (described in substep 10 on [page 57](#)) for this release. You need to make the same change to any CA-Disk open zaps that you can apply at any time in the future.

If you select an SVC number other than 244 for the CA-Disk SVC, specify its value in sysparm SVCNODMS. The SVC number in the open zap must correspond to the SVC routine installed, and the number logged in sysparm SVCNODMS.

- Determine if any of the fields maintained by the CA-Disk SVC will cause a problem at your site. This might be due to other products that maintain fields in the F1-DSCB. If any of the fields will cause a problem, modify the CA-Disk SVC to either move the field or not maintain the field.

To move a field, take the equate label from the table below and modify the SVC source to equate the label to the new value (minus 44) and set the corresponding CA-Disk sysparm from the table below to the same value (not minus 44).

To not maintain a field, take the equate label from the table below and modify the SVC source to comment out any code that uses the equate to examine or change the DSCB, then set the corresponding CA-Disk sysparm from the table below to a value of 000 in your parmlib data set.

**TABLE 11. SVC Equate Labels and Explanations**

SVC Equate Label	Sysparm Name	Default Value	Explanation
DMSUSEDT	DSCBLUSD	75	last opened date (SU 60)
DMSDSIND	n/a	93	change bit x'02' (SU 60)
DMSMODDT	DSCBLMOD	70	last modified date
DMSSVCMO	DSCBSVMD	103	CA-Disk SVC mode
DMSJOBNM	DSCBJBNM	62	job name or accounting code
DMSOPCNT	DSCBOPCD	73	open count (two bytes)

The value of each SVC equate label (minus 44) must be equal to the value of the corresponding CA-Disk sysparm.

- Update the SVC source to exclude from DSCB updating the desired set of data sets, DASD volumes, job names and program names. Be sure to review the default entries.

6. A diagnostic facility exists within the SVC which can display applicable data used to determine if DSCB updates are to be made to maintain fields within the DSCB. These diagnostic messages are automatically generated if the job name is equal to the job name identified in field "DIAGNAME". The default name is "DMSTEST". To change the job name used to generate diagnostic messages, update the "DIAGNAME" field. Similar diagnostic facilities are provided for the VSAM date stamp module and auto-restore catalog management hook. If you change the job name for the SVC, consider changing the job names used for the other diagnostic facilities. This is discussed where applicable for the other functions later in this section.
7. If you have installed a version of the CA-Disk SVC from a release prior to Release 8.1, you can have compatibility problems when using this CA-Disk SVC. See ["CA-Disk SVC" on page 121](#) for a discussion of these problems and how to correct them.
8. The CA-Disk SVC is now installed into the CA-Disk LPA List library. It does not need to be copied to SYS1.LPALIB. You only need to add the CA-Disk LPA List library to your system's LPA list.

A copy of the CA-Disk SVC with default values is placed in the CA-Disk LPA List library at product installation time. If the default values are appropriate for your installation, you do not need to reassemble the SVC.

To change any of the default values, make your changes to the source found in member ADSMVS60 of the DMSASM library. Then run job USERMOD1 in the installation library to reassemble and link the SVC into the CA-Disk LPA List library.

If you are running MVS 2.2.0 or higher, you do not have to change the name of the CA-Disk SVC to IGC00nnn in order to install it. However, install the SVC by adding the following line to the IEASVCXX member of your SYS1.PARMLIB:

```
SVCPPARM 244,REPLACE,TYPE(4),APF(NO),EPNAME(ADSMVS60)
```

9. To activate the SVC, it must be called as part of the processing for each disk data set. This is done by applying a zap to the proper IBM module for your operating system. The following table contains the IBM module names to be zapped depending on IBM product level. Apply 1 or more of the following:

**TABLE 12. List of ZAPS for IBM Maintenance Levels**

IBM Module	CSECT	DFP/DFSMS Level	MODEL	USERMOD
IFG0194A	IFG0194E	All levels	ZUZMODEL	USERMOD2
IGC0001I	IFG0196W	All except DFP 2.4 and below	ZOZMODEL	USERMODH
IGGDADSM	IGGDAU01	DFSMS 1.2 w/UW28103 DFSMS 1.3 w/UW28104 DFSMS 1.4 and above	ZZZMODEL	USERMODI
IGC00020	IFG0201R	DFSMS 1.2 w/UW28103 DFSMS 1.3 w/UW28104 DFSMS 1.4 and above	ZYZMODEL	USERMODJ

There are many levels of the IBM modules, and each requires a slightly different form of the zap. Zaps to match these maintenance levels are provided in the installation library. Use the one in which all the VERs match your version of the IBM modules. To accomplish this, locate your Operating System level in Table 12 and do the following:

- Get a zap dump of the IBM module
- Use SMP to determine the RMID of this same module

---

*Note:* Try to match the RMID to a corresponding member in the CA-Disk installation library named ZUYnnnnn or ZUWnnnnn. If you cannot find a supplied zap that matches your Operating System level, use a model zap and adjust it to fit. In the MODEL column of Table 12 corresponding to your system level is the name of a member in the installation library. If you need assistance in making the adjustments, contact the CA-Disk Technical Support Center.

---

- Make sure that all of the VERs in the zap match your dump of the module.
10. If the CA-Disk SVC number you are using is not the default of 244 (x'F4'), you must now adjust the zaps to be sure they execute the correct SVC; for example, for SVC 245, change 0AF4 to 0AF5 within the body of the zap itself.
  11. In the USERMOD column of Table 12 corresponding to your system level is the name of a member in the installation library. You can use this member as sample JCL to SMP/E install the IBM zap. This JCL installs a usermod onto your operating system.
  12. IPL the system to load the modified IBM modules and the CA-Disk SVC into storage. For MVS, do an MLPA or CLPA to initialize the link pack area.

---

## The CA-Disk VSAM Date Stamp

Last use date support for VSAM is maintained by IBM module IDATMSTP in their ICF VSAM support. CA-Disk provides a sample date stamp routine for you to use as member IDATMSTP in the DMSASM library. The advantage of using the CA-Disk version of IDATMSTP is that it provides exemption entries so that the CA-Disk management tasks themselves do not cause VSAM data sets to appear used, similar to the support in the CA-Disk SVC for non-VSAM data sets.

---

*Note:* The change bit (DS1IND02 bit in the Format-1 DSCB) is also updated by the VSAM Date Stamp module.

---

Modifications can be necessary to the source for the CA-Disk VSAM data stamp module to initialize exclusion tables and to optionally change the diagnostic job name.

To provide the exemption lists support, modify the IDATMSTP module in the same manner as described for the CA-Disk SVC. Note that the CA-Disk SVC and IDATMSTP exemption lists can contain different entries.

A diagnostic facility exists within the VSAM date stamp module which can display applicable data used to determine if the last use date is to be maintained. These diagnostic messages are generated automatically if the job name is equal to the job name identified in field "DIAGNAME". The default name is "DMSTEST". To change the job name used to generate these messages, update the "DIAGNAME" field in module IDATMSTP. Similar diagnostic facilities are provided for the IBM SVC module and auto-restore catalog management hook. If you change the job name for the VSAM date stamp module, consider changing the job names used for those functions as well.

After you have modified module IDATMSTP, you are ready to install it.

## Installing the IDATMSTP Module

Use member USERMOD3 in the CA-Disk installation library to install the usermod. SMP/E assembles and properly links the module into the correct library, regardless of your operating system level.

## Testing the Installation of the IDATMSTP Module

Now test the IDATMSTP module to see if it has been installed successfully. Submit a job that updates an ICF VSAM data set, using the jobname that you specified within the IDATMSTP module. Diagnostic messages are printed to show that IDATMSTP has been entered and what action, if any, is taken. A return code is also printed when IDATMSTP is exited. A return code of zero indicates that the last use date is not updated. A return code of 4 indicates that the last use date is updated.

---

## *The Auto-Restore Function*

One of the more important reasons an installation purchases any data storage management system is to minimize the amount of DASD space allocated to data sets that are rarely or never accessed. This can be accomplished in CA-Disk by both explicit archival and implicit archival based on last used date. Typically, the more constrained an installation is for DASD space, the shorter the time allowed for a data set to be unused prior to archival. As the unused time window shrinks, more and more data sets are archived that do get accessed infrequently. This is especially common for data sets that get referenced only on a monthly or yearly basis.

Although this archival scheme accomplishes one of the DASD manager's primary functions--ensuring enough DASD free space exists for day-to-day operations--it can also create a less than amicable relationship with the data center's users if not implemented properly. For example, if a system is set up to archive all data sets not used within two weeks, those data sets that are only accessed once a month are archived between each job cycle.

The user must either respond to this problem by restoring all required data sets prior to a job run, or artificially referencing the data sets periodically to avoid having them archived. In the first case, an extra workload is created for the user, and in the second case the DASD manager is prevented from doing an effective job. Although the DASD manager could exempt these data sets from being archived by CA-Disk, this cannot be a good alternative since the data sets could then remain on DASD long after the system that uses the data sets is removed from production. And if an installation is critically short of free DASD space, it cannot have the luxury of letting seldom-used data sets tie up valuable DASD space between job cycles.

So how does an automatic restore capability help to solve this potential conflict between the DASD manager and the end user? By allowing data sets to be archived by the DASD manager and then restored automatically by CA-Disk if they are required by an application at a later time. This solves the DASD manager's problem of keeping free space available, and also relieves the user of the burden of restoring required data sets prior to cyclical job runs.

## The Two Auto—Restore Methods

CA-Disk has provided two methods to perform auto-restores. One method is the S213 Abend Exit (F1-DSCB Not Found) and the other method is a Catalog Management hook.

New users of CA-Disk are encouraged to install the Catalog Management hook and NOT install the S213 Abend Exit. If you have installed the S213 Exit from previous releases, you are encouraged to remove it. Documentation for installing and removing the S213 Abend Exit is found in [“S213 Abend Exit” on page 126](#).

### Catalog Management Hook (IGG026DU Module):

The IGG026DU module is provided by IBM in catalog management to initiate auto-restores. This module is not really an exit, but it hooks into the front end and back end of catalog management processing. Since many catalog management requests do not require auto-restores, this module must decide dynamically which requests to intercept and which to pass on.

### DFHSM SVC Hook (IGX00024 Module):

The IGX00024 module is installed in conjunction with the Catalog Management Hook and comprises the second method. It is used primarily to intercept DFHSM auto recall requests that should be processed by CA-Disk.

Several program products, DB2 for example, issue catalog locates and look at the returned volser to anticipate the need for a data set to be restored. The program recognizes the special volser MIGRAT as the volser DFHSM uses when it archives a data set. If this information shows that the data set is archived, the program tries to directly invoke DFHSM to restore the data set. Because CA-Disk functions differently from DFHSM, the products do not recognize that a data set is archived by CA-Disk and can encounter errors later. This hook, in conjunction with the Catalog Management hook, provides a means of overcoming this problem.

This hook is designed to work on your system, whether or not you have DFHSM. It is automatically installed with the Catalog Management Hook.

A catalog locate is usually issued by a routine trying to establish the existence of a data set. If the locate is successful, the data set exists; if it fails, it either does not exist or it is not cataloged. The key for the catalog management hook to go to work is when the locate is issued from a selected function (dynamic allocation, for example) and the return from catalog management is successful, but the data set is cataloged to the CA-Disk pseudo-volume. (The pseudo-volume--the CA-Disk default name is ARCIVE--is an imaginary volume to which CA-Disk optionally re-catalogs a data set when it is archived and scratched, to help identify the data set as being archived.) When these conditions are met, CA-Disk automatically restores the data set and then returns to the requestor of the locate the real volume to which it was restored. In this way, the locate requestor is never aware that a restore operation took place. We discuss how to catalog data sets to the pseudo-volume later in these customization instructions.

Programs that check for DFHSM's MIGRAT volser can have problems during the catalog management process of auto restore. For these programs, CA-Disk does not actually auto-restore the data set. It passes back the MIGRAT volser contained in the catalog entry, instead of the ARCIVE volser. This action allows the programs to issue a request to DFHSM to restore the data set. Then, CA-Disk intercepts the request with the DFHSM SVC Hook and invokes an auto-restore.

CA-Disk can recognize that the DFHSM request is for a CA-Disk archived data set by doing its own catalog locate. If the ARCIVE volser is returned, CA-Disk was the program that archived the data set.

In the preceding description of the catalog management hook, we said that only specific functions causes auto-restores to take place. For example, it is not good to automatically restore all data sets referenced by an IDCAMS LISTCAT job. Instead, these locates are ignored by the catalog management hook so that the output

listing show a volume of ARVICE for the data set. However, CA-Disk intercept locate requests from dynamic allocation, job initiation, and certain ISPF/TSO requests.

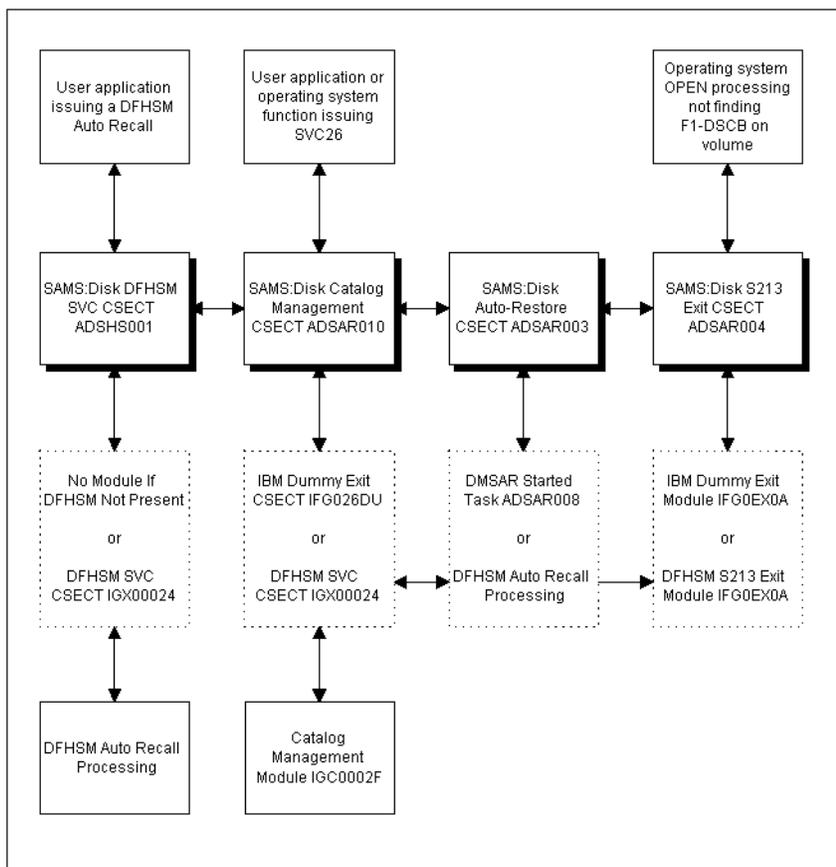
If you use IDCAMS to delete a data set that has been cataloged to the CA-Disk pseudo-volume, the data set is uncataloged from the pseudo-volume instead of being auto-restored. The return code for the IDCAMS program is set to indicate the delete was successful. If the data set cannot be uncataloged (possibly for security reasons), CA-Disk attempts to auto-restore the data set, so that IDCAMS can attempt the delete.

You see in the installation section that CA-Disk does not actually replace IFG0EX0A, IGG026DU and IGX00024 modules because there are other software products that also replace these modules. To allow the user to install both products, we have decided to call our modules by different names. In the installation process, you dynamically or statically relink the appropriate IBM programs and define their entry points as the CA-Disk modules. We bring up this topic now because it is important if your installation is also using IBM's program product DFHSM.

## Conceptual Diagram of CA-Disk Auto-Restore Methods

CA-Disk's catalog management hook uses CSECT ADSAR010 as part of module IGG026DU, and ADSHS001 as part of the module IGX00024. CA-Disk's F1-DSCB missing exit uses CSECT ADSAR004 as part of module IFG0EX0A. After installing the appropriate modules, you have the following conceptual arrangement:

FIGURE 6. Conceptual Diagram of Auto-Restore Method



This diagram shows that control is passed from the user doing a catalog locate to the catalog management hook and then to either the DFHSM hook or a dummy module. From there, control is passed to catalog management.

Catalog management performs its specified function and returns control to the next module up the chain, giving DFHSM a chance to auto recall the data set if appropriate. DFHSM returns control to the next module up the chain. Note that if the dummy module is installed, it just returns control up the chain without any other processing.

CA-Disk now gets control. If DFHSM had successfully recalled the data set, it now appears to CA-Disk that the data set exists on a real volume. If the data set is cataloged to the CA-Disk pseudo volume, CA-Disk takes one of two actions. If the program issuing the SVC 26 is recognized as one that needs the special handling provided by the CA-Disk DFHSM SVC hook, it returns a volser of MIGRAT. If CA-Disk does not recognize the calling program, CA-Disk attempts to auto-restore the data set by using the DMSAR started task. If either DFHSM or CA-Disk auto-restored the data set, it appears to the user that the data set exists on a real volume.

This design allows auto-restores from either the CA-Disk archives or DFHSM migration volumes on the same system without interference. Other software products such as SAMS:Allocate and CA-ACF2 also front- and back-end catalog management.

The diagram shows that control is passed from the user invoking a DFHSM Auto Recall to the DFHSM SVC hook. Then, control is passed to the catalog management hook. From this point on, the catalog management hook process is as documented above. When the auto-restore is complete, control is returned to the DFHSM SVC hook. If the data set was not archived by CA-Disk, control is passed to the DFHSM SVC for its own Auto Recall processing.

Again, by examining the diagram you can see that if the F1-DSCB missing exit is invoked, CA-Disk gets control first in its exit as CSECT ADSAR004 of module IFG0EX0A. If it cannot restore the data set, it calls the CSECT IFG0EX0A to attempt the restore before returning to the user.

## Installing the Auto-Restore Function

The auto-restore function applies to MVS installations only, and should be implemented only after a thorough review of the following installation procedure. Be sure to complete each step in the process before continuing to the next. The following discussion applies to all three hooks.

1. Review the procedure named "DMSAR" in the proclib library shipped with your CA-Disk system. In particular, review the following items:

---

*Note:* If you rename the DMSAR catalog procedure, you must also change the value of sysparm ARTASKNM (the default name is DMSAR). Otherwise, the support does not work!

---

- Review the dispatching priority assigned to the procedure and make any necessary adjustments based on your installation's requirements.
- Supply the proper data set name for the STEPLIB.
- If you want dynamic tape allocation, leave the asterisk in column 3 of the ARCHIVES DD statement. If you do not want dynamic allocation to be performed by CA-Disk, remove the asterisk. The preferred method is to let CA-Disk dynamically allocate the tape device, since this allows requests that do not require tape devices to be processed immediately, even if no tape devices are available. This condition occurs for those data sets that have been archived to disk, have no record in the CA-Disk archive index, or are rejected by the user screening exit ARESPREX.

If you put an ARCHIVES or ARCHIVER DD statement in your JCL, do not specify the DSN= parameter. If you specify a data set name, no concurrent auto-restores can take place. In addition, you cannot be able to run auto-restores at the same time as CA-Disk backup, archival, restore or recover jobs.

If the archives reside on disk rather than tape, the appropriate disk archive data set is dynamically allocated with no operator intervention required.

- Review the IDCUT1 and IDCUT2 DD statements. These statements were discussed in [“Activating VSAM Support” on page 41](#).
- If your installation maintains multiple files data sets and you want to make them all available to the auto-restore function, use the MFILES DD statement instead of the FILES DD statement. Add each files data set name to the MFILES concatenation, up to 256 files data sets are supported. The order of the concatenation determines the order of search for the data set. The first files data set that contains an index record for the archived data set is used, even if another files data set contains a more recent version of the data set. Therefore, it is very important that files data sets be specified in the proper order.

---

*Note:* The FILES DD statement is ignored if there is also an MFILES DD statement in the procedure. Note that concatenated files data sets are not supported with the FILES DD statement.

---

- Ensure that the DMSAR proc is in a proclib accessible during an MVS START command; for example, SYS1.PROCLIB.
- Users with security packages should note that since the CA-Disk auto-restore started task cannot be identified by a JOB statement or a LOGON procedure, most security packages have a Started Procedures Table to associate a started task name with a user ID and possible group name. If your security package protects resources accessed by started tasks and you would like to auto-restore data sets protected by your security package, you must create an entry in your Started Procedures Table to associate the CA-Disk auto-restore started task name DMSAR with a user ID and possible group name. For RACF, the Started Procedures Table is documented in the IBM manual RACF Installation Reference and in the IBM manual Systems Programming Library: RACF.
- Now that you have the DMSAR started task identified to your security package, anyone can auto-restore user data sets in the CA-Disk archives. Although this causes no security exposure (after an auto-restore completes, OPEN, SCRATCH, RENAME or IDCAMS processing prevents user access to data sets to which they are not authorized), two methods have been provided which control this effect.

### Front-End Method:

This method is always in effect and works well if your security package does not use strict volume rules (for example, a user might have access to a data set on one volume, but does not have access to it on another volume), and you do not make extensive use of discrete RACF profiles.

CA-Disk auto-restore processing uses the System Authorization Facility (SAF) to determine if the user has READ access to the data set name in question. Unfortunately, the auto-restore front-end processing does not have actual volume or discrete RACF profile information available. If queried from the catalog management hook, CA-Disk passes to SAF the CA-Disk pseudo- volume volser along with the data set name. If queried from the S213 exit, CA-Disk passes to SAF the volume on which the OPEN was attempted along with the data set name. CA-Disk allows the auto-restore to proceed only if it receives a return code of less than 8 from SAF.

If you are a CA-ACF2 user, you must activate your SAF interface to let the SAF call issued by CA-Disk work properly. Also, you must add the following entry to your SAFPROT GSO list:

```
SAFPROT.DMS CLASSES(data set) CNTLPTS(ADSAR010)
SUBSYS(DMS)
```

**Back-End Method:**

If you use the CA-Disk RACF security interface, or if you use the CA-Disk CA-Top Secret security interface and have implemented the facility activated by sysparm ARSECURE (see [“Installing the CA-Top Secret Security Interface” on page 29](#)), you can use this additional method.

To install this method, specify sysparm ARSECURE with a value of R or Y. For more information, see the sysparm description for [“ARSECUREn” on page 129 in the Systems Guide](#).

2. Review the documentation for sysparms ARESUNIT, ARESUNIC, and ARESUNIn (if multiple types of cartridge devices are used). If the defaults are unacceptable, specify their values in your production sysparms.
3. Review the documentation for sysparm ARTAPEOK. This sysparm controls the degree to which tape mounts will be accepted--if at all--from auto-restore tasks.
4. Review the documentation for user exit ARESPREX. This user screening exit can selectively reject auto-restore requests, and can also perform logging activities.
5. If you are installing the catalog management hook and the DFHSM SVC hook, you can use pool support as well. Pool support allows you to dynamically decide the volume to which to restore a data set, based on a combination of pool definitions and an optional user exit. See [“Setting Up DASD Pools for Auto-Restore” on page 70](#). This does not have to be done at installation time. You can install the basic auto-restore facilities first and add pool support later.
6. The catalog management hook and DFHSM SVC hook of CA-Disk are installed with a started task. A system IPL is not required. You install the hooks by issuing the following command from the operator's console:

```
S DMSAR ,DMSAR=INSTALL
```

Output is directed back to the operator's console. When the interface is successfully installed, this started task will end.

**Implementation Notes:**

- You must issue this command on each CPU for which you want CA-Disk's auto-restore function implemented.
- The CA-Disk catalog management hook and DFHSM SVC hook are removed each time you IPL. Therefore, you must use the above procedure to reinstall the hooks when you IPL. Consider installing the command in the automatic start-up procedures.
- If you run SAMS:Allocate, issue the start command for DMSAR before the start command for SAMS:Allocate.

You can check if the CA-Disk catalog management hook and DFHSM SVC hook are installed by issuing the operator command:

```
S DMSAR ,DMSAR=STATUS
```

or simply:

```
S DMSAR
```

You can remove the CA-Disk catalog management hook and DFHSM SVC hook by issuing the operator command:

```
S DMSAR ,DMSAR=REMOVE
```

If you do issue this removal command, there are several processing steps to be aware of:

After a remove command is accepted, the SVC table entries for SVC 26 and the DFHSM SVC are modified to point to their original addresses. This disables the hooks. The removal task waits eight seconds so that current requests can complete, then DMSAR deletes its hooks from CSA memory. The hooks are completely removed and the started task terminates.

- If you plan to use a CA-Disk pseudo-volume name other than ARCIVE, it must be specified through sysparm RECATVOL. Do not select a pseudo-volume name that is a real volume name, or likely to become one. If your pseudo-volume name is a real volume, you cannot reference your data sets on that volume through the catalog. Set the pseudo-volume name as the value for the sysparm RECATVOL. For related information, see the sysparm description for “[PSUDEVTpN](#)” on [page 159 in the Systems Guide](#).
- An auto-restore diagnostic facility exists within both the Catalog management hook and the DFHSM SVC hook which can display applicable data used to determine if an auto restore is to be performed. These diagnostic messages are available if you have installed the Test Auto-Restore Interface as documented below, and if your job name is equal to the test jobname (default is TESTDISK). If you want to change this default jobname, simply zap the jobname you prefer. Refer to the TSTHOOKS member in the INSTALL library and installation instructions for the Test Auto-Restore Hook for further information. Similar diagnostic facilities are provided for the CA-Disk SVC module and VSAM date stamp module. The default jobname for the other diagnostic facilities is DMSTEST, so you can make this the same name for all functions. The ability to change the job names for the other functions is discussed where applicable.

---

## *Test Auto-Restore Hook Interface*

### **Overview**

This function lets you install a new or different release of the Auto-Restore operating system hook simultaneously with your production Auto-Restore hook. The Test Auto-Restore Hook only operates on jobs/tasks which have a certain predefined name or DD statement. It also allows you to reproduce problems and receive diagnostics

through the use of sysparm ARDIAGNM. Your production Auto-Restore hook continues to operate on all other jobs. This lets you safely test changes or upgrades to your Auto-Restore function while you leave your production CA-Disk system running.

## How it Works

CA-Disk Auto-Restore Hook have program logic at their entry points which determines if each instance of the hook is to run or not. This determination can be based on either a specific job/task name or DD statement. In the case of the Test Auto-Restore Hook, care has been taken to ensure that the production hook runs only when the test hook does not run, and vice versa.

The Test Auto-Restore Hook only operates on jobs/tasks which have a certain predefined name as their job/task name, or on jobs which include a certain pre-defined DD statement name. This is configurable using the following 2 system parameters:

- ARJOBEXC — the value of this system parameter is the pre-defined job name. It can also be a task name (i.e., a userid). For detailed information, see sysparm description for [“ARJOBEXC\\_jobname” on page 129 in the Systems Guide](#).
- ARDDNEXC — the value of this system parameter is the pre-defined DD statement. For detailed information, see the sysparm description for [“ARDDNEXC\\_ddname” on page 128 in the Systems Guide](#).

## Components of the Test Auto-Restore Hook Interface

The Test Auto-Restore Hook Interface consists of the following components:

**TABLE 13. Test Auto-Restore Components**

Component	Explanation
TSTHOOKS member in the INSTALL library. (Hook customization job.)	This job creates the Test Auto-Restore Hook. This job copies the production hook modules to test names and then ZAPs them so that they use the test job name and test Auto-Restore PROC. Run this job even if you just want to use the default options.
TSTAR member in the PROC library. (Test Auto-Restore PROC.)	This PROC is the same as the DMSAR PROC, except that it has been modified for use by Test Auto-Restore. This PROC is started by the Test Auto-Restore hook in order to perform a test Auto-Restore.
TSTEXMPT member in the INSTALL library. (Hook exempt job.)	This job refreshes the production Auto-Restore Hook to begin excluding the job/task name and/or the DD statement name based on the value specified for sysparms ARDDNEXC and ARJOBEXC.
TSTINSTL member in the INSTALL library. (Hook install job.)	This job dynamically installs the Test Auto-Restore Hook and begins to accept auto-restores from the job/task name and/or DD statement name based on the value specified for sysparms ARDDNEXC and ARJOBEXC. You do not need to IPL the system to install this hook.
TSTSTATS member in the INSTALL library. (Hook status display job.)	This job displays the status of the CA-Disk Auto-Restore operating system hook on the system console.
TSTREMOV member in the INSTALL library. (Hook remove job.)	This job dynamically removes the Test Auto-Restore Hook from your system. You do not need to IPL your system to remove this hook.
SVC 26 Hook Module.	Module TSTAR010 (a modified version of module ADSAR010).
SVC 109 Hook Module.	Module TSTHS001 (a modified version of module ADSHS001).

## Preparing to Use the Test Auto-Restore Hook Interface

There are several things which you must do before you use the Test Auto-Restore Hook feature:

1. Tailor the following jobs provided in the INSTALL or PROC libraries on your CA-Disk distribution tape: TSTHOOKS, TSTINSTL, TSTREMOV, TSTSTATS, TSTEXMPT, and TSTAR. You need to tailor the job accounting, load library name (DD=STEPLIB), and the parameter library name (DD=PARMLIB).
2. Tailor the Q= parameter in the TSTAR PROC. This parameter provides a data set name prefix for the CA-Disk libraries in your system.

- You must select the job name and/or the DD statement you want to have trigger the Test Auto-Restore feature. This is accomplished through the ARJOBEXC and ARDDNEXC system parameters. After selecting a job name and/or DD statement, supply these sysparms to the TSTEXMPT member in the form on a sysparm override as shown in the following figure:

FIGURE 7. Sample JCL for the TSTEXMPT member

```

EDIT (TSTEXMPT) 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOBNAME JOB (ACCT INFO)
000200 /** REFRESH THE PRODUCTION VERSION OF THE CATALOG MANAGMENT HOOK **
000300 /** TO EXCLUDE THE "TSTAR010" TEST HOOK (SEE TSTINSTL). **
000400 //DIMCAT EXEC DIM,CMD=REFRESH,SVC=26,HOOK=ADSAR010
000500 //MSGPRINT DD SYSOUT=*
000600 //PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
000700 //SYSPARMS DD *
000800 ARJOBEXCDMSTJOB
000900 ARDDNEXCDMSTDDN
001000 /** REFRESH THE PRODUCTION VERSION OF THE HSM HOOK **
001100 /** TO EXCLUDE THE "TSTHS001" TEST HOOK (SEE TSTINSTL). **
001200 //DIMHSM EXEC DIM,CMD=REFRESH,TYPE=ESR,SVC=(3,24),HOOK=ADSHS001
001300 //MSGPRINT DD SYSOUT=*
001400 //PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
001500 //SYSPARMS DD *
001600 ARJOBEXCDMSTJOB
001700 ARDDNEXCDMSTDDN
001800 /*
***** ***** Bottom of Data *****
Aa A TCP/IP R 4 C 15 16:06 11/24/97

```

When this job is executed, it refreshes the production Auto-Restore Hook to begin excluding the job/task name and/or the DD statement name based on the value specified for sysparms ARDDNEXC and ARJOBEXC.

**CAUTION:** The same values you specify for sysparms ARJOBEXC and ARDDNEXC in the TSTEXMPT job, must also be specified in the TSTINSTL job. Otherwise, unpredictable results occur.

## Installing the Test Auto-Restore Hook Interface

The Test Auto-Restore Hook Interface can be installed by carefully following these step by step instructions:

- Locate and submit the TSTHOOKS member you customized. This job creates the Test Auto-Restore environment by copying your production Auto-Restore modules to test names.
- Locate and submit the TSTEXMPT member you customized. This job refreshes the production Auto-Restore Hook to begin excluding the job/task name and/or the DD statement name based on the value specified for sysparms ARDDNEXC and ARJOBEXC.

3. Locate and submit the TSTINSTL member you customized. This job dynamically installs the Test Auto-Restore Hook and begins to accept auto-restores from the job/task name and/or DD statement name based on the value specified for sysparms ARDDNEXC and ARJOBEXC.

## Operating the Test Auto-Restore Hook Interface

One of the objectives of this interface is to allow you to activate an Auto-Restore environment at a release level different from that of your production system. The Test Auto-Restore system you just installed only operates on the job/task name and/or DD statement name that match the values you set for ARDJOBEXC and ARDDNEXC sysparms.

For example, with sysparm ARDDNEXC set to a value of TESTAR, the Test Auto-Restore Hook Interface is invoked for all jobs displaying the TESTAR DD statement as shown in the following figure:

FIGURE 8. Sample JCL to invoke Test Auto-Restore

```

File Edit Transfer Options Connection Macro Window Help
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT                               (TESTAR) - 01.00                Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOB CARD JOB (XXXX,XXX), 'XXXXXXXX',
000200 //          CLASS=X,MSGCLASS=T,NOTIFY=&USERID
000300 //-----
000400 //* THIS JOB WILL INVOKE THE SAMS:DISK TEST AUTORESTORE
000500 //* FACILITY (TSTAR), VIA DDNAME. TESTAR VALUE SHOULD MATCH
000600 //* THE ARDDNEXC VALUE IN INSTALL LIBRARY MEMBER TSTINSTL.
000700 //-----
000800 //STEP1 EXEC PGM=IDCAMS
000900 //TESTAR DD DUMMY
001000 //SYSPRINT DD SYSOUT=*
001100 //SYSIN DD *
001200 PRINT IDS(ARCHIVE.DATASET)
001300 /*
***** ***** Bottom of Data *****
  
```

Note: The ARDDNEXC only operates successfully on archived data sets that are referenced dynamically.

The convenient quality of this environment is that you can now thoroughly test this new Auto-Restore system without impacting your production system.

## Sysparm ARDIAGNM

There is a diagnostic feature in this interface that can help you solve problems related to Auto-Restore. To activate this feature, simply assign a job name value to sysparm ARDIAGNM. Setting this sysparm instructs the Test Auto-Restore Interface to issue diagnostic messages to each job that enters the system with this job name.

The following figure shows some of the typical messages routed to SYSLOG when the ARDIAGNM sysparm is activated:

FIGURE 9. Sample SYSOUT of ARDIAGNM

```

SDSF SYSLOG 11627.103 SIV9 SIV9 11/24/97 LINE 49,844 COLUMNS 51 130
COMMAND INPUT ---->
0090 ADSAR010 DIAG01: SUPPL=00001152 7F7326B8 009D69A8 009D6998
0090 ADSAR010 00000000 00000400 00000000 00000000
0090 ADSAR010 DIAG02: PROGRAM INITIATING CALL WAS: IEFBR14
0090 ADSAR010 DIAG03: ENTNAME=ISPDLM1.DMS.JCL
0090 ADSAR010 DIAG04: CATNAME=SSL801ICFCAT.USSL801
0090 ADSAR003 DIAG99: RESTORE INITIATED BY INTERFACE
0090 DMS2987 DATA SET CATALOGED TO SAMS:DISK PSEUDO-VOLUME ARCIVE
0090 DMS2987 SAMS:DISK HAS ARCHIVED ISPDLM1.DMS.JCL
0090 ADSAR003 DIAG99: AUTORESTORE BY START CMD=DMSAR
0294 S DMSAR.DMS00037,,,ARCB=00C24AB4,ASID=00037,RTIME=1737330 ** SAMS:DISK
AUTO RESTORE **
0081 $HASP100 DMSAR ON STCINRDR
0290 IEF695I START DMSAR WITH JOBNAME DMSAR IS ASSIGNED TO USER DMSAR
, GROUP SYSSTC
0091 $HASP373 DMSAR STARTED
0090 IEF403I DMSAR - STARTED - TIME=17.37.34
0090 VAM0100 SAMS:Allocate RELEASE 5.3.0NV
0090 VAM0100 SAMS:Allocate ASR ROUTINE IS ISPMCD1.R53.INSTALL(ASALLV )
0090 VAM0100 SAMS:Allocate STORAGE POOL IS ISPMCD1.R53.INSTALL(TSOST )
0090 VAM0100 SAMS:Allocate IS RUNNING RESIDENT
0090 VAM0100 SIV9 01/06/95 JBB5522 SP 5.2.2 9672 JES2 DFSMS 1.3.0
0090 VAM0100 SAMS:Allocate LAST SUCCESSFUL INSTALL OR REFRESH
Aa A TCPIP R 2 C 21 17:46 11/24/97

```

For detailed information, see the sysparm description for “ARDIAGNM\_jobname” on page 128 in the [Systems Guide](#).

## Member TSTREMOV

After your testing has completed, shutdown the Test Auto-Restore Hook Interface. Doing so reverses what was excluded from your production Auto-Restore while the Test Auto-Restore Hook were active. The Test Auto-Restore Hook Interface can be removed by carefully following these step by step instructions:

1. Locate and submit the TSTREMOV member you customized. This job dynamically removes the Test Auto-Restore Hook.

2. Issue the REFRESH command to your production Auto-Restore system. This command sets sysparms ARJOBEXC and ARDDNEXC back to their default values, which deactivates them. The following command is an example of the REFRESH:

```
S DMSAR ,DMSAR=REFRESH
```

---

## *Setting Up DASD Pools for Auto-Restore*

This section applies only if you have installed—or plan to install—the catalog management hook and DFHSM SVC hook. Using the pool support is completely optional.

The purpose of the pool support in the Auto-Restore facility is to allow the maximum flexibility possible in determining the target volume to which the data set is restored. Although this capability is not essential to the basic function of Auto-Restore, it does relieve the DASD administrator of a lot of the manual effort required to manage free space distribution among packs. Perhaps the best way to explain its advantages is to show what management is like without it.

Without pool support, any data set that is Auto-Restored is required to go back to the volume from which it came. Two problems become obvious:

1. What happens if the volume no longer exists?
2. What if there is not enough space for the data set to fit or the available space is too fragmented?

The answer is that the restore fails and the task that requested the restore fails. This is not too serious if a TSO/ISPF user caused the restore, but it can cause problems if a production job abends because it could not gain access to the data set.

Pool support within the Auto-Restore facility is accomplished separately for VSAM and non-VSAM data sets.

### For VSAM Data Sets:

For VSAM data sets, pooling is technically complex, and must be accomplished by SAMS:Allocate, Computer Associates's Volume Allocation Manager, or some other allocation control product. It is not done within CA-Disk itself. If SAMS:Allocate or some other allocation control product is not used, Auto-Restore must place VSAM data sets back on the volumes from which they were archived. If you do not have an allocation product available, consider using [“RESPRIEX — Screen Restore Requests” on page 237 in the Systems Guide](#).

### For Non-VSAM Data Sets:

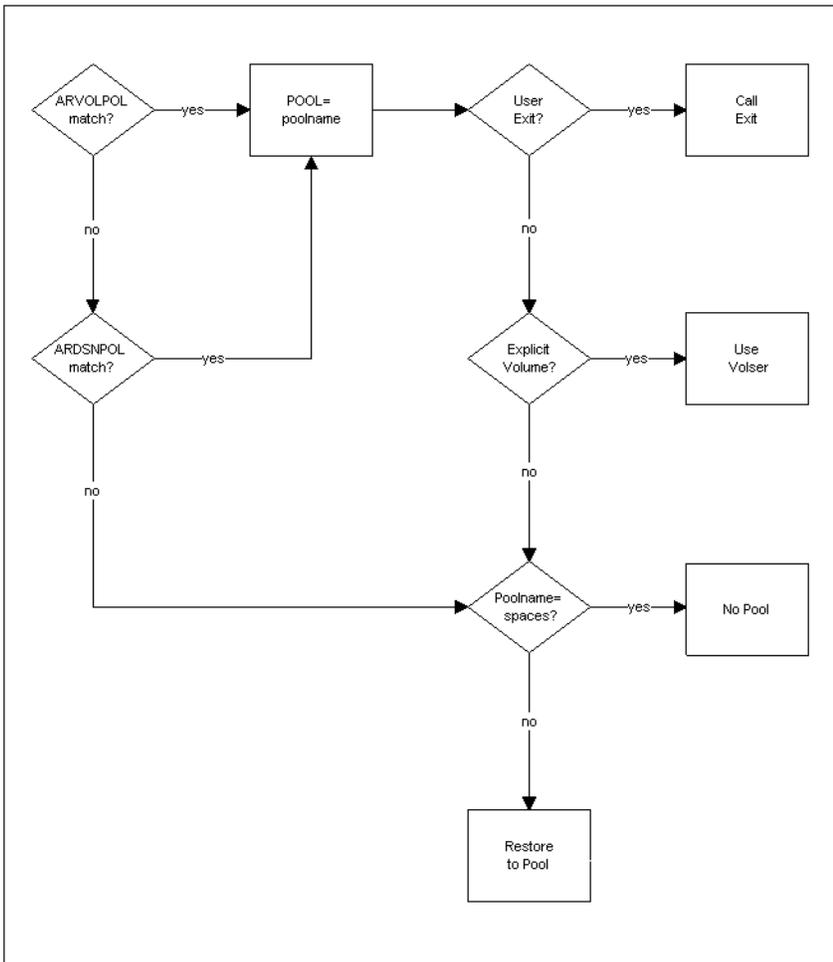
For non-VSAM data sets, pooling can be accomplished through either or both of two methods. The first method is by using SAMS:Allocate or some other allocation control product. The second method is accomplished within CA-Disk's Auto-Restore facility, and can be used alone or in conjunction with SAMS:Allocate.

Several advantages become apparent when volume pooling is implemented, such as:

- The data set can be restored to any volume in the pool that has sufficient free space. Each volume in the pool is searched until one with sufficient free space is found, or until it is determined that none of the volumes in the pool has enough free space.
- If a volume (or a string of volumes) becomes unavailable, specify an entry in the volume pool that redirects all restores to that volume to a different pool of volumes.
- You can reserve a set of special volumes used only for Auto-Restores as its own pool. This allows you the capability of managing this pool differently than perhaps your TSO or production packs.
- You can code your own user exit to decide the pool to which to restore, based upon criteria unique to your installation.

Pool support can be thought of as the algorithm used to determine the destination volume for a data set being Auto-Restored. It consists of several possible steps, depending on options you specify. Let's look at a schematic of the process, then we'll come back and fill in the details.

FIGURE 10. Example of DASD Pool Support for DMSAR



The next couple of figures illustrate typical examples of how the syntax works for DASD Pools. These figures are then followed by a series of explanatory notes to help describe the illustration.

FIGURE 11. Sample source for ARVOLPOL sysparm

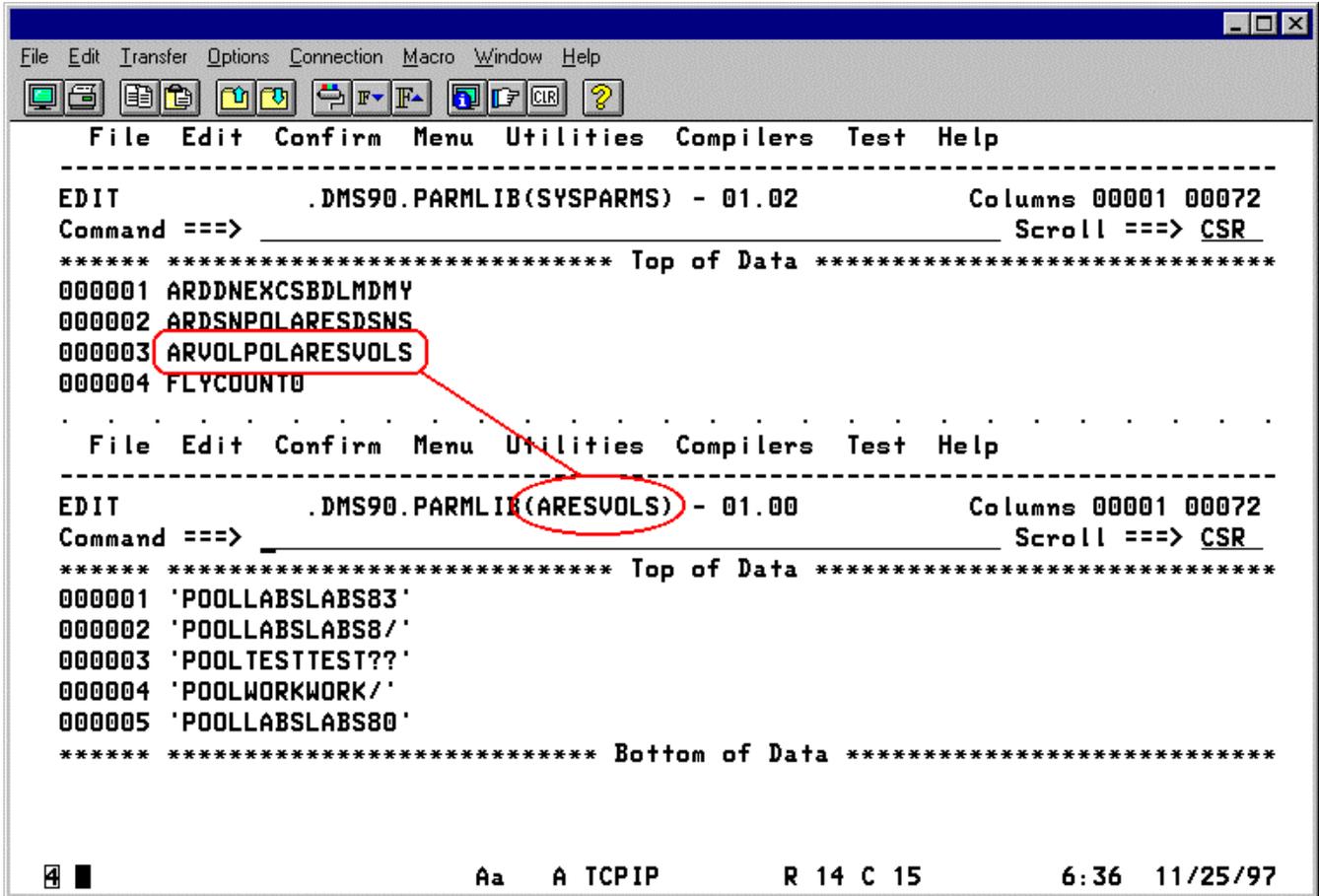


FIGURE 12. Sample source for ARDSNPOL sysparm

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          .DMS90.PARMLIB(SYSPPARMS) - 01.02          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 ARDDNEXCSRDLMDMY
000002 ARDSNPOLARES DNS
000003 ARVOLPOLARES VOLS
000004 FLYCOUNT0
. . . . .
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          .DMS90.PARMLIB(ARES DNS) - 01.00          Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 'POOLTESTA.B.C'
000002 'POOLLABSD.E./'
000003 'POOLLABSD./?'
000004 'POOLTEST???.*.TEST/'
000005 'POOLWORK/S80'
***** ***** Bottom of Data *****

```

Aa A TCPIP R 20 C 23 6:53 11/25/97

FIGURE 13. Sample source for DASDPOOL member in PARMLIB

```

File Edit Transfer Options Connection Macro Window Help
-----
EDIT          .DMS90.PARMLIB(DASDPOOL)  01.00          Columns 00001 00072
Command ==>  _____          Scroll ==> CSR
***** ***** Top of Data *****
000001 'LABS80POLLABS'
000002 'LABS81POLLABS'
000003 'LABS83POLLABS'
000004 'PACK01TSODATA'
000005 'PACK12TSODATA'
000006 'PACK15TSODATA'
000007 'PACK20TSODATA'
000008 'TEST01POOLTEST'
000009 'VOL003RJEWORK'
000010 'VOL004RJEWORK'
000011 'VOL006RJEWORK'
000012 'WORK01POOLWORK'
***** ***** Bottom of Data *****
Aa  A TCPIP          R 14 C 9          7:04  11/25/97

```

*Note:* All pools referred to in ARVOLPOL and ARDSNPOL must be defined in the member DASDPOOL in your parmlib data set. Extensive rules for coding this member are found in [“DASD Pool List” on page 481 in the Systems Guide.](#)

*Note:* All standard CA-Disk pattern matching capabilities are supported in both ARVOLPOL and ARDSNPOL members. The first entry that causes a match causes its corresponding pool name to be used, even if there are other entries in the member that can match. This means start each of these members with the most specific entries first and have the least specific entries entered last.

*Note:* Pool capability for VSAM data sets is supported by SAMS:Allocate or another allocation control product, not by the methods just described.

To use SAMS:Allocate or another allocation control product for non-VSAM data set pooling, create an ARVOLPOL with at least one volume in it (a volume that is normally empty or used for temporary data sets only). This is for two reasons:

1. CA-Disk always checks to see if the needed volume is online and has space available before attempting to allocate a data set it is restoring. If the original volume is offline or does not have enough space, CA-Disk fails the restore without attempting to allocate the data set. No allocation control product gets control. By supplying an ARVOLPOL, you provide CA-Disk with an online volume to which it can attempt to make allocations.
2. An ARVOLPOL gives you a means of seeing how well your allocation control product rules are doing. By monitoring the volumes defined in this ARVOLPOL, you can determine which data sets are not being controlled by your rules. You can update your rules to prevent similar problems from occurring later.

Going back to the schematic, we start at Step (a) to determine if a source volume pool match is found. There are actually two steps to this process, which also applies to Step (b). First CA-Disk retrieves sysparm ARVOLPOL. If you have not specified this sysparm, a no-match condition is assumed. If you do specify the sysparm, the value you supply for it must be the name of the member in your parmlib that contains the pool entries. For instance, if a value of ARVOLPOLVOLPOOL is specified, you must code your entries for this pool in the member VOLPOOL in your parmlib. Assuming this pool is defined, CA-Disk compares the data set's original source volume to each explicit or pattern volume in the table. If a match is found, the pool name associated with that entry is used and processing continues to Step (d). If no match is found, or if sysparm ARVOLPOL is not specified, processing goes on to Step (b).

Step (b) follows the same general logic as Step (a). You notify CA-Disk of the presence of a data set name pool by specifying sysparm ARDSNPOL with the member name that contains the pool associations. In this pool, CA-Disk compares each entry to the data set name being restored. As soon as an explicit or pattern name is matched, that pool name is used and processing continues with Step (d). If no match is found, the pool name is initialized to spaces (Step (c)), indicating that no pool name is used in the restore.

We are now at Step (e), which is an optional user exit that can be coded. If you want to write a special pool selection routine, this is where you add it. CA-Disk calls the exit (Step (f)) when sysparm ARPOOLEX is specified with the module name to be called. For more information, see the User Exit description for ["ARPOOLEX — Auto-Restore DASD Pool Exit" on page 210 in the Systems Guide](#)). With this exit, you can see if any pool was selected, based on the entries in the ARDSNPOL and ARVOLPOL members. If no match was found or neither of the members were defined, the pool name is blanks (Step (i)); otherwise it contains the name of the pool that are used (Step (k)). You can change the pool name to another valid pool name or it can be blanked out to suppress pool support for the restore. You also have the option of passing back an explicit target volume (Step (g)), which overrides any pool specification (Step (h)). The diagram shows that there is no requirement to use the pool support that CA-Disk provides--just by coding the user exit you can customize pool support.

If, after the processes above, you exit with the name of a pool, this pool must be defined in the member DASDPOOL in your parmlib. The eligible volumes to which the data set can be restored are all the entries in this member that have the same pool name.

At this point, read ["RESTORE/RECOVER" on page 237 in the User's Guide](#). A detailed description of the rules CA-Disk follows in determining the volume to which a data set is restored.

## *Customizing the TSO/ISPF Auto-Restore Environment*

This section applies only if you have installed the catalog management hook. If you have installed the hook but have not attempted any restores from a TSO/ISPF environment, do so at this time. It is much easier to conceptualize the modifications about to be described if you are familiar with how the support works in its default mode.

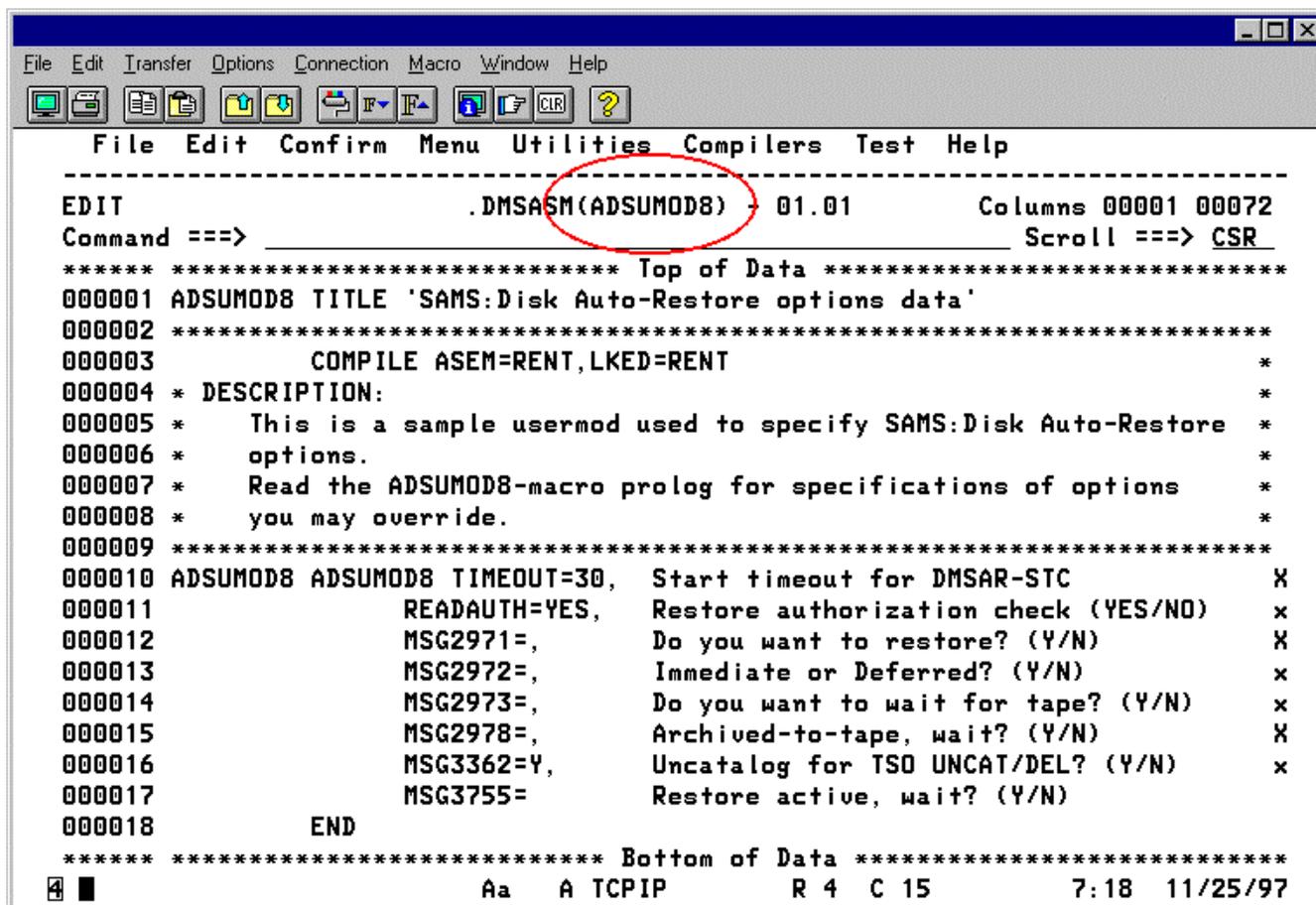
In its default mode, the TSO/ISPF user is prompted for answers to the following questions:

1. DO YOU WANT TO RESTORE THE DATA SET? (yes/no) If the user replies anything other than “no”, the next question is asked.
2. DO YOU WANT AN IMMEDIATE OR A DEFERRED RESTORE? (I/D) If the user responds anything other than “deferred”, the next question is asked.
3. DO YOU WANT TO WAIT FOR THE RESTORE TO COMPLETE? (yes/no) If the user answers “no”, the terminal will be unlocked and the user is notified when the restore is complete. If the user responds anything other than “no,” the terminal remains locked until the restore is complete, unless a tape mount is required (and allowed by sysparm ARTAPEOK). If a tape mount is required, the user is prompted one more time with the following question:
4. DATA SET IS ARCHIVED TO TAPE. DO YOU STILL WANT TO WAIT? (no/yes) If the user responds “yes”, the terminal is remain locked until the restore is complete. If the user responds anything other than “yes”, the terminal will unlock and the user will be notified when the restore is complete. The Auto-Restore task begins before the user responds to this prompt, and the response has no effect on the progress of the Auto-Restore job.
5. A WAIT RESPONSE WILL SUSPEND YOUR SESSION. DO YOU WANT TO WAIT (Y?N)? This message allows the TSO user another option in the wait process in instances where another task is restoring the data set.

This series of questions can be very confusing to some users. To suppress any or all of the questions, you can. This is done by installing user exit "USERMOD8" as follows:

1. Locate the JCL for ADSUMOD8 in the library associated with the DMSASM DDDEF. Below is a sample of that JCL:

FIGURE 14. Sample source for ADSUMOD8



2. In order to ensure that the changes you make to ADSUMOD8 are protected during future CA-Disk installs or maintenance, copy this member into the source library associated with the //USERASM dd statement in USERMOD8.
3. Customize ADSUMOD8 as follows:
  - The TIMEOUT= parameter defaults to 30 seconds before the STC passes control back to the USERID. After this wait period is exceeded, message 2977 is issued and the TSO user (or console operator in the case of a batch job) is asked if the Auto-Restore should be terminated. A "Y" response terminates the Auto-Restore. Any other response extends the wait period an additional 30 seconds. If your installation is prone to resource conflicts which slows task initiation, consider increasing this wait period and eliminate this additional prompt. This can be done by increasing the value of this parameter.
  - The READAUTH= parameter defaults to "YES" forcing DMSAR to invoke a SAF check for Read authority based on the USERID invoking the Auto-Restore prior to starting the STC. Specifying "NO" bypasses this security prior to starting the DMSAR. In ACF2, or depending upon the setting of ARSECURE sysparm in a RACF or Top Secret environment, the data set could be restored when the USERID requesting the restore had no read authority to the data set. Although this causes no real security exposure, it does

waste resources. It is not recommended that customers modify this value in the ADSUMOD8 source unless they encounter a specific security situation where they need this ability. Refer to [“Security Processing” on page 311 in the Systems Guide](#) for more details of security protection during Restore for all security interfaces.

- The MSGnnnn= parameters are used to customize the default responses to the TSO prompts. Specify a “Y” to pre-answer the TSO response as YES. Specify a “N” to pre-answer the TSO response as NO. Specifying the “null parameter” causes auto-restore to prompt the TSO user for a response. For MSG=2972, specify an “I” to pre-answer an immediate restore, a “D” to pre-answer a deferred restore, or the “null parameter” causes auto-restore to prompt the TSO user for a response.

4. Save your work by issuing SAVE at the TSO command line.

## USERMOD8

USERMOD8 must be installed using SMP/E. Sample JCL is provided for you in member USERMOD8 of the install library. Install USERMOD8 in the following manner:

1. Locate the source for USERMOD8 in the install library. Below is a sample of that source:

FIGURE 15. Sample SMP/E JCL for USERMOD8

```

EDIT .INSTALL(USERMOD8) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
-----
***** Top of Data *****
000001 //JOBNAME JOB (ACCT INFO)
000002 /** * INSTALL AUTO RESTORE ANSWERS *
000003 /** * YOU MUST BRING DMSAR DOWN AND UP FOR CHANGES TO TAKE EFFECT *
000004 //SMP EXEC PGM=GIMSMP,REGION=5120K,
000005 // PARM='CSI=SAMSSMPE.GLOBAL.CSI'
000006 //SYSPRINT DD SYSOUT=*
000007 //SMRPT DD SYSOUT=*
000008 //SMPOUT DD SYSOUT=*
000009 //SMPHOLD DD DUMMY
000010 //USERASM DD DISP=SHR,DSN=USERID.DISK.ASM
000011 //SMPCNTL DD *
000012 SET BOUNDARY(GLOBAL).
000013 RECEIVE S(SDU1818) .
000014 SET BOUNDARY(DMSTGT) . /* <--- SAMS:DISK TARGET */
000015 APPLY S(SDU1818).
000016 //SMPPTFIN DD *
000017 ++USERMOD(SDU1818).
000018 ++VER(Z038) FMID(SDM1900) .
000019 ++SRC(ADSUMOD8) DISTLIB(ADMSASM) DISTMOD(ADMSLOAD) TXLIB(USERASM).
Aa A TCPIP R 4 C 15 7:27 11/25/97
    
```

2. Customize USERMOD8 by supplying the appropriate information for the following:
  - jobcard information
  - Global and Target CSI names
  - USERASM data set name
3. Submit USERMOD8
4. Refresh the Auto-Restore hook on all systems for this change to take effect. Issue the following command to accomplish this:

```
S DMSAR ,DMSAR=REFRESH
```

---

## *Auto-Restore Implementation Guidelines*

The hardest task in implementing the Auto-Restore function is in determining how much time is allowed to elapse between the time a data set is last referenced and when it becomes a candidate for archival. If this time window is too large, few archivals take place and a severe shortage of DASD space probably occurs. On the other hand, if the time window is too small, too many archivals take place. If this happens, the cost of archiving and restoring the data sets probably exceed any benefits realized by the temporary DASD space savings.

Obviously there are too many variables to determine an absolute rule in calculating the best time to archive a data set based on last use date. The best method appears to be by trial and error. First, pick a period of time that seems reasonable based on current knowledge of the user environment. Then set up a monitoring period and keep totals on the number of auto-restore requests being processed on a per-shift basis. If this number stays sufficiently low during the trial evaluation, and no shortage of DASD space is evidenced during this period, then you probably have struck upon a good balance between archiving and restoring. Your evaluation period should run through your center's month-end processing cycle to verify that this load does not produce an unreasonable number of restore requests. If it does, your archival scheme probably needs to be re-evaluated.

One issue that must be considered on the archival side is the increased catalog usage due to the need to recatalog data set to the CA-Disk pseudo-volume so they can be automatically restored.

One facility that can help in the maintenance of your catalogs is the specification of sysparm UNCATPSU with a value of Y. This causes both MERGE and IXMAINT to issue an uncatalog (if the data set is cataloged and does not have a F1-DSCB on the source volume) when the last DSNINDEX record for the data set is deleted from the archive index. We highly recommended that this sysparm be specified with a Y if you are using the Auto-Restore function.

The established CA-Disk customer faces the problem that all data sets that were archived under an old system of "scratch, uncatalog" are not accessible by the Auto-Restore function. To assist these customers, a procedure has been implemented to catalog data sets that are found in the archive index but that are no longer cataloged and do not have a F1-DSCB on the source DASD volume. This procedure, named IXCATLG, is supplied with CA-Disk and is documented in ["IXCATLG Utility" on page 115](#).

## VSAM Alternate Indexes

When a VSAM cluster is archived and recataloged to the CA-Disk pseudo-volume, alias entries are also defined for each of the alternate index and path names, as well as for each of the components' names for the cluster. This enables the auto-restore function to be invoked when you reference the base cluster, any of its components, an alternate index, or path name associated with the base cluster.

Restore processing removes the catalog entry to the pseudo-volume and all of the aliases associated with it, and redefines the correct VSAM catalog entries. Should the restore process fail, CA-Disk attempts to re-create the initial status by recataloging the cluster to the pseudo-volume and associating the proper alias names to it.

## Restrictions

If the catalog management hook and DFHSM SVC hook are installed, the following restriction applies:

- A reference to an uncataloged archived data set in a batch job with volume and unit information hard coded in the JCL fails with a JCL error.

## Auto-Restore Delayed by Merge

Use SYSPARM MERRELTMnn, where DMSAR can be delayed by a Merge job. Specify Merge release time MERRELTM of 01 to 99 minutes (default is 00) to be used in both Merge and DMSAR. During Merge, if MERRELTM is non-zero and there is room on the current volume for the next Archvol, the time interval since the last output volume swap is compared to MERRELTM. If that time interval is greater than or equal to MERRELTM, a test ENQ for DMSWVOLS is issued for that volume. If DMSAR has been waiting with a DMSWVOLS ENQ for that volume, the volume is released by Merge. DMSAR only issues the DMSWVOLS ENQ if MERRELTM is non-zero; the actual value of MERRELTM is picked up in Merge. Tuning can be done within Merge with MERRELTM from 01 to 99 minutes, depending on individual needs.

## Propagating Enqueues

In multiple CPU environments with cross-system enqueue packages, you must ensure that all enqueues are propagated across all CPUs. For auto-restores in particular, enqueues for DMSAUT0, DMSAUT1, DMSAUT2, DMSAUT3, and DMSAUT4 must be propagated accurately. These enqueues ensure that data integrity is maintained.

---

*Note:* Failure to propagate these enqueues can lead to data loss problems. For example, jobs executing in a multi-system JES complex can corrupt a data set being auto-restored on another system in the same JES complex when the DMSAUTx enqueues are not propagated correctly.

---

## Additional Considerations

### Precautions in Changing Immediate to Deferred Requests:

Sysparm ARTAPEOK (with a value of D) and user exit ARESPREX (by setting the return code to H'1') both provide the ability to change an immediate restore request into a deferred request; that is, place the Auto-Restore request in the restore queue rather than process it immediately. The sysparm option automatically defers a request only if it is for a TSO user and a tape mount is required. Batch jobs and restores from disk archives are not affected. The user exit is more general, however, and can defer any of the various request types.

There is also a slight exposure that you should consider before using either of them. As you can see from the description below, the exposure is extremely low under most circumstances, especially with the sysparm option, but indiscriminate use of the user exit option can cause a problem.

The concern exists only if two jobs and/or TSO users happen to want the exact same archived data set at the "same" time. CA-Disk Auto-Restore processing for the second request detects that another Auto-Restore task (serving the first request) is already processing the same data set. The second job is then placed in a wait state (if it's batch), or informed that the restore is in progress, and to try again later (if it's a TSO user).

If the sysparm option is used and the first request is from TSO and the second one from batch, the batch job will fail because of its assumption that the Auto-Restore for the TSO user was actually restoring the data set, when in fact it just placed it in the queue.

If the user exit is used to force a request to be queued instead of restoring it immediately, the results for the second job of two simultaneous requests can be as just described, without regard to the TSO or batch status of either requestor!

To summarize, the exposure exists only if two jobs/users attempt to access the same archived data set at the same time. This probability is very low. If the sysparm option is used, the probability is further reduced by the fact that the first job must be a TSO request, and the second a batch job.

### Rejecting Auto-Restore Requests Via Exclusion Tables

The same user exit mentioned above, ARESPREX, can also be used to reject an Auto-Restore request. The exit has access to considerable data regarding the restore itself, and therefore provides a great deal of flexibility in the decisions that can be made, and consequently should be considered first if you have needs in this area.

Another method also exists, but it provides much more limited flexibility, and requires the assembly and linking of exclusion tables into a CA-Disk module. ADSAR003 is responsible for starting the DMSAR Started Task. This module is statically linked into ADSAR010, the catalog management hook. Before ADSAR003 even starts the restore task, it has the ability to screen the request and reject it, based upon either the dsname that is needed, the name of the executing program, or the jobname.

### PGMEXMPT Member

The programs and/or dsnames that are to be excluded are provided in several CSECTS statically linked into ADSAR010 and available to ADSAR003. These CSECTS define exclusion tables for programs, data sets, jobnames, and TSO USERIDs. These exclusion tables are created using member PGMEXMPT located in your DMSASM library.

If you have program names, data set names, or jobnames that can always be excluded from Auto-Restore, then add them to these exclusion tables. This is accomplished by defining the desired exclusion table in a user assembly process and linking these into the Auto-Restore hook, ADSAR010 and ADSHS001.

You can also exclude programs that use the functionality provided by the DFHSM SVC hook. This can be necessary for programs that do not process correctly if the true catalog volser is not returned, or if they attempt communicating with DFHSM through direct subsystem calls. The method of exclusion is similar to the method described above. Entries are made to a table (CSECT ADSAR006 in module ADSAR010) which identifies program names that should not be passed the volser name of MIGRAT. See Installation library member USERMODB for an example.

The following figure is a sample of the PGMEXMPT source:

FIGURE 16. Sample source for PGMEXMPT member

```

EDIT                                     .DMSASM(PGMEXMPT) - 01.01          Columns 00001 00072
Command ==>                               Scroll ==> CSR
***** ***** Top of Data *****
000001 PGMEXMPT TITLE 'AUTO-RESTORE EXEMPTION BY PROGRAM NAME TABLE'
000002 *****
000003          COMPILE ASEM=RENT,LKED=RENT                                *
000004 *                                                                    *
000005 * DESCRIPTION:                                                         *
000006 *   PGM EXCLUSION/EXEMPTION TABLE                                   *
000007 *                                                                    *
000008 * MAINTENANCE HISTORY                                                 *
000009 * -----*
000010 * 12/14/95 DLH NONE 133694 REMOVE OBJECT CROSS-REFERENCE DATA      *
000011 * 01/24/96 DLH NONE 133719 SUPPORT PATTERN MATCHING                *
000012 * *****
000013          SPACE 3
000014 PGMEXMPT SCRNLIST LENGTH=8,                                           X
000015          SCREEN=(USERPGM1,USERPGM2,.....USERPGM2, X
000016          ADSMI/)
000017 *
000018          END
***** ***** Bottom of Data *****
Aa  A TCPIP          R 21 C 31          8:27 11/25/97

```

Note: The LENGTH parameter within the PGMEXMPT source must be left untouched. This is the maximum length for each program name, and is set at 8.

After your exclusion lists have been defined, an assembly process can be applied with SMP/E in the form of a usermod. Installation library member USERMOD9 provides an example of this process.

For your convenience, the following figure is a sample of USERMOD9:

FIGURE 17. Sample JCL for USERMOD9

```

EDIT                                     .INSTALL(USERMOD9) - 01.00          Columns 00001 00072
Command ==>                               Scroll ==> CSR
-----
000001 //JOBNAME  JOB (ACCT INFO)
000002 //SMP      EXEC PGM=GIMSMP, REGION=5120K, PARM='CSI=SAMSSMPE.GLOBAL.CSI'
000003 //SYSPRINT DD SYSOUT=*
000004 //SMPRPT  DD SYSOUT=*
000005 //SMPOUT   DD SYSOUT=*
000006 //SMPHOLD DD DUMMY
000007 //USERASM  DD DISP=SHR, DSN=USERID.DISK.ASM
000008 //SMPCNTL  DD *
000009 SET BOUNDARY(GLOBAL).
000010 RECEIVE S(SDU1819).
000011 SET BOUNDARY(DMSTGT).
000012 APPLY S(SDU1819) ASSEM .
000013 //SMPPTFIN DD *
000014 ++USERMOD(SDU1819).
000015 ++VER(2038) FMID(SDM1900) .
000016 ++SRC(PGMEXMPT) DISTLIB(ADMSASM) DISTMOD(ADMSLOAD) TXLIB(USERASM).
000017 ++SRC(DSNEXMPT) DISTLIB(ADMSASM) DISTMOD(ADMSLOAD) TXLIB(USERASM).
000018 ++SRC(JOBEXMPT) DISTLIB(ADMSASM) DISTMOD(ADMSLOAD) TXLIB(USERASM).
000019 ++SRC(TSOEXMPT) DISTLIB(ADMSASM) DISTMOD(ADMSLOAD) TXLIB(USERASM).
000020 /*
Aa  A TCPIP          R 4 C 15          8:17 11/25/97

```

## Installation of IGGDASU2 User Exit

This user exit is provided for those who want to make use of SMS GDG's and do not have HSM. A problem can be encountered where LOGREC records are being cut when a SMS GDG rolls out of the sphere and is cataloged to the CA-Disk pseudo volser ARCIVE. The problem is with scratch processing for a volume that does not exist. The User exit, IGGDASU2, is provided by IBM to keep logrec records from being cut.

If you plan to use a CA-Disk pseudo-volume name other than ARCIVE, it must be specified through sysparm RECATVOL. Do not select a pseudo-volume name that is a real volume name, or likely to become one. If your pseudo-volume name is a real volume, you cannot reference your data sets on that volume through the catalog. Set the pseudo-volume name as the value for the sysparm RECATVOL. For more information, see the sysparm description for "RECATVOLvvvvv" on page 166 in the Systems Guide.

ADSDASU2 is the user exit provided by CA-Disk. This module gets control during DADSM scratch processing. It looks at the parameter list passed and if the volser is ARCIVE, the request is ignored, indicating to the caller that processing was successfully completed.

If it is determined that this exit is needed, it should be installed as part of the automatic processing during IPL. ADSDASIN is the installation program for this user exit. The EXEC statement should specify PARM=I to

dynamically install the exit. PARM=R can be used to remove the exit for testing purposes or in the event that any problems are encountered.

The following is sample JCL to install the user exit. A copy of this JCL is found in the INSTALL library member ADSDASIN.

FIGURE 18. Sample source for ADSDASIN Install Member

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT                               .INSTALL(ADSDASIN) 01.07      Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 //JOBNAME  JOB (ACCT INFO)
000002 //*
000003 //*****
000004 //*
000005 //*                INSTALL ADSDASU2 SCRATCH EXIT
000006 //*
000007 //*****
000008 //STEP1  EXEC PGM=ADSDASIN,PARM=I      (USE PARM=R TO REMOVE EXIT)
000009 //STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOADLIB
000010 //PAMRLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
000011 //MSGPRINT DD SYSOUT=*
000012 //SYSUDUMP DD SYSOUT=*
***** ***** Bottom of Data *****

```

Aa A TCPIP R 4 C 15 8:32 11/25/97

## Customizing the CA-Disk Tape Management Support

Tapes have traditionally been managed by the expiration dates written in the tape labels. Standard IBM support for date-protected data sets requires operator intervention to rewrite a protected tape before its expiration date.

*Note:* For related information, see [“Year 2000 Considerations”](#) on page 6 in the User’s Guide.

Tape management systems usually extend this support in several ways. They interpret certain expiration dates as codes to indicate the type of control governing the use of the tape. A master control file, or “tape management catalog,” records this information, and controls access to each tape. If the control file indicates a tape as an

available scratch tape, it allows it to be rewritten without operator intervention, regardless of the expiration date in the label.

For example, CA1, Computer Associates uses an EXPDT=99000 to mean that a tape is eligible as a scratch tape when it becomes uncataloged. As long as the tape is cataloged, it is protected from being used as a scratch tape. This is commonly known as “catalog control.” An EXPDT=99365 is often used as the code to mean “permanently protected” -- the tape cannot be reused as a scratch tape unless the control file is updated to change this status. If you are using LDATE/ddd (to retain ddd days after the tape was last used) be aware that CA1 assigns an expiration date of 98ddd which might be confusing since it shows as 1998ddd in some of our reports. Consult the documentation for your tape management system to find the various means of control it provides through expiration date codes.

With the introduction of MVS/XA 2.2 (DFP 2.3.0), IBM also recognizes 99365 as a “never expire” date. This means that for standard applications, operator intervention are required to rewrite such tapes even in year 2000 and beyond. Beginning with DFP 1.1, however, IBM has also provided an exit to allow operator intervention to be avoided. CA-Disk optionally makes use of this exit, based upon the value of sysparm TAPEFSCR.

CA-Disk uses standard techniques to open and write tape data sets, which causes tape labels and their expiration dates to be created by normal means. As distributed, CA-Disk dynamically allocates all tapes that are needed and assigns them an expiration date of Julian 99365 (from the default value of sysparm DYNEXPDT). This value guarantees the integrity of the data by ensuring that a tape never expires before the data that it contains. CA-Disk determines when all of the data on each tape has expired, and at that point returns it as an eligible scratch tape.

CA-Disk also provides an option to catalog each tape data set that it creates. By specifying expiration dates and catalog options correctly, CA-Disk is fully compatible with all major tape management systems. As mentioned above, sysparm DYNEXPDT controls the expiration date for dynamically allocated tapes. If you supply DD statements for the output tapes, the expiration date you supply in the JCL is used.

A description of several implementation options relating to tape management in general is presented in the topic [“Tapepool Considerations” on page 45 in the Systems Guide](#). CA-Disk provides three methods for controlling tapes. Select one of these methods and follow the instructions for installing the tape management support.

## Method 1 — Controlling Tapes Via the EDM

Beginning with Release 4.8 of CA1, a facility was provided (External Data Manager Interface) to allow other system software products to manage their own tapes. CA-Disk is one such system. It is through this interface that the greatest amount of control with the least amount of risk is available. Refer to the CA1 manuals for a complete description of the External Data Manager Interface. The advantages of using this method to control CA-Disk-created tapes over any other methods are as follows:

1. Activation of the External Data Manager (EDM) interface for CA-Disk is greatly simplified.
2. Changes within CA1 or CA-Disk do not require the reinstallation of the interface.
3. A single point of control is established to determine when a tape should be scratched.
4. CA1 does not attempt to prematurely scratch a CA-Disk- owned tape.
5. CA1 does not allow other programs to overwrite a CA-Disk- owned tape.
6. If CA-Disk creates a tape and then later abends, the tape are not scratched, as it normally would without the EDM facility.
7. Future changes in CA-Disk or CA1 do not increase the likelihood of tapes being scratched prematurely.

When CA-Disk is identified as an EDM to CA1, any archive/backup tapes created by CA-Disk will be managed by CA-Disk (that is, CA-Disk will inform CA1 when to scratch the tape). CA1 exempts these tapes from its normal processing. Any tapes created through sequential migrate--or UNLOAD to tape--processing is not considered externally managed, however, and are scratched by CA1 based on the tape's expiration date.

A tape can be identified as externally managed by its expiration date (99365) and an indicator flag within the volume's TMC record. The actual expiration date of the tape is kept in the ARCHVOLS record within the files data set. Normally this is 1999365, if the default value of sysparm DYNEXPDT is used. If this is so, the date is scratched when the last data set on the tape expires. When CA-Disk determines that all data on the tape has expired, it expires the tape through the EDM interface. Directly changing the expiration date for a tape must be done through the appropriate CA-Disk commands, not through CA1. Only when a tape is expired by CA-Disk should the status within the TMC change.

CA1 releases 4.8 and 4.9 allow for only one EDM. If you are unable to identify CA-Disk as an EDM because you already have another system identified, consider using method 2 described below. The default expiration date for all tapes should be used, however, since there are several limitations in using the direct interface recommended with method 2.

The activation of EDM support is done in two parts: First, sysparm TMSCTLEX must be set to indicate to use the CA-Disk EDM program as the tape management interface. Set this sysparm to ADSTH014. Second, CA-Disk must be identified as the EDM within CA1. This is accomplished by modifying the TMOEDMxx member of your CA1.PPOPTION data set. The easiest method of identifying CA-Disk as an EDM is by program name. For example:

```
EDM=SAMS , PGM=ADSMI002
```

---

*Note:* Refer to your CA1 manuals for more information on the External Data Manager Interface.

---

If you have previously created archive/backup tapes with CA-Disk, you also need to change their status within the TMC to indicate that they are also externally managed. This action is required because CA-Disk attempts to scratch the tapes through the EDM interface as soon as the sysparm TMSCTLEX is set.

The conversion of existing tapes to EDM control can be done using documented CA1 utilities. The volume record for each tape must have the following fields set:

**TABLE 14. EDM Control Volume Records**

Volume Record	Value
1STVOL	hex zeros
NEXTVOL	hex zeros
PREVVOL	hex zeros
FLAG3	hex '20'
VOLSEQ	hex '0001'
EXPDT	99365

---

*Note:* If the conversion of existing tapes is not done properly, CA1 begins issuing TMSTVEXT-08 messages. Circumvent this problem by executing the TMSUPDTE Utility, changing flag '3' to '20'. For more information about this utility, refer to your CA1 manuals.

---

More detailed procedures can be obtained from CA1 Technical Support for the release of CA1 you are running. More general procedures can be obtained from CA-Disk Technical Support Center.

Other CA-Disk sysparms which control the data set name, expiration date, and catalog action of CA-Disk tapes can be set as described below. The expiration date on the internal label of an externally managed tape is not affected by the EDM facility.

## Method 2 — Controlling Tapes by Expiration Date

As distributed, CA-Disk defaults to assigning expiration dates of 99365 to all tapes, but does not catalog them. This technique is intended for those installations that do not have a tape management system, and as an option for those using earlier releases of CA1. When CA-Disk determines that all data on the tape has expired, it is released (through a direct interface) and made eligible as a scratch tape.

---

*Note:* If CA1 release 5.0 or above is installed, do not use the CA-Disk/CA1 direct interface. CA1 macro changes have made this interface incompatible.

---

Whether or not you also catalog the tapes that are actually under expiration date control is a choice you should make, based upon whether it provides you with any additional benefit. Neither CA-Disk nor CA1 makes use of the catalog status, but it does allow for a simple list of the catalog entries as a means to find the CA-Disk tapes.

To have CA-Disk catalog the tapes it creates, do not specify `DISP=(NEW,CATLG)` in the JCL. Instead, specify sysparm `ARCTNAME` with an appended value of `C`. CA-Disk generates a unique name for each tape it creates, and then catalog it. You should also specify sysparm `UNCATARC` with a value of `Y`, to have CA-Disk help keep your system catalog clean by uncataloging each tape when it returns it to the scratch pool. Review the use of sysparm `TAPEFSCR` with a value of `Y`, as well. CA1 users should not specify it, but it is recommended for all other users.

For expiration date control within a CA1 (Release 4.7 and below) environment, we also recommend that you install the direct interface that CA-Disk additionally provides. See the information for sysparm [“TMSCTLEXmmmmmmmm” on page 184 in the Systems Guide](#) and user exit [“TMSCTLEX — Tape Management System Control Interface Exit” on page 249 in the Systems Guide](#), as it pertains to the direct interface. Then decide if the direct interface to CA1 is applicable in your installation. If it is not, change the expiration date being assigned via sysparm `DYNEXPDT`, described in [“DYNEXPDTE99365” on page 143 in the Systems Guide](#), or your JCL to specify either a true expiration date or the value `99000`, which can then be used to place the tapes under catalog control as described in Method 2 above. See [“Assigning Tape Expiration Dates” on page 45 in the Systems Guide](#) for other applicable rules.

## Method 3 — Controlling Tapes by Catalog Status

Consider this technique if your tape management system supports an option to designate tapes as in-use as long as they are cataloged, and as available scratch tapes when they are uncataloged. This technique is appropriate for either TLMS or CA1, both from Computer Associates. When CA-Disk determines that all data on a tape has expired, CA-Disk releases the tape and makes it eligible to be a scratch tape by uncataloging it.

To implement catalog control, you must use the expiration date that your tape management system defines for that purpose. For tapes dynamically allocated by CA-Disk, you must specify this value in sysparm `DYNEXPDT`. If at some point you decide to override dynamic allocation by providing JCL for the output tapes, you must provide this value in the `LABEL=EXPDT=yyddd` parameter for the tape DD statements.

To have CA-Disk catalog the tapes it creates, do not specify `DISP=(NEW,CATLG)` in the JCL. Instead, specify sysparm `ARCTNAME` with an appended value of `C`. CA-Disk generates a unique name for each tape it creates,

and then catalog it. You should also specify sysparm UNCATARC with a value of Y, to have CA-Disk help keep your system catalog clean by uncataloging each tape when it returns it to the scratch pool.

**Warning!** In the event you lose the catalog that your CA-Disk tapes are cataloged in, your tape management system can scratch all CA-Disk tapes. Take appropriate steps to prevent this from occurring.

## Method 4 — Controlling Tapes Using TLMS/EDM

This support provides a module for the TMSCTLEX user exit that interfaces with the EDM developed for release 5.0.4 of TLMS. To enable the support, sysparms TMSCTLEX and DYNEXPDT must be set to the values ADSTH017 and E99365 respectively. The TLMS documentation supplied by CA must also be performed.

For detailed information regarding this support, refer to the [“ADSTH017 — Interface for TLMS”](#) on page 252 in the [Systems Guide](#).

## CA-Disk/CA1 Interfacing Considerations

We recommend that you run the DSNDELETE command of IXMAINT (documented in [“DSNDELETE Command and Parameters”](#) on page 322 in the [User’s Guide](#)) nightly before the CA1 scratch and clean functions are run. This makes the expired CA-Disk tapes available to CA1 as scratch tapes.

For users who do not use the EDM interface, if a job abends while writing a tape, CA1 defaults to considering that output tape a scratch. To CA-Disk, however, the partial tape is a good tape and must be kept. Take special precautions to prevent CA1 from marking CA-Disk output tapes as scratch tapes after an abend. This can be done manually, or you can make use of one of the exits within the CA1 system. A CA-Disk user has supplied a sample for this exit in member SLI035 of the user mod library. Consult your local CA1 support staff or Computer Associates directly, if necessary, if you have further questions regarding their exit.

## Capabilities of the ISPF Interface

### Limiting ISPF Users' Access to Data Sets

Specify sysparm SPFUSRID, described in [“SPFUSRIDn”](#) on page 180 in the [Systems Guide](#), with a value of Y if ISPF functions are to allow users to process only those data sets with names prefixed with their TSO user ID. This feature can be overridden for selected users by specifying their user IDs in the TSouserI member of parmlib, described in [“TSO/ISPF USERID Authorization List”](#) on page 492 in the [Systems Guide](#).

### ISPF Function-Generated JCL

CA-Disk generates JCL to execute various functions in the background environment. All generated JCL executes standard CA-Disk JCL procedures that are distributed with the system. It is assumed that these procedures are installed in an accessible procedure library and are compatible with those originally shipped with the system.

## Defining ISPF and DSCL User Options

CA-Disk dynamically builds its own menus of available ISPF and DSCL functions. This is done to provide a flexible way for CA-Disk installation coordinators to control the use of CA-Disk in their installation.

When a user enters the CA-Disk option on the ISPF primary option menu, CA-Disk determines the user ID of the user and goes to an authorization list stored as a member in the CA-Disk parmlib data set. This authorization list

indicates which users can use CA-Disk and what functions of the support they are authorized to use. Based on this information, a menu is presented to the user indicating only those functions that are authorized for their use.

At this point of the ISPF customization, members SPFOPTNS and DMCOPTNS should be created in the parmlib data set by copying in members SAMPSPFO and SAMPDMCO respectively, and then reviewing and modifying as needed.

SPFOPTNS and DMCOPTNS parmlib members contains three fields of information that are coded free-form on each list entry. As many entries as needed can be included in the list. The three fields are: TSO USERID, INCLUDE/EXCLUDE Specification and Function Specification.

### Field 1 — TSO USERID

This field is a TSO USERID or prefix (partial user ID ending with a slash). This field must start in the first position of the entry after the starting single quote. The slash (/) can be used to indicate groups of users to which the entry applies. A single slash indicates that the entry applies to all users. This field must be followed by one or more spaces.

### Field 2 — INCLUDE/EXCLUDE Specification

This field is a one-character field indicating the specification type: “I” for include, “E” for exclude. It must follow the user ID field and be separated before and after with at least one space.

Entries are specified to allow users to process certain options (include) or to disallow users from using options (exclude).

This can be useful by specifying an include statement for a user indicating that all options are allowed. Then by specifying an exclude statement for one option, the user is allowed access to all options except one. This prevents the need to specify every function that a user can use.

Any number of statements can be specified for a user or group of users. One or more include statements must be specified to allow a user access to a function. Any exclude statement for a function overrides an include statement for it; that is, an exclude statement applied to all user IDs will deny access to the specified function regardless of any other include statements.

### Field 3 — Function Specification

This field is a list of CA-Disk functions that a user or user group is authorized to use. Function codes are seven characters and can be abbreviated to the first three characters. Functions must be separated by commas (not spaces). Up to 150 characters of functions can be specified using the continuation rules of CA-Disk parmlib entries. The last function code is followed by a single quote.

Following is a list of the valid specifiable functions:

**TABLE 15. ISPF Valid Function Specifications**

CODE	ISPF FUNCTION	DESCRIPTION
ALL	SPECIAL FUNCTION	Indicates that all functions are to be assumed for this specification. In specifying the ALL function, all features are assumed. For any function that has not been licensed, subsequent use of the ISPF option to invoke it can result in abends looking for routines you do not have. If applicable, this value should be specified alone in field 3 of the statement.
BARCHIV	ARCHIVE/BACKUP -SUBMIT BATCH JOB TO ARCHIVE DATA SET	Generate JCL for batch job submission.

TABLE 15. ISPF Valid Function Specifications

CODE	ISPF FUNCTION	DESCRIPTION
BDARCHI	ARCHIVE/BACKUP -QUEUE A REQUEST (VIA BATCH)	Generate JCL for batch job submission.
FDARCHI	ARCHIVE/BACKUP - QUEUE A REQUEST ONLINE	ISPF foreground function — allow user to enter a deferred archive command.
BRESTOR	RESTORE - SUBMIT BATCH JOB TO RESTORE DATA SETS	Generate JCL for batch job submission.
BDRESTO	RESTORE - QUEUE A REQUEST (VIA BATCH)	Generate JCL for batch job submission.
FDRESTO	RESTORE - QUEUE A REQUEST ONLINE	ISPF foreground function — allow user to enter a deferred restore command.
FRESTOR	RESTORE - EXECUTE RESTORE REQUEST (VIA TSO)	Invoke TSO RESTORE command
LDSINDX	LIST - ARCHIVE/BACKUP INDEX ENTRIES	ISPF foreground function — allow user to list entries only.
DDSINDX	LIST - (OR DELETE) ARCHIVE/BACKUP INDEX ENTRIES	ISPF foreground function — allow user to list and delete entries for archived data sets.
CDSINDX	LIST - (OR CHANGE) ARCHIVE/BACKUP INDEX ENTRIES	ISPF foreground function — allow user to list and change expiration date values for archived data sets.
ADSINDX	LIST - (DELETE OR CHANGE) ARCHIVE/BACKUP INDEX ENTRIES	ISPF foreground function — allow user to list change expiration dates and delete entries for archived data sets. (If the delete function is requested for an index record associated with a RACF-protected data set for which a discrete profile has been saved, authorization of the ISPF session is required. This can be accomplished if you have a user SVC to obtain authorization, and indicate it to CA-Disk via sysparm RADELSVC. If this cannot be provided, give your users the CDSINDX function instead of this one.)
LQUEUED	LIST - QUEUED ARCHIVE/RESTORE REQUESTS	ISPF foreground function — allow user to list deferred archive and restore requests previously entered.
DQUEUED	LIST - (OR DELETE) QUEUED ARCHIVE/RESTORE REQUESTS	ISPF foreground function — allow user to list and delete deferred archive and restore requests previously entered.
EDSKDSK	MIGRATE - MOVE A SPECIFIC DATA SET TO A NEW VOLUME	Generate JCL to invoke the Move/Copy function using the COPY command
EPDSCOM	COMPRESS - COMPRESSA SPECIFIC PDS	Generate JCL to invoke the PDS Compress function for specific data sets.
ERELOAD	COMPRESS - RESTART COMPRESS OF SPECIFIC PDS	Generate JCL to invoke the PDS Compress RELOAD function.
IREPORT	REPORTS - SCAN VTOCS: GENERATE FIXED FORMAT REPORTS	Generate JCL to invoke the REPORT function.
IARCHIV	ARCHIVE/BACKUP - SCAN VTOCS TO SELECT DATA SETS	Generate JCL to invoke the implicit archive (RETAIN) function.
FVRPT	REPORTS - FIND SPECIFIC VSAM CLUSTER OR GROUP	Generate JCL to invoke the explicit VSAM (FIND) function to generate VSAM reports only.
FVSAM	ARCHIVE/BACKUP -FIND SPECIFIC VSAM CLUSTER OR GROUP	Generate JCL to invoke the explicit VSAM (FIND) function.

**TABLE 15. ISPF Valid Function Specifications**

<b>CODE</b>	<b>ISPF FUNCTION</b>	<b>DESCRIPTION</b>
IVRPT	REPORTS - SCAN CATALOGS TO SELECT VSAM CLUSTERS	Generate JCL to invoke the implicit VSAM function to generate VSAM reports only.
IVSAM	ARCHIVE/BACKUP - SCAN CATALOGS TO SELECT VSAM CLUSTERS	Generate JCL to invoke the implicit VSAM function.
IRCOVER	IMPLICIT RECOVERY - RESTORE DATA SETS FROM A VOLUME	Generate JCL to invoke the implicit recovery function.
IXMAINT	MAINTENANCE - DELETE EXPIRED ENTRIES IN ARCHIVE INDEX	Generate JCL to invoke the archive index maintenance procedure (IXMAINT)
VOLDISP	LIST - ARCHIVE VOLUME INFORMATION	Display information about CA-Disk archive volumes in the foreground ISPF environment.
TAPEPOO	MAINTENANCE - UPDATE ARCHIVE TAPE POOLS ONLINE	Invoke the interactive CA-Disk tapepool management functions.
RELEASE	RELEASE - SCAN VTOCS: RELEASE IDLE SPACE	Generate JCL to invoke the idle space release function.
IDSKDSK	MIGRATE - SCAN VTOCS: MOVE DATA SETS TO NEW VOLUMES	Generate JCL to invoke the implicit Move function.
IPDSCOM	COMPRESS - SCAN VTOCS: COMPRESS ANY PDS IF ITS NEEDED	Generate JCL to invoke the implicit PDS Compress function.
IPRELOA	COMPRESS - SCAN VTOCS: RESTART AFTER AN ERROR	Generate JCL to restart the PDS Compress RELOAD function.
ISEQMIG	MIGRATE - SCAN VTOCS: MOVE PS DATA SETS TO TAPE	Generate JCL to invoke the implicit sequential migration to tape function.
ALTDISP	MISCELLANEOUS - DISPLAY RELEASE, FILES, AND PARMLIB	When this option is selected it will display a panel that shows the names of the CA-Disk data sets that will be used for CA-Disk requests.
SYNCHEK	MISCELLANEOUS -SYNTAX CHECK A MEMBER OF PARMLIB	This utility function causes CA-Disk to check the syntax for any member of the CA-Disk parmlib data set. This is normally used after a user has modified some member in parmlib.
ONLIRPT	REPORTS - DEFINE YOUR OWN EXECUTE ONLINE (OPTIONAL)	Generate online reports using the DSCL selection language. After defining a report online it can be produced either online or via a batch job.
FILDUMP	MISCELLANEOUS - DISPLAY FILES DATA SET RECORD IN HEX	Generate online hexadecimal dump of subfile records in the files data set.
DSCL	GENERATE DSCL COMMANDS	Generate JCL for batch job submission.
ONLIRPT	REPORT USING SPFRPTS	Generate a REPORT command with the SPFRPTS parameter.
IREPORT	REPORT USING DSCL	Generate a REPORT command.
VBACKUP	BACKUP A PHYSICAL VOLUME	Generate a VBACKUP command.

Since there are more options available than menu lines on a screen, it is sometimes necessary for CA-Disk to implement two levels of menus. This occurs when more than 15 options are given to any user. If a user has fewer than 15 options, all options will be displayed on one selection menu. If more than 15 options are specified, the first menu will display a list of functional categories from which the user must select. Once this has been specified, a second menu will be displayed with the appropriate functions listed.

When specifying user options, the CA-Disk installation coordinator consider carefully what options to give each user. Several of the options are similar to each other, and are supplied to give the installation flexibility in distributing functions. Every user does not need every function available. The distributed sample members SAMPSPFO and SAMPDMCO attempts to eliminate this duplication of function, but users have different requirements and need to have different options specified.

In addition, remove any options to which you do not want your users to have access. If your installation has not implemented the CA-Disk archive/backup tapepool facility, it should also be removed from the menu display.

The sample options members SAMPSPFO and SAMPDMCO supplied with your system are already set up with entries that can be further tailored for your installation. If you have not already done so, copy them in to create members SPFOPTNS and DMCOPTNS. Two sample entries exist in the member SAMPSPFO. The first entry specifies the ISPF functions that are normally given to all CA-Disk users. The second entry specifies the ISPF functions that are to be given to only those persons responsible for DASD management.

The two entries are:

FIGURE 19. Sample source for SAMPSPFO

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT                      .PARMLIB(SAMPSPFO) - 01.01          Columns 00001 00072
Command ==>                Scroll ==> CSR
***** ***** Top of Data *****
000001 * ----- THIS IS A SAMPLE TO BE COPIED INTO MEMBER SPFOPTNS -----
000002 * ----- IT SHOULD THEN BE TAILORED TO MEET YOUR OWN NEEDS -----
000003 * FOLLOWING IS AN ENTRY THAT GIVES EXPLICIT PROCESSING FUNCTIONS
000004 * TO ALL USERS.
000005 *
000006 * '/ I  FDARCHI, FDRESTO, ADSINDX, DQUEUED, EDSKDSK, EPDSCOM, ERELOAD '
000007 *
000008 * IN ADDITION TO THE FUNCTIONS GIVEN ABOVE, THE DASD MANAGER IS
000009 * GIVEN THE FOLLOWING ADDITIONAL FUNCTIONS.  THIS ENTRY IS CURRENTLY
000010 * SET UP TO PERTAIN TO ALL USERS.  TO RESTRICT IT TO JUST THE DASD
000011 * MANAGER, PLACE THE DASD MANAGERS USERID IN THE PLACE OF THE '/'
000012 * IN THE ENTRY.
000013 *
000014 * '/ I  IRE, IAR, IUR, IUS, IRC, IXM, TAP, REL, IDS, IPD, IPR, ISE, ALT, SYN, VOL, ONL '
000015 *
000016 *
000017 * THE FOLLOWING LINES ARE COMMENTS ONLY.  THEY PROVIDE A LIST
000018 * OF THE AVAILABLE SAMS:DISK FUNCTIONS AND THE CODES THAT
000019 * CORRESPOND TO THEM.  SPECIFY 3 TO 7 CHARACTERS OF THE CODE
-----
Aa  A TCPIP          R 4  C 15          9:44  11/25/97
  
```

### ISPF Return Function Restriction

The CA-Disk ISPF function does not fully support the ISPF RETURN or JUMP facility. Users can use the facility to go from CA-Disk panels to non-CA-Disk panels, but cannot specify multiple menu level numbers when entering CA-Disk functions. For example, “=8” and the <RETURN> key can be used to get to the CA-Disk ISPF menu, but “=8.4” cannot be used to get to a specific CA-Disk ISPF function. However, the ISPF command delimiter character can be used in place of the period. When the command delimiter character is a “;”, entering

```
=8;4;2
```

selects the fourth option of the condensed menu, then the second option from the selection menu.

---

## Customizing the CA-Disk TSO Support

### Tape Units Allocated Concurrently

By default, only one tape unit (one restore) is allowed to be allocated by dynamic restore at any given time. Sysparm TSOTULMT can be specified with a numeric value to permit two or more concurrent allocations (restores).

```
TSOTULMT05          <=== tape unit limit set to five
```

### Tape Unit Name

The default unit name for allocating a tape drive is “TAPE”. If this unit name does not cover the devices, use the following sysparm to provide the correct unit name.

```
TSOTUNITnnnnnnnn   <=== where "nnnnnnnn" is the correct name
```

### Limiting TSO User Access

The DARCHIVE and DRESTORE commands allow TSO users to archive and restore any data set. To restrict most users to archive and restore only those data sets prefixed with their user IDs, sysparm TSOURID must be specified with a value of Y. Access to all data sets can then be given to selected users by placing their user IDs in member TSOUSERI of the parmlib data set.

To allow DARCHIVE requests to have access against offline volumes, sysparm TSOVOLOF must be specified with a value of Y. The default (N) enforces that the data set requested must be found online or the request is rejected. Update parmlib member SYSPARMS with entries TSOURIDY and TSOVOLOFY, and create member TSOUSERI with the proper user IDs to accomplish both of these.

## TSO Screening Exits for Archive and Restore

Exits are also available that allow an installation to screen archive and restore requests. These exits are documented under “RESCRNEXmmmmmmmm” on page 167 in the [Systems Guide](#) and can be referenced there if more elaborate restrictions are needed.

Both user ID restrictions (sysparm TSOUSRID = Y) and the TSO user screening exits can be used at the same time. The user ID screening check is made first. The user exit module never sees the request if the ID requirement is not met.

## TSO Command and ISPF/PDF Task Library Override - TASKLIB

USERMODD determines how modules are loaded for the TSO commands and the ISPF/PDF interface. Modules are loaded from //DSMOSLIB-DD by default. The DDNAME can be changed by altering the value of the DDNAME=keyword. If the value is not supplied, modules are loaded from the current //JOB LIB-DD, //STELIB-DD and link-list library (normal search).

FIGURE 20. Sample USERMOD used to modify the DDNAME

```

Tn3270 - EXTRA! for Windows 98 / Windows NT
File Edit View Tools Session Options Help
Menu Utilities Compilers Help

BROWSE .BASE.ASM(TASKLIB) - 01.00 Line 00000000
Command ==> Scro
***** Top of Data *****
TASKLIB TITLE 'SAMS:Disk Task library data definition for TSO/ISPF
*****
COMPILE ASEM=RENT,LKED=RENT
*****
* DESCRIPTION:
* This is a sample usermod used to modify the SAMS:Disk DDNAME
* for programs running in the TSO/ISPF environment.
* Read the TASKLIB-macro prolog for specifications of options
* you may override.
* *****
TASKLIB TASKLIB DDNAME=DMSOSLIB SAMS:Disk load library DDNAM
* - Nullify for normal search.
END
***** Bottom of Data *****
    
```

---

This section presents an introductory approach to evaluating CA-Disk, so that you can become familiar with the basic functions, commands and coding conventions. After these are well in hand, you can explore the full capabilities and multitude of options in each function to come up with an innovative and effective implementation for your specific site. The section [“Examples” on page 493 in the User’s Guide](#), coupled with the functional explanations, provides you with further ideas and assistance. If your installation has installed the CA-Disk ISPF panels, you can also use the menus and HELP text for self-training and evaluation.

As you read these examples, refer to [“Basic System — DMS PROC” on page 33 in the User’s Guide](#). It help you in understanding these examples more thoroughly and provide additional ideas in customizing them to better suit your needs.

---

## *Testing the CA-Disk System*

Start by running some sample reports.

## Step 1. Run Some CA-Disk Reports

For each volume found online, the following JCL produces summary reports that include such information as volume free space, types of data sets, and data set size distributions. Since these are summary reports, each volume produces only one report line. 24 volumes fits nicely on one page.

If your configuration has a large number of DASD devices, simply limit the report to a selected few volumes by adjusting the SELECT command to the input stream.

FIGURE 21. Sample DSCL Report Commands

```

EDIT (STEP1) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOB CARD JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X,
000200 // MSGCLASS=T,
000300 // NOTIFY=&USERID
000400 // JCLLIB ORDER=WORKSHOP.R90.PROCLIB
000500 //REPORTS EXEC DMS, P='WORKSHOP.R90.INSTALL',
000600 // Q='WORKSHOP.R90'
000700 //STEPLIB DD DISP=SHR,
000800 // DSN=WORKSHOP.R90.LOADLIB
000900 //FILES DD DUMMY
001000 //MSGPRINT DD SYSOUT=*
001100 //PARMLIB DD DISP=SHR,
001200 // DSN=WORKSHOP.R90.PARMLIB
001300 //SYSPRINT DD SYSOUT=*
001400 //SYSIN DD *
001500 SCAN REALVOLS
001600 SELECT VOL=(PACK0/,PACK1/)
001700 VREPORT ALLOCS,ATTRBS,DISTR
001800 /*
***** ***** Bottom of Data *****
Aa A TCPIP R 4 C 15 10:16 11/25/97

```

*Note:* Most parameters can be abbreviated to the first three characters. Also, a CA-Disk command (for example, SELECT) can start in any column, including column one. If the volume list contains only one entry, no parentheses are needed.

To produce a report of all the data sets on volume "MVSRES" and any volume that starts with "TSO", use either of the following:

FIGURE 22. Example of Reporting on ALL data sets on a Volume

```

File Edit Transfer Options Connection Macro Window Help
-----
EDIT                               (STEP2) - 01.00                Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOB CARD JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X, MSGCLASS=T, NOTIFY=&USERID
000200 // JCL LIB ORDER=WORKSHOP.R90.PROCLIB
000300 //REPORTS EXEC DMS,P='WORKSHOP.R90.INSTALL',
000400 // Q='WORKSHOP.R90'
000500 //STEPLIB DD DISP=SHR,DSN=WORKSHOP.R90.LOADLIB
000600 //FILES DD DUMMY
000700 //MSGPRINT DD SYSOUT=*
000800 //PARMLIB DD DISP=SHR,DSN=WORKSHOP.R90.PARMLIB
000900 //SYSPRINT DD SYSOUT=*
001000 //SYSIN DD *
001100 SCAN REALVOLS
001200 SELECT VOL=(MVSRES, TSO/)
001300 REPORT SVD
001400 *
001500 SCAN REALVOLS
001600 SELECT VOL=(MVSRES, TSO/)
001700 REPORT MVD
001800 /*
***** ***** Bottom of Data *****
Aa A TCPIP R 4 C 15 10:24 11/25/97

```

The SVD report produces an alphabetic listing of data set names on each volume. That is, each volume's data sets are listed alphabetically, a new report for each volume. The MVD report produces an alphabetic listing of data set names on all the volumes. All the data set names are listed alphabetically within a single report.

## Step 2. Archive Some data sets

The second function of CA-Disk evaluated is explicit archival. Archiving a data set consists of two phases. First, the data is read and stored on a tape or disk that is indexed by CA-Disk. Second, disposition for the archived data set is taken. Disposition consists of scratching and uncataloging or recataloging.

The following is an example of archiving a cataloged data set. The data set is scratched and uncataloged as the default disposition. Below that is an example of modifying the disposition action to no-scratch and no-catalog.

FIGURE 23. Sample DSCL ARCHIVE Command

```

EDIT                               (STEP2) - 01.00                Columns 00001 00072
Command ==> _____ Scroll ==> CSR
-----
***** ***** Top of Data *****
000100 //JOB CARD  JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X, MSGCLASS=T, NOTIFY=&USERID
000200 // JCLLIB  ORDER=WORKSHOP.R90.PROCLIB
000300 //REPORTS  EXEC DMS, P='WORKSHOP.R90.INSTALL',
000400 //          Q='WORKSHOP.R90'
000500 //STEPLIB  DD  DISP=SHR, DSN=WORKSHOP.R90.LOADLIB
000600 //FILES     DD  DUMMY
000700 //MSGPRINT  DD  SYSOUT=*
000800 //PARMLIB  DD  DISP=SHR, DSN=WORKSHOP.R90.PARMLIB
000900 //SYSPRINT  DD  SYSOUT=*
001000 //SYSIN    DD  *
001100 FIND DSN=A.B.C
001200 ARCHIVE
001300 *
001400 FIND DSN=A.B.C
001500 ARCHIVE DISP=KEEP
001600 /*
***** ***** Bottom of Data *****

```

Aa A TCPIP R 4 C 15 10:53 11/25/97

Note that a full explicit data set name or a data set name pattern can be specified.

### Step 3. Restore the data sets

Now that you have one or more data sets in the archives, try the RESTORE function with JCL similar to the following:

FIGURE 24. Sample RESTORE Command

```

EDIT (RESTORE) - 01.00 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOBNAME JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X,MSGCLASS=T,NOTIFY=&USERID
000200 // JCLLIB ORDER=WORKSHOP.R90.PROCLIB
000300 //RSTR EXEC RESTORE
000400 //FILES DD DISP=SHR,DSN=WORKSHOP.R90.FILES
000500 //MSGPRINT DD SYSOUT=*
000600 //PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
000700 // DD DISP=SHR,DSN=WORKSHOP.R90.PARMLIB
000800 //SYSPRINT DD SYSOUT=*
000900 //SYSIN DD *
001100 RESTORE DSN=A.B.C,
001110 VOL=PACK02,
001120 NOC
001200 *
001400 RESTORE DSN=A.B.C,
001410 NEW=TEMP.B.C
001500 /*
***** ***** Bottom of Data *****

```

If you want the data set restored to a volume other than the one from which it was archived, include the VOL= parameter to direct CA-Disk to a particular volume. Default disposition for a restore operation is to catalog the data set following the restore. If the original data set still exists on DASD, you cannot want CA-Disk to attempt to catalog the data set. Use the NOCATALOG (NOC) parameter to suppress catalog action.

The NEW= parameter causes CA-Disk to restore the original data set under a new name. This allows you to restore a data set to the same volume from which it was archived without a name conflict. The second example in Figure 24 illustrates this.

## Step 4. Submit a Backup Job

The data set backup function can be considered as an automated archival. The operation scans all the data sets on a volume or group of volumes and, based on the criteria you supply, archives certain data sets and retain others on the volume. This is a very powerful function and, as such, a SIMULATE mode of operation is provided. It is highly recommended until you are satisfied you obtain the results you want.

Specification of the retention criteria is very flexible and, as such, is also fairly complex. Refer to the information in “Basic System — DMS PROC” on page 33 in the User’s Guide, “BACKUP/ARCHIVE” on page 191 in the User’s Guide, and “PARMLIB” on page 474 in the Systems Guide when you set up your own DSCL selection criteria.

FIGURE 25. Example of DSCL DELETE and BACKUP Commands

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT                               (BACKUP) - 01.00                Columns 00001 00072
Command ==> _____                Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOB CARD JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X, MSGCLASS=T, NOTIFY=&USERID
000200 // JCLLIB ORDER=WORKSHOP.R90. PROCLIB
000300 //REPORTS EXEC DMS, P='WORKSHOP.R90. INSTALL',
000400 // Q='WORKSHOP.R90'
000500 //STEPLIB DD DISP=SHR, DSN=WORKSHOP.R90.LOADLIB
000600 //FILES DD DISP=SHR, DSN=WORKSHOP.R90.FILES
000700 //MSGPRINT DD SYSOUT=*
000800 //PARMLIB DD DISP=SHR, DSN=WORKSHOP.R90.PARMLIB
000900 //SYSPRINT DD SYSOUT=*
001000 //SYSIN DD *
001100 SET MODE=SIMULATE
001200 SCAN REALVOLS
001300 SELECT VOL=PACK01,
001400 CRITERIA=(TEMP, EQ, YES, AND, CREDIT, LE, TODAY-2)
001500 DELETE
001600 SELECT VOL=PACK01,
001700 CRITERIA=(MODIFIED, EQ, YES)
001800 BACKUP
001900 /*
-----
Aa A TCPIP R 4 C 15 11:12 11/25/97

```

The first selection criteria scratches all two-day-old temporary data sets. The second selection criteria backs up any changed VSAM and non-VSAM data sets.

## Step 5. Produce a LISTD/LISTV Report

Now you have even more data sets in the archives. If you want to determine what data sets are in the archives, you can run a LISTD, which displays the archive index. The archive index exists as two files, DSNINDEX and ARCHVOLS. The DSNINDEX is keyed on data set name and has one entry for each data set or copy of a data set that resides in the archives. The ARCHVOLS is keyed on volume serial and contains one entry for each archive tape volume. These files are listed as follows:

FIGURE 26. Sample LISTD/LISTV Commands

```

EDIT                                     (LISTD) - 01.34                               Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //JOBNAME  JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X, MSGCLASS=T, NOTIFY=&USERID
000200 // JCLLIB  ORDER=WORKSHOP.R90.PROCLIB
000300 //LIST     EXEC LISTD
000400 //PARMLIB  DD  DISP=SHR, DSN=SAMS.DISK.PARMLIB
000500 //          DD  DISP=SHR, DSN=WORKSHOP.R90.PARMLIB
000600 //SYSPRINT  DD  SYSOUT=*
000700 //FILES    DD  DISP=SHR, DSN=SAMS.DISK.FILES
000800 //SYSIN    DD  *
000900 LISTD (DSN=...)
001000 LISTV (VOL=...)
001100 /*
***** ***** Bottom of Data *****

```

Individual data sets or groups can be listed by adding the DSN= parameter to the LISTD, or individual volumes can be listed by adding the VOL= parameter. These can be used as an alternative to listing all of the entries.

## Step 6. Run a MERGE Job

Each time an archive or backup run is made, another archive tape is created (or an archive data set on disk, if you have implemented that option). As time passes, consolidate archive tapes (or move the archives on disk off to tape) and get rid of expired data sets.

CA-Disk provides an archive MERGE function that successively processes each archive tape volume, or an archive data set on disk, as input. It copies and distributes only the unexpired data sets according to their expiration dates and the limits you have specified. This allows you to produce archive tape volumes intelligently that expire some time in the future. You produce two new archive tape volumes: one that contains only those data

sets that expire in the coming 30 days, and the other, data sets that do NOT expire in the coming 30 days. This means that 30 days from the run, one tape is completely expired and is not be mounted for the next merge operation. Also, during the 30-day period, the second tape contains no expired data.

To perform the forward MERGE of archive data contained on tape volumes, execute a job similar to the following:

FIGURE 27. Sample MERGE Command

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT (MERGE) - 01.03 Columns 00001 00072
Command ==> Scroll ==> CSR
***** ***** Top of Data *****
000100 //XXXXXXXXX JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X,MSGCLASS=T,NOTIFY=&USERID
000200 // JCLLIB ORDER=WORKSHOP.R90.PROCLIB
000300 //MERGE EXEC MERGE,S='*',REGION=4096K,
000400 // P='WORKSHOP.R90.INSTALL',Q='WORKSHOP.R90'
000500 //UNLOAD.FILES DD DISP=SHR,DSN=SAMS.DISK.FILES
000600 //MERGE.FILES DD DISP=SHR,DSN=SAMS.DISK.FILES
000700 //MERGE.PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
000800 // DD DISP=SHR,DSN=WORKSHOP.R90.PARMLIB
000900 //MERGE.COPYFILE DD DISP=(,PASS),DSN=SAMS.DISK.COPYFILE
001000 //MERGE.SYSIN DD *
001100 MERGE LIMITS=(30,999),
001200 PERCENT=70,
001300 TYPE=TAPE
001400 /*
***** ***** Bottom of Data *****

```

The MERGE control statement goes in the merge step. Therefore the //MERGE.SYSIN DD \* is required with the proper step name.

## Step 7. Recover a DASD Volume

After CA-Disk has been installed and running for some time, the archives contains a recent copy of many data sets. If you experience the loss of a DASD volume due to a hardware or software failure, CA-Disk can help in recovering the lost data. For example, executing the following JCL recovers (and if necessary, scratch) all data sets from all the SSL8/ volume patterns, excluding volume SSL805:

FIGURE 28. Sample RECOVER Command

```

EDIT (RECOVER) - 01.03 Columns 00001 00072
Command ==> Scroll ==> CSR
-----
***** Top of Data *****
000100 //XXXXXXXXX JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X,MSGCLASS=T,NOTIFY=&USERID
000200 // JCLLIB ORDER=WORKSHOP.R90.PROCLIB
000300 //RECOVER EXEC FMS,P='WORKSHOP.R90.INSTALL',Q='WORKSHOP.R90'
000400 //STEPLIB DD DISP=SHR,DSN=WORKSHOP.R90.LOADLIB
000500 //FILES DD DISP=SHR,DSN=SAMS.DISK.FILES
000600 //MSGPRINT DD SYSOUT=*
000700 //PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
000800 // DD DISP=SHR,DSN=WORKSHOP.R90.PARMLIB
000900 //SYSPRINT DD SYSOUT=*
001000 //SYSIN DD *
001100 SCAN DSNINDEX
001200 SELECT DSN=/,VOL=(SSL8/)
001300 EXCLUDE VOL=SSL805
001400 SELECT CAT=/,VOL=(/)
001500 EXCLUDE CAT=/,VOL=SMS/
001600 RECOVER SCRATCH
001700 /*
***** Bottom of Data *****

```

Note: Executing Figure 28 also recovers all catalogs except those backed up from the SMS/ volume pattern.

## Step 8. Compress a PDS

CA-Disk provides a PDS management facility also. This includes some special PDS reports and a PDS compression utility. The PDS reports are run using the standard CA-Disk report procedure and include a PDS Status Report and a PDS Member Report. The member report is a 12-up listing of the member names that reside in the PDS, while the status report includes the data set and directory statistics.

The compression utility can be run explicitly or in a scan mode. The scan mode can evaluate the current status of a PDS to determine if the data set should be compressed. The user specifies certain threshold values that the evaluation module uses to make its determination.

The following step compresses any PDS that resides on volume PACK01 and,

1. is more than 75 percent full
2. has two or more extents
- or
3. has a directory that is more than 90 percent full

FIGURE 29. Sample COMPRES Command

```

EDIT                                     (COMPRESS) - 01.00                Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //XXXXXXXXX JOB (XXXX,XXX), 'NNNNNNNN',
000200 //          CLASS=X,
000300 //          MSGCLASS=T,
000400 //          NOTIFY=&USERID
000500 // JCLLIB ORDER=WORKSHOP.R90.PROCLIB
000600 //COMPRESS EXEC COMPRES,P='WORKSHOP.R90.INSTALL',
000700 //          Q='WORKSHOP.R90'
000800 //STEPLIB DD DISP=SHR,DSN=WORKSHOP.R90.LOADLIB
000900 //COMPWORK DD DISP=(MOD,KEEP,CATLG),
001000 //          DSN=SAMS.DISK.COMPWORK
001100 //MSGPRINT DD SYSOUT=*
001200 //PARMLIB DD DISP=SHR,DSN=WORKSHOP.R90.PARMLIB
001300 //SYSPRINT DD SYSOUT=*
001400 //SYSIN DD *
001500 SCAN VOL=PACK01,
001600 THRESHOLD=(75,2,90)
001700 /*
***** ***** Bottom of Data *****
Aa A TCPIP R 4 C 15 12:54 11/25/97

```

## Step 9. Release Idle Space

If data sets are over-allocated and wasting space, CA-Disk provides a flexible and convenient means to free this space for use by others. Thresholds can be established that allow some extra space but, if exceeded, the unused space is released.

The following example releases all of the over-allocated space found in sequential data sets and PDS data sets. It is not necessary to release all of the over-allocated space, as this example demonstrates.

FIGURE 30. Sample RELEASE Command

```

File Edit Transfer Options Connection Macro Window Help
-----
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT                               (RELEASE) - 01.00                Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //XXXXXXXXX JOB (XXXX,XXX), 'NNNNNNNN',
000200 //          CLASS=X,
000300 //          MGCLASS=T,
000400 //          NOTIFY=&USERID
000500 // JCLLIB ORDER=WORKSHOP.R90.PROCLIB
000600 //RELEASE EXEC DMS,P='WORKSHOP.R90.INSTALL',
000700 //          Q='WORKSHOP.R90'
000800 //STEPLIB DD DISP=SHR,DSN=WORKSHOP.R90.LOADLIB
000900 //FILES DD DUMMY
001000 //MSGPRINT DD SYSOUT=*
001100 //PARMLIB DD DISP=SHR,DSN=WORKSHOP.R90.PARMLIB
001200 //SYSPRINT DD SYSOUT=*
001300 //SYSIN DD *
001400 SCAN REALVOLS
001500 SELECT VOL=PACK01,
001600          CRITERIA=(DSORG,EQ,PS,OR,DSORG,EQ,PO)
001700 RELEASE
***** ***** Bottom of Data *****
Aa  A TCPIP  R 4  C 15  13:00  11/25/97

```

## Special Testing Environment

Testing and implementing CA-Disk functions that make use of a data set's last used date (for example, archival), or perhaps its last modified date (for example, special backup option), requires some special consideration if the CA-Disk SVC (or MVS SU 60) has not been installed long enough to have these fields properly updated on all data sets.

The first option is to simply wait an appropriate period of time before implementing a dependent function. The second is to use the DSCB Update Utility (see ["DSCB Update Utility" on page 109](#)) to initialize these fields to an appropriate starting value. The third is to use the following sysparms for testing, and then follow either option one or two.

Sysparms DSCBLUSD and DSCBLMOD normally provide the decimal offset into the format-1 DSCB where the last used and last modified dates are being recorded. For testing, however, you specify them with an offset of either the creation date (decimal offset 53) or the expiration date (decimal offset 56). For your test purposes, these values cause CA-Disk to treat the creation date (or expiration date) as the last used and last modified dates as well. In the example here, the sysparms have been specified with the offset of the creation date:

```
DSCBLUSD053
DSCBLMOD053
```

---

*Note:* Be sure to remove these entries after the CA-Disk SVC or SU 60 has been active long enough to have recorded proper values. Be sure that you do not have these testing values set if you execute the DSCB Update Utility to initialize the fields!

---

If a local modification to OPEN is already being used to record either or both of these fields in the DSCB, the sysparm entries can be used to direct CA-Disk to the fields maintained by your own local modification. Refer to the descriptions of these system parameters, beginning on [page 140](#) in the *Systems Guide*.

---

## Conclusion of Evaluation

This concludes the beginning evaluation of CA-Disk. Many functions and options have not been covered, but we hope that this preliminary testing has served to introduce some techniques whereby CA-Disk can be used to deal with data storage management problems. “Examples” on [page 493](#) in the *User's Guide* provides assistance in constructing specific applications as needed at your installation.

---

## Testing the CA-Disk Auto-Restore Feature

There are several good reasons to perform thorough system testing of the auto-restore feature. Some of the benefits that can be derived are:

- You become familiar with the characteristics of the system.
- You experience what the system does under varying external conditions.
- The system operators have a chance to get used to the messages that the system issues. They also have a chance to respond to its operator prompts where they are required.
- If a user screening exit has been written, this provides an excellent opportunity for testing its effectiveness.
- You can formulate some ideas of your own for implementation guidelines.

## Forcing an Auto-Restore to be Initiated

To force invocation of the catalog management support, you must first recatalog an archived data set to the CA-Disk pseudo-volume. The easiest way to do this is to run an explicit CA-Disk archive job with the parameter DISP=RECAT specified. This causes CA-Disk to recatalog the data set after the scratch has taken place.

After you have the data set recataloged, you can invoke the support by either running a batch job that references the data set (such as IEBGENER), or by attempting to edit the data set from a TSO/ISPF environment. If you do this from an online environment, you are prompted by several messages:

1. Do you want to restore the data set?,
2. Do you want an immediate or deferred restore?,  
and, if an immediate restore is requested,
3. Do you want to wait for the restore to complete?

Each of these questions is explained more thoroughly in [“Customizing the TSO/ISPF Auto-Restore Environment” on page 77](#).

## Testing Auto-Restore in a Production Environment

There are many possible job combinations that can be run to simulate production environments. The following list details possible tests that can be run. The output reports created by the system task DMSAR should be routed to hard copy so you can review. All data set references in the examples are to non-existent data sets; that is, no format-1 DSCBs exist for the data set.

1. Run a test specifying a data set that can be found in the archives index, and one that cannot.
2. Execute two auto-restore tasks simultaneously by forcing two jobs to execute at the same time (their job names must be different), each job requesting a different archived data set.
3. Try running the test as in 3 above, but this time reference different data set names that reside on the same archive tape volume (you will have to do a LISTD on an archive volume to find a suitable pair of data sets).
4. If you are supplying an ARESPREX screening exit, or using other options to tailor the function to your environment, try running a few conditions that test your implementation. Make sure the restore requests are processed as expected.

After you have exercised enough test conditions to feel comfortable with the auto-restore feature, you are ready to move on to planning the implementation. Be sure any changes you made for testing (such as sysparm values or procedure references) are changed back for a production environment.

This utility permits you to update most fields in the format-1 DSCB, in both IBM and CA-Disk defined fields. You can verify your updates very easily by running VTOC reports before and after execution of the update utility. For standard IBM fields, either CA-Disk reports or IBM VTOC listings can be used for the verification. When fields unique to CA-Disk are being updated, the changes can be verified by running the DSUTIL report for the respective data sets.

---

## Overview

The format-1 DSCBs to update are selected by data set name or pattern and volume name or pattern. An exclusive enqueue is obtained for each data set unless the DSENG parameter is supplied. If the CA-Disk interface to security checking routines has been installed, it is active for this utility as well. Before updating a DSCB, CA-Disk verifies that the requestor is authorized to both update and scratch the associated data set. The update utility runs in simulate mode unless the parameter NOSIM is supplied on the input command.

Implicit updates generate the following message on the operator console:

```
DMS3190 F1DSCB UPDATE UTILITY STARTED, ENTER APPROVAL IF OK
```

To finish executing the update utility, the operator must reply with an "O". Any other response terminates the update.

## Caution

This utility performs the functions of an automated and fairly sophisticated "superzap" utility for the VTOC. As a result, the benefits and convenience can be very rewarding. However, the potential for causing tremendous damage is equally as great, especially when it is run in implicit mode. Any implicit update should always be run in

simulate mode first to ensure you update only the data sets you want to update. Consider using your security system to restrict access to program ADSDM235, or move that module to a library where even read access is tightly controlled!

## Condition Codes

The following condition codes are returned from the DSCB Update Utility:

**TABLE 16. DSCB Update Utility Condition Codes**

Code	Description
0	Updates performed as requested
2	Error in specification of a field value
3	Could not acquire specified DSENQ (that is, EXC or SHR)
4	Security authorization failed for one or more data sets
5	Field name not defined
8	data set or volume not found
12	Operator denied the request
24	I/O error reading format-1 DSCB

## JCL — DSCB Command

The following JCL is required to update one or more DSCBs.

```
//UPDATE EXEC PGM=ADSMI002,PARM=ADSDM235,REGION=1024K
//STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOADLIB
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=A
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//SYSIN DD *
        DSCB NOSIM,DSNAME=ddd,VOLUME=vvvvvv,.....
```

## SYNTAX

One or more DSCB UPDATE commands can be placed following the SYSIN dd statement.

```
DSCB DSNAME=, VOLUME=, DSENQ=, NOSIM, LUSDT=, LMODT=, JBNM=, CREDIT=, EXPDT=, OPCNT=, OPCNTH=,
    LSTVL=, SVCMODE=, PSWRD=, CHBIT=, PTRDS=, DSSNO=, VSEQN=, DSORG=, RECFM=, ICFCD=, BLKLN=,
    LRECL=, KEYLN=, RKEYP=, RACFBIT=, L8BLKBIT=, CHKPTBIT=, SCALO=, LSTAR=, TRBAL=, ALLOTYP=,
    NOEPV=, NOBDB=, FIELD=, FROMCOL=, ONLY
```

**DSNAME=** - The data set name(s) or pattern(s) to supply the key for the format-1 DSCB(s) to be updated. Up to 20 data set or pattern names can be entered.

**VOLUME=** - This parameter can optionally be specified to designate which volume(s) contain the DSCB when the system catalog is not to be used. If this is an implicit update (pattern matching used), the volume(s) or pattern(s) must be specified. Up to 50 volume or pattern names can be entered.

**DSEQ=** - The type of enqueue to perform for each data set updated. There are three valid values: EXC for an exclusive enqueue (this is the default), SHR for a shared enqueue and NO for no enqueue.

**NOSIM** - This parameter is the reverse of the standard SIM parameter. The update utility runs in simulate mode unless this parameter is supplied.

**LUSDT=** - A date in an accepted CA-Disk format to replace the last use date is specified as the value of this parameter. Unless overridden by a value specified in sysparm DSCBLUSD, this date is converted to the discontinuous binary format and is stored at a displacement of 75 bytes.

**LMODT=** - A date in an accepted CA-Disk format to replace the last modified date for non-VSAM data sets is specified as the value of this parameter. Unless overridden by the value specified in sysparm DSCBLMOD, this date is converted to the discontinuous binary format and is stored at a displacement of 70 bytes. VSAM components are ignored for this processing because VSAM modified date information is maintained in the SYSTEM-TIMESTAMP information in the catalog.

**JBNM=** - The non-VSAM job name or accounting field that referenced the data set is specified as the value of this parameter. Unless overridden by the value specified in sysparm DSCBJBNM, the job name will be stored at a displacement of 62 bytes. VSAM components are ignored for this processing because this field is an CA-Disk SVC (non-VSAM only) field.

**CREDIT=** - A date in an accepted CA-Disk format to replace the creation date is specified as the value of this parameter.

**EXPDT=** - A date in an accepted CA-Disk format to replace the expiration date is specified as the value of this parameter.

**OPCNT=** - The cumulative count of opens for the respective non-VSAM data set is specified as the value of this parameter. Unless overridden by the value specified in sysparm DSCBOPCD, it will be stored as a binary fullword at a displacement of 78 bytes. The highest value for this parameter is 99999. VSAM components are ignored for this processing because this field is an CA-Disk SVC (non-VSAM only) field.

**OPCNTH=** - The cumulative count of opens for the respective non-VSAM data set is specified as the value of this parameter. Unless overridden by the value specified in sysparm DSCBOPCD, it will be stored as a binary halfword at a displacement of 78 bytes. VSAM components are ignored for this processing because this field is an CA-Disk SVC (non-VSAM only) field.

**LSTVL=** - Specify ON to cause the data set to be indicated as the last volume on which the data set resides. Specifying OFF indicate that the data set is part of a multivolume data set, and not the last part. Sets hex 80 bit of DS1DSIND field at offset dec 93.

**SVCMODE=** - Specify the CA-Disk SVC MODE with which the DSCB has been updated. This parameter should only be specified during conversion of DSCBs maintained with a CA-Disk SVC from a release prior to Release 8.1 to DSCBs maintained with a CA-Disk SVC from a Release 8.1 or later.

**PSWRD=** - Specify NO to turn off both the write protect bit and the read/write protect bit. Specify RW to turn on the read/write protect bit. Specify W to turn on the write protect bits (sets the hex 10 and hex 04 bits of DS1DSIND field at offset dec 93).

---

*Note:* To change a data set that is currently W-protected to RW-protected, first use a command with NO followed by one with RW.

---

**CHBIT=** - Specify ON to turn the change bit on, indicating that the data set contents have changed. OFF turns the change bit off (sets the hex 02 bit of the DS1DSIND field at offset dec 93 as established by SU 60 for MVS).

**PTRDS=** - This is the cylinder, track and record in CCHHR form pointing to the format-3 DSCB for the data set. It is a five-byte hexadecimal field and anything from 0000000000 to FFFFFFFF is accepted. A value of 0000000000 indicates no format-3 DSCB exists.

**DSSNO=** - This parameter replaces the data set serial number which is a six-character field.

**VSEQN=** - This parameter replaces the volume sequence number with a value between 0 and 255.

**DSORG=** - The data set organization bytes. This parameter is entered as a two-byte hexadecimal value. Anything from 0000 to FFFF is accepted. Standard values are:

8000	for indexed sequential (ISAM)
4000	for sequential (BSAM, QSAM)
2000	for direct (BDAM)
0200	for partitioned (BPAM)
0008	for VSAM

**RECFM=** - The record format field. This parameter is a one- byte hexadecimal value. Anything from 00 to FF is accepted. Standard values are:

C0	bits for undefined
80	bit for fixed
40	bit for variable
20	bit for track overflow
10	bit for blocked
08	bit for standard (fixed) or spanned (variable)
04	bit for ASA control characters
02	bit for machine control characters

**ICFCD=** - This is the OPTCD field in the DSCB. It indicates whether a data set is an ICF catalog or a cluster defined in an ICF catalog. It is a one-byte hexadecimal value. Anything from 00 to FF is accepted. Standard values are:

80	for an ICF catalog
40	for cluster defined in an ICF catalog

**BLKLN=** - The data set block size. The value range is 0 to 32760.

**LRECL=** - The logical record size for the data set. It can range from 0 to 32760.

**KEYLN=** - The key length is a one-byte field. The highest value accepted is 255.

**RKEYP=** - The relative key position is the offset within the data block of the data key. Values from 0 to 32760 are accepted.

**RACFBIT=** - Specify ON to indicate the data set is RACF- protected, and OFF to remove the RACF protection.

**L8BLKBIT=** - Specify ON to indicate the block length supplied in the DSCB is given as a multiple of eight bytes and OFF to indicate the block length is the real length in bytes.

**CHKPTBIT=** - Specify ON to indicate this is a check point data set. Specify OFF to remove this indication.

**SCALO=** - Replaces the secondary allocation value. The highest value accepted is 99999.

**LSTAR=** - The last relative track and record written for a data set. This parameter is entered as a three-byte hexadecimal value. The first two bytes are the relative track number and the third byte is the record number. Anything from 000000 to FFFFFFFF is accepted.

**TRBAL=** - The number of bytes remaining on the last track written on a data set. Values from 0 to 32760 are accepted.

**ALLOTYP=** - The allocation type field. This parameter is a one-byte hexadecimal value. Any value from 00 to FF is accepted. Standard values are:

C0	bits for cylinder
80	bit for track
40	bit for block
00	bits for absolute
08	bit for contiguous
01	bit for rounding

**NOEPV=** - Number of extents on this volume for the data set. The maximum valid value is 16.

**NOBDB=** - Number of bytes used on the last directory block for this partitioned data set.

The following parameters allow you to move CA-Disk-supported fields from one position in the format-1 DSCB to another. Before using these parameters, decide on a new offset for a field and update your CA-Disk SVC and the appropriate CA-Disk sysparm to reflect the new offset. Then supply a list of the CA-Disk-defined fields to move and their original offsets in the format-1 DSCB. The target offsets are obtained from the sysparm that corresponds to that CA-Disk-defined field. This utility then gets the value from the input offset and stores it at the sysparm-defined offset.

**FROMCOL=** - A list of offsets for the fields specified in the FIELD parameter. There must be one offset for each field name. Data is moved from the offsets specified here to the new offsets as specified by the corresponding CA-Disk sysparms.

**ONLY** - Specifies that only DSCBs identified with a CA-Disk SVC MODE value of less than 4 be updated. A value of less than 4 indicates that the DSCB was last maintained with a CA-Disk SVC from a release prior to Release 8.1.

- This parameter prevents the DSCB Update Utility from converting DSCBs that were previously converted.

**FIELD=** - A list of CA-Disk fields to move. Up to seven field names can be supplied. The valid field names are:

**TABLE 17. List of DSCB Fields that can move**

Field	Description
CRJB5	first five bytes of creating job name (pre-R8.1)
CRJB3	last three bytes of creating job name (pre-R8.1)
LUSJB	job name or account field (pre-R8.1)
LMODT	last modified date
LUSDT	last use date
OPCNT	cumulative open count in a fullword
OPCNTH	cumulative open count in a halfword

Each of these fields has a corresponding sysparm that supplies the target offset for the move.

*Note:* For the open count, if the from count is a fullword value, specify OPCNT as the “from field” name. If it is a halfword value, specify OPCNTH as the “from field” name. If you are changing the open count from a fullword to a halfword, make certain that you specify OPCNT for the fullword from field, and that sysparm DSCBOPCD has an “H” specified in the fourth position (that is, DSCBOPCDxxxH) for the new halfword target field. If you are changing the open count from a halfword to a fullword, be sure that you specify OPCNTH for the halfword from field, and that sysparm DSCBOPCD has a blank specified in the fourth position for the new fullword target field.

## User Exits

The user exit “[DSCBCVEX — Format-1 DSCB Conversion Exit Location](#)” on page 217 in the *Systems Guide*, is also available to help you when moving CA-Disk-support fields.

---

This maintenance feature reads the DSNINDEX archive index and catalog those data sets that are no longer on DASD. This permits these data sets to be auto-restored if they are required some time in the future. The facility has several options that allow the user to catalog only a subset of archived data sets, depending on command parameters specified. Simulation mode can be specified to produce a report showing all catalog actions that would take place, without performing any catalog updating.

---

### *IXCATLG Function Description*

When you run the IXCATLG utility, CA-Disk reads the archive index sequentially until all archive records are processed. Only the most current index record for each data set is examined. If the data set was not scratched when it was archived, the index record is ignored and the data set is not cataloged.

If parameter VSAMONLY is specified, only data sets marked as VSAM in the index is processed. Conversely, if NONVSAM is specified, only non-VSAM data sets are processed. The default is to process all data sets.

If the DSN= and/or VOL= parameters were specified, the data set name and/or archive volume is matched against the pattern(s) specified. If they both match, or if only one was specified and it matches, the data set is selected for processing. Then a check is made to see if the EXCLUDE parameter was specified. If it was, the selection decision made by the DSN= and VOL= parameters is reversed. That is, if the data set would have been included it is now excluded, and vice versa.

If the data set has passed all selection testing to this point, a check is made to determine if the DATE= parameter was specified. If it was, the archive date in the DSNINDEX record is compared against the date entered on the command. If the archive date is less than the command date, the data set is bypassed. If the DATE= parameter is not specified, processing continues for the data set.

At this point the data set has gone through all preliminary screening. The user exit IXCTLGEX is now invoked, if specified, with the DSNINDEX record currently being processed and a selection flag. The exit can change the selection flag to either include or exclude the data set. More information regarding the exit can be found below.

If the data set passed all screening requirements, a catalog locate is issued to determine if the data set is currently cataloged. If it is not currently cataloged, an obtain is issued for the format-1 DSCB on the volume from which the data set was archived. If the data set is not on the volume, it is cataloged to the CA-Disk pseudo-volume (assuming SIMULATE mode was not specified).

If the data set is currently cataloged, CA-Disk checks to see if the cataloged volume is the same as the pseudo-volume. If it is, no processing is required for the data set. If it is not the same volume, CA-Disk checks to see if a list of staging volumes was specified (STAGEVOLS= parameter). If none were specified, or if the catalog volume matches one in the STAGEVOLS= list, CA-Disk issues an obtain for the format-1 DSCB of the data set on the volume to which the data set is cataloged. If it is not found on that volume, the data set is recataloged to the pseudo-volume (again assuming SIMULATE mode was not specified).

At the end of the run, CA-Disk prints statistics about the number of catalog actions required to perform the conversion.

We recommend that you run this utility in a simulate mode first to verify that the proper data sets are cataloged. Because of the number of locates, obtains and catalog actions processed by this utility, a significant amount of run time can be expected. The more active your installation is in archival (with scratch specified), the higher the CPU usage is.

```
//CATLG      EXEC IXCATLG
//SYSIN      DD   *
CATALOG SIM,DSN= ,VOL= ,EXCLUDE ,DATE= ,EXPIRED ,FAILSAFE ,VSAMONLY ,NONVSAM ,STAGEVOLS=
```

## IXCATLG Command and Parameters

**SIM** - Specify this OPTIONAL parameter to run the utility in simulate mode, which will determine which data sets would be cataloged if this job were run in a live mode. This parameter allows the catalog utility to perform all processing except the final cataloging action. This parameter does not reduce significantly the amount of CPU time required by the maintenance utility.

**DSNAME=** - Use this OPTIONAL parameter to specify from 1 to 50 data set names and/or patterns. If this parameter is specified, only those data sets that match any pattern are cataloged. This parameter's function can be changed by parameters VOLUME= and EXCLUDE. See these parameters for more details.

---

*Note:* If the IXCATLGEX exit is being invoked, then ALL data sets (selected and non-selected) are passed to the userexit.

---

**VOLUME=** - Use this OPTIONAL parameter to specify from 1 to 50 volumes and/or patterns. If this parameter is specified, only those data sets that were archived from one of the DASD volumes specified are cataloged. If the DSNAME= parameter is also coded, the data set must match one of the data set name patterns specified and it must have been archived from one of the volumes specified. This parameter's function can be changed by parameter EXCLUDE.

**EXCLUDE** - Use this OPTIONAL parameter in conjunction with the DSNAME= and/or the VOLUME= parameters. When this parameter is specified, the patterns entered for DSNAME= and VOLUME= become exclusion lists and only those data sets NOT matching the patterns are cataloged. EXCLUDE has no effect on any parameters other than those mentioned.

**DATE=** - Specify this OPTIONAL parameter to limit the data sets being cataloged to those that were archived on or after the date specified. The date can be specified in any standard CA-Disk format.

**EXPIRED** - Specify this OPTIONAL parameter to allow cataloging of expired data sets. Default processing excludes expired data sets from being cataloged.

**FAILSAFE** - This REQUIRED parameter serves no function in the recataloging facility, but was added as a failsafe mechanism to prevent users with “old” JCL from submitting the same job stream without performing updates. If this parameter is not supplied, the job fails with a command input error.

**NONVSAM** - You can specify this OPTIONAL parameter to force CA-Disk to process only non-VSAM data sets. Default processing is that all data sets are eligible.

**VSAMONLY** - You can specify this OPTIONAL parameter to force CA-Disk to process only VSAM data sets. Default processing is that all data sets are eligible.

**STAGEVOLS=** - You can specify up to ten volume names. This OPTIONAL parameter was designed for CA-Disk users who used the auto-restore capability in a prior release (pre-7.1) and recataloged archived data sets to staging volume(s). This parameter is intended as a means of reducing the CPU requirements of this job. When this parameter is specified, an obtain for the format-1 DSCB is attempted only if the data set is currently cataloged to one of the volumes specified in the STAGEVOLS= parameter.

## IXCTLGEX - IXCATLG User Screening Exit

This screening exit allows the user to modify individually the data set selection made by the IXCATLG maintenance utility. This exit is invoked after all selection tests have been made, based on the input parameters specified by the user.

It is passed two parameters:

1. the DSNINDEX record being processed
2. the selection flag field

The selection flag is set to either a Y or N when the exit is invoked, Y meaning the data set is cataloged, N meaning it is bypassed. Based on any information in the DSNINDEX record, the screening exit can change it to either a Y or N value.

Sample source for the IXCTLGEX module is supplied in the CA-Disk DMSASM library. It does no processing by default. To code your own exit, you can modify this version of the module. For additional information, refer to [“IXCTLGEX — User Exit for IXCATLG Utility” on page 223 in the Systems Guide](#).

If your current release of CA-Disk is 8.1.7 or earlier, you must convert your files data set into a different format. The following steps describe how to accomplish the conversion.

1. Make a backup copy of the current files data set.
2. Run an UNLOAD pointing to the current files data set and the current LOADLIB using the following sample:

FIGURE 31. Sample UNLOAD JCL and Command

```

-----
EDIT                               (UNLOAD) - 01.00                               Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //XXXXXXXXX JOB (XXXX,XXX), 'NNNNNNNN', CLASS=X,MSGCLASS=T,NOTIFY=&USERID
000200 //UNLD      EXEC PGM=ADSM1000, PARM=ADSDM177, REGION=4096K
000300 //STEPLIB   DD  DISP=SHR,DSN=WORKSHOP.R90.LOADLIB
000400 //ABNLDUMP  DD  DUMMY
000500 //CMDPRINT  DD  SYSOUT=*
000600 //FILES     DD  DISP=SHR,DSN=SAMS.DISK.FILES
000700 //MSGPRINT  DD  SYSOUT=*
000800 //PARMLIB   DD  DISP=SHR,DSN=SAMS.DISK.PARMLIB
000900 //          DD  DISP=SHR,DSN=WORKSHOP.R90.PARMLIB
001000 //SYSPRINT  DD  SYSOUT=*
001100 //SYSUDUMP  DD  SYSOUT=*
001200 //SEQFILES DD  DISP=(,CATLG,DELETE),
001300 //          DSN=SAMS.DISK.SEQFILES,
001400 //          DCB=(LRECL=268,BLKSIZE=6144,DSORG=PS,RECFM=VB),
001500 //          SPACE=(6144,(360,360),RLSE),UNIT=SYSALLDA
001600 //SYSIN     DD  *
001700 UNLOAD ALL
001800 /*
***** ***** Bottom of Data *****
Aa  A TCP/IP      R 4  C 15      13:53  11/25/97

```

- 
3. Estimate the capacity needs of the files data set. Refer to [“Estimating Subfile Capacity Requirements” on page 20 in the Systems Guide](#).
    - The physical space your files data set resides on increases.
    - This increase occurred when the logical record length of the DSNINDEX subfile changed in release 8.1 from 090 to 256.
    - Calculate your new space requirements for the files data set. Refer to [“Estimating Physical Space Requirements” on page 21 in the Systems Guide](#).
  4. Create a FILEDEFN member in your Release 9.0.2 parmlib.
    - You can use a FILEDEFN member from a prior release if you ensure the LRECL and KEY lengths for the following entries are adjusted as shown:

```
DSNINDEX256044C00075000YNNY
RESTCMDS164044C00005000YYNN
```

- You can create a new FILEDEFN member. If you create a new FILEDEFN member, use member FDSAMPLE in the Release 9.0.2 parmlib as a sample.
5. Run a FILEINIT, using the FILEDEFN member you selected in step 4, to allocate and initialize the files data set. The following sample JCL can also be found as member FILEINIT in the installation library.

```
//FILEINIT EXEC PGM=ADSMI002,PARM='ADSDM100'
//STEPLIB DD DISP=SHR,DSN=SAMS.DISK.LOADLIB
//ABNLDUMP DD DUMMY
//CMDPRINT DD SYSOUT=A
//FILES DD DSN=SAMS.DISK.FILES,DISP=(,CATLG,DELETE),UNIT=SYSDA,
// DCB=(DSORG=DA),SPACE=(CYL,10,,CONTIG)
//MSGPRINT DD SYSOUT=A
//PARMLIB DD DISP=SHR,DSN=SAMS.DISK.PARMLIB
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
```

6. Run a RELOAD pointing to the UNLOADED files data set (created in step 2) and the 9.0.2 formatted files data set (created in step 5) using the 9.0.2 LOADLIB.

FIGURE 32. Sample RELOAD JCL and Command

```
File Edit Transfer Options Connection Macro Window Help
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT                               (RELOAD) - 01.00                Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000100 //XXXXXXXXX JOB (XXXX,XXX), 'NNNNNNNN',
000200 //          CLASS=X,
000300 //          MSGCLASS=T,
000400 //          NOTIFY=&USERID
000500 //RELD      EXEC PGM=ADSMI002, PARM=ADSDM192, REGION=4096K
000600 //STEPLIB   DD  DISP=SHR, DSN=WORKSHOP.R90.LOADLIB
000700 //ABNLDUMP  DD  DUMMY
000800 //CMDPRINT  DD  SYSOUT=*
000900 //FILES     DD  DISP=SHR, DSN=SAMS.DISK.FILES
001000 //MSGPRINT  DD  SYSOUT=*
001100 //PARMLIB   DD  DISP=SHR, DSN=SAMS.DISK.PARMLIB
001200 //          DD  DISP=SHR, DSN=WORKSHOP.R90.PARMLIB
001300 //SYSPRINT  DD  SYSOUT=*
001400 //SYSUDUMP  DD  SYSOUT=*
001500 //SEQFILES  DD  DISP=OLD, DSN=SAMS.DISK.SEQFILES
001600 //SYSIN     DD  *
001700 RELOAD ALL, FORMAT
001800 /*
***** ***** Bottom of Data *****
Aa  A TCPIP          R 4  C 15          14:06  11/25/97
```

Congratulations! Your files data set has now been converted.

The following information is very important if you have an 8.0 or earlier release of the CA-Disk SVC installed and intend to run CA-Disk release 9.0.2 with that older CA-Disk SVC. If you have an 8.1 or later release of the CA-Disk SVC installed, skip this section.

With Release 8.0, the CA-Disk SVC maintained the following fields by default:

**TABLE 18. 8.0 CA-Disk SVC Maintained Fields**

<b>Equate Label</b>	<b>Sysparm</b>	<b>Location</b>	<b>Explanation</b>
DMSUSEDT	DSCBLUSD	75	LAST OPENED DATE (SU 60)
DMSDSIND	N/A	93	CHANGE BIT X'02' (SU 60)
DMSCRJ05	DSCBCRJ5	70	CREATING JOB NAME (FIRST 5 BYTES)
DMSCRJ03	DSCBCRJ3	45	45 CREATING JOB NAME (LAST 3 BYTES)
DMSMODDT	DSCBLMOD	48	LAST MODIFIED DATE
DMSSVCMO	N/A	103	SVC MODE (TYPE 1)
DMSUSJOB	DSCBUSJD	62	LAST USING JOB NAME
DMSOPCNT	DSCBOPCD	00	OPEN COUNT (FULL WORD)

With Release 8.0, CA-Disk stopped maintaining the open count because the SMS subsystem uses offset 78 to keep control information. Previously, CA-Disk used offset 78 to keep the open count.

The Release 9.0.2 version of the CA-Disk SVC now maintains the following fields by default:

**TABLE 19. 9.0.2 CA-Disk SVC Maintained Fields**

<b>DSCB Update Parm</b>	<b>Equate Label</b>	<b>Sysparm</b>	<b>Location</b>	<b>Explanation</b>
LUSDT	DMSUSEDT	DSCBLUSD	75	LAST OPENED DATE (SU 60)
CHBIT	DMSDSIND	N/A	93	CHANGE BIT X'02' (SU 60)
LMODT	DMSMODDT	DSCBLMOD	70	LAST MODIFIED DATE
SVCMODE	DMSSVCMO	DSCBSVMD	103	SVC MODE (TYPE 4)
JBNM	DMSJOBNM	DSCBJBNM	62	JOB NAME/ACCT INFO
OPCNTH	DMSOPCNT	DSCBOPCD	73	OPEN COUNT (HALF WORD)

Several changes were made for Release 8.1. The last modified date was moved to a new offset. The value of sysparm DSCBLMOD was changed to reflect the new offset. CA-Disk now supports only one job name or eight-byte account code. The open count was reinstated as a half word value. The value of sysparm DSCBOPCD was changed to reflect the new offset of where the data set open count is now maintained. Three sysparms (DSCBCRJ5, DSCBCRJ3, and DSCBUSJD) were deleted and two new sysparms (DSCBJBNM and DSCBSVMD) added.

Installing the new CA-Disk SVC is optional. Whether you install this new CA-Disk SVC or not, several tasks must be completed to allow Release 9.0.2 to work when using the information maintained in the F1-DSCB to process data sets. **Read this appendix in its entirety.**

If you do not install the new CA-Disk SVC, review the specification of several CA-Disk sysparms and review the effects of processing these fields. Further information is provided later in this appendix.

Do not run the Release 9.0.2 CA-Disk SVC with an 8.0 release of CA-Disk. The CA-Disk SVC automatically attempts to convert F1-DSCBs to the new format. An 8.0 release of CA-Disk cannot detect and handle the two different formats.

Do not run the Release 9.0.2 CA-Disk SVC on a system which shares DASD with other systems running the Release 8.0 CA-Disk SVC. Since data sets can be updated by both CA-Disk SVCs and the old CA-Disk SVC does not know of the new offsets, DSCBs can become corrupted after they were converted.

CA-Disk Release 9.0.2 can detect the difference between old format and new F1-DSCBs. You can use Release 9.0.2 to process all data sets. Old format DSCBs need to be converted.

**POSSIBLE DATA LOSS** - If you changed the default offsets for the fields maintained within the F1-DSCB, the use of MODDT (last modified date) selection criteria is only effective for converted F1-DSCBs. It is possible for data sets not to be selected for CA-Disk processing based on this criteria. Also, reports on unconverted F1-DSCBs can show blank or incorrect values for the last modified date field.

**Warning** - Changing CA-Disk SVC modes incorrectly can result in overlapping TTR problems. The procedure necessary to change from one mode to another is to first locate the following line in the ADSMVS60 source code:

```
SVCMODE DC C'4'
```

Then change it to the mode you want.

---

## Conversion for the CA-Disk SVC

In the new CA-Disk SVC, the default locations of several fields have changed from those run prior to 8.1.0. These changes were made for two reasons: 1) to accommodate several changes made by IBM wherein previously unused fields within the F1-DSCB are now used; and 2) to include support for job accounting information. To maintain compatibility with other software products, we highly recommend that you use the new CA-Disk SVC. Using the new CA-Disk SVC helps you avoid problems over new releases of CA-Disk and other products.

To use the new offsets, you must convert your existing F1-DSCBs. This conversion must be done in a logical sequence by following the steps below in the order shown.

1. Install CA-Disk Release 9.0.2 on all systems. You must have Release 9.0.2 running on all systems that share DASD before you install the new CA-Disk SVC.

Since the new CA-Disk SVC also does conversion of DSCBs to the new format, there will be a time when both old and new format DSCBs exist. Release 9.0.2 understands both new and old formats of the DSCB fields. Older releases do not. Under an older release, data sets can be either not selected or improperly selected for critical CA-Disk functions such as backup or archive.

2. Install the 9.0.2 release of the CA-Disk SVC on all systems. You must, however, install the new CA-Disk SVC on all systems at the same time. Since the old and new CA-Disk SVCs are not compatible, running them together on shared DASD can corrupt the fields maintained in the DSCBs.
3. Convert existing F1-DSCBs to new offsets. The three methods that can be used to convert existing F1-DSCBs to the new offsets are: 1) implicitly, using the new CA-Disk SVC; 2) implicitly, using any CA-Disk function that updates it; 3) explicitly, using the DSCB UPDATE utility.

The implicit methods take the longest time, since they require each data set to be processed before the conversion takes place. For certain data sets, this processing can take some time.

The explicit method needs to be done only if the JOBNAME or the MODDT field is required in your routine processing. Don't confuse the MODDT field with the MODIFIED DSCL criteria, there is no functional relation between the two.

The conversion can be done in phases because each of the three methods can distinguish between converted F1-DSCBs and unconverted ones. CA-Disk makes this distinction based on the value of the SVC MODE.

Before the F1-DSCB is converted, the value of the SVC MODE is 1, 2, or 3. After it is converted, the value is 4, 5, or 6. Old SVC MODE 1 corresponds to new SVC MODE 4; old MODE 2 corresponds to new SVC MODE 5; old MODE 3 corresponds to new SVC MODE 6.

The following is a discussion of how DSCBs are converted and what you must do to ensure that the conversion is done correctly.

## Conversion within the CA-Disk SVC

The CA-Disk SVC converts each old F1-DSCB it encounters. It moves the appropriate fields to their new positions and change the SVC MODE to indicate that it has been converted.

The exact process used to convert each F1-DSCB can be seen in the code of the CA-Disk SVC (member ADMSVS60 in the DMSASM library). A "FIND" on "CONVERT" in column 1 positions the cursor on this conversion code.

By default, the conversion code retains the creating job name, move the last modified date, reset the open count, update the SVC MODE, and restore the volser field.

Conversion is done using the old default offsets. If you modified the CA-Disk SVC and maintained fields at different offsets, you must modify the conversion code to reflect your changes. Also, to maintain the last using job name instead of the creating job name, you must update the code to reflect this.

## Conversion within Normal CA-Disk Processing

When CA-Disk functions cause reallocation of data sets (for example, Restore, Move, and PDS Compress), conversion can also take place. This conversion is done through a user exit. Sysparm DSCBCVEX identifies the program to receive control and do the conversion. A sample conversion exit has been provided in member ADSEX016 of the DMSASM library.

By default, the conversion exit retains the creating job name, moves the last modified date, resets the open count, and updates the SVC MODE.

Conversion is done using old default offsets. If you have modified the CA-Disk SVC and maintained fields at different offsets, modify this exit to reflect your changes. Also, to maintain the last using job name instead of the creating job name, you must update the exit to reflect this.

Do not begin using this conversion exit until you have the new CA-Disk SVC installed. Since the old CA-Disk SVC does not know of the new offsets, these fields can become corrupted after they are converted.

After you have determined that this exit works as desired, specify sysparm DSCBCVEX with the value of "ADSEX016". This value causes conversion to take place as CA-Disk reallocates data sets during its processing.

## Conversion Using the DSCB Update Utility

You can also use the DSCB UPDATE utility to convert to the new offsets for all data sets on DASD. This process can be phased in by processing a volume at a time and by using the ONLY parameter on your DSCB UPDATE command. The ONLY parameter causes the DSCB UPDATE utility to verify the SVC MODE before data is moved, and therefore prevents the conversion of previously converted DSCBs.

Use the Release 9.0.2 load library when you run the DSCB UPDATE utility. See ["DSCB Update Utility" on page 109](#) for more information.

Do not begin using the DSCB Update Utility to convert F1-DSCBs to the new format until you have the new CA-Disk SVC installed. Because the old CA-Disk SVC does not know of the new offsets, these fields can become corrupted after they are converted.

Use the following steps in setting up your DSCB Update Utility commands:

1. Decide which fields in the F1-DSCB need to be converted.
2. In your Release 9.0.2 parameter library, review and update all sysparms which identify the offsets for CA-Disk fields within the F1-DSCB. All these sysparms are prefixed with "DSCB". The value of these sysparms should reflect the new offsets.
3. Run the DSCB UPDATE utility. Use simulate mode first to determine if your commands are correct.

**Examples:**

- Convert the last modified date:

```
DSCB NOSIM,VOL=volser,ONLY,SVCMODE=4,FIELD=(LMDT),FROMCOL=(48)
```

- Convert the creating job name:

```
DSCB NOSIM,VOL=volser,ONLY,SVCMODE=4,FIELD=(CRJB3,CRJB5),
FROMCOL=(45,70)
```

- Convert the full word open count to a half word open count:

```
DSCB NOSIM,VOL=volser,ONLY,SVCMODE=4,FIELD=(OPCNTH),FROMCOL=(80)
```

- Convert the last modified date, creating job name and the full word open count in one job:

```
DSCB NOSIM,VOL=volser,ONLY,SVCMODE=4,FIELD=(LMDT,CRJB3,CRJB5,OPCNTH),
FROMCOL=(48,45,70,80)
```

**Running Release 9.0.2 with a Previous Release's CA-Disk SVC**

You can run CA-Disk Release 9.0.2 with an CA-Disk SVC from a previous release. This can be necessary until you have installed Release 9.0.2 on all systems which share DASD. To run Release 9.0.2 and maintain compatibility with an old CA-Disk SVC, take the following steps:

1. In your Release 9.0.2 parameter library, review and update all sysparms which identify the offsets for CA-Disk fields within the F1-DSCB. All these sysparms are prefixed with "DSCB". The value of these sysparms should reflect the offsets used by your current CA-Disk SVC.

The following is an example of how sysparms should be set if you are running Release 9.0.2 with the Release 8.0 CA-Disk SVC using the default offsets:

**TABLE 20. Sysparm Settings Under CA-Disk 8.0 SVC**

Sysparm Value	Meaning
'DSCBJBNM062'	Last Using JOBNAME
'DSCBLMOD048'	Last Modified Date
'DSCBOPCD000'	Open Count (not maintained)
'DSCBSVMD103'	SVC Mode

2. Because only one job name field is supported in Release 9.0.2, sysparm DSCBJBNM is set by default to point to the creating job name field. So, only the creating job name is maintained during Restore, Move and PDS Compress processing. Also, only the creating job name shows up on the DSUTIL report.
3. The fields which contain the last using job name are no longer maintained. So, the last using job name is lost after the above CA-Disk functions have processed the data set. To continue maintaining the last using job name, you must convert the DSCB by moving the last using job name to the last using job name field.

---

This exit is not recommended because some installations, particularly those using SMS, are less likely to have hard coded volsers in JCL. Also, using this exit can result in restore errors because the time at which it is called does not allow for data set allocations to change volumes. These changes are more likely to occur in an SMS environment.

---

### *“F1-DSCB Not Found” S213 Abend Exit (IFG0EX0A Exit)*

The recommendation to not use this exit is:

1. The methods that cause the exit to be invoked are not generally practical and they are not compatible within an SMS environment. These methods are:
  - Archiving data sets and not changing the volume indicator in the catalog to reflect that they are archived.
  - Coding the VOLSER in JCL to allocate to data sets that are already in existence.
2. The S213 exit has the following significant limitations:
  - Only non-VSAM data sets can be auto-restored.
  - No pooling capabilities.
  - Restores cannot be invoked by ISPF.
  - Installation requires an IPL.
  - False invocation of this exit has been know to occur.

The Catalog Management hook (see [“The Two Auto—Restore Methods” on page 59](#)) has the following advantages over this exit:

1. VSAM and non-VSAM data sets can be auto-restored.
2. Restored data sets can be pooled by CA-Disk, CA-Allocate or DFSMS.
3. Restores can be invoked by ISPF.
4. Installation and/or removal of the Catalog Management hook is accomplished dynamically with a started task.

The first exit that CA-Disk supported was the "F1-DSCB-not-found" user exit in OPEN and EOV processing. The exit is also called the "S213abend" exit. It is invoked during OPEN processing (non-VSAM) when the data set being opened is cataloged to a volume, but no F1-DSCB exists on that volume for that data set. It is also invoked by batch jobs when the volume of a data set is hard-coded in the JCL, but does not exist on the volume.

At this point, the exit is invoked as a last-ditch effort to save the job from a system 213abend. If the data set is restored to the volume by the exit, it can signal OPEN to retry the DSCB search again; since it just restored the data set, there should be one out there! If it couldn't restore the data set, the job gets the S213abend. The dummy exit that IBM supplies with the base MVS system merely sets the return code to tell OPEN to go ahead and abend. The exit name that gets called is IFG0EX0A.

Why two different methods? Because neither of them alone can initiate auto-restores at all times when they are needed. The IFG0EX0A exit does not get called for VSAM data sets, nor does it get invoked for TSO/ISPF functions-- the catalog management hook does, however. So why not implement just the catalog management support? If you only did that, you cannot automatically restore data sets in batch jobs that had the volume hard-coded in the JCL, or when OPEN is processing a data set that is cataloged to a real volume on which the data set doesn't reside (F1-DSCB missing!). Conceptually, think of the two methods as complementary functions. Whenever processing for a data set is going to occur, two primary questions must be answered:

1. What volume does the data set reside on?
2. What are its attributes on the volume?

To answer the first question, catalog management is invoked with a locate option, which asks where the data set is currently located. It searches its catalogs until it finds the entry for the data set. With this information, OPEN can allocate the volume and then perform an obtain for the F1-DSCB to get the data set's attributes and extent information. By understanding the order of information retrieval, you understand how these methods work, and also why the catalog management hook allows more flexibility.

Assume in the above example that the data set is cataloged to a real volume instead of the CA-Disk pseudo-volume. If the data set was being opened for processing, a catalog locate is issued to determine the location of the data set. Since the data set is cataloged to a real volume, the catalog management hook lets the locate pass through without any modifications. OPEN processing allocates the volume(s) for the data set. It issues an obtain for the F1-DSCB that describes that data set. If this obtain fails, the IFG0EX0A exit gets control to determine if it can restore the data set. If it can restore the data set at this point, however, it must go back to the volume to which it is cataloged. OPEN has already allocated the volume and is expecting the data set on that volume only. This is why the catalog management hook allows more flexibility--it intercepts the locate before OPEN allocates the volume(s). It can therefore put the data set back to any volume it wants, as long as it passes back the correct volume list to the requestor of the locate.

## Installing the S213 Abend Exit

Installing the S213 exit requires an IPL.

1. If you installed module IFG0EX0A into your LPALIB from a release of CA-Disk prior to Release 7.1, remove it as follows.
  - Assemble and link member IFG0EX0A, contained in the CA-Disk DMSASM library, into your CA-Disk LPA list library.
  - If you also have DFHSM and want CA-Disk and DFHSM to auto-restore data sets, use the IFG0EX0A supplied by DFHSM.
2. If you did not install module IFG0EX0A into your LPALIB from a release of CA-Disk prior to Release 7.1, continue here.

3. Sample JCL is provided in the installation library member USERMOD7.
4. After completing the relink, run the following JCL to verify your changes.

```
//LIST      EXEC PGM=AMBLIST
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DISP=SHR,DSN=SYS1.LPALIB
//SYSIN    DD  *
           LISTLOAD MEMBER=(IFG0EX0A)
```

Make sure that the correct version of IFG0EX0A has been link-edited. If DFHSM is not installed, the IFG0EX0A CSECT in module IFG0EX0A should consist of only two instructions:

```
SR 15,15
BR 14.
```

5. IPL your system with either the CLPA or MLPA option. The auto-restore function is now active. If MLPA is used, be sure to add the proper entry to SYS1.PARMLIB member IEALPaxx and specify it during the IPL.

---

*Note:* When changing modules that reside in LPA, it can be common practice to link the new version into a linklist library other than SYS1.LPALIB, and then use the MLPA ability to test the new module. If all testing goes well, you might use IEBCOPY to copy and replace the old version in SYS1.LPALIB, and then schedule an IPL with a CLPA. This all works fine, but only if you specify all of the alias names as well as the true member name in the SELECT MEMBER=(...list...) statement for IEBCOPY. Omitting the alias names can cause the IPL with CLPA to fail with unpredictable symptoms. This can be very confusing, because "The MLPA worked just fine! How can the CLPA possibly not work?"

---

If archived data sets are cataloged to the CA-Disk pseudo-volume, they can be restored by the catalog management hook to any one of a pool of volumes. If the data set is recataloged to a real volume, only the S213 exit of OPEN will be invoked and the restored data set has to go back to that specific volume. The real volume to which archived data sets are recataloged can be thought of as a common "staging" volume for auto-restores. The volume can be monitored for activity and easily managed to ensure enough space to handle the restores.

If the S213 exit is installed, the following restrictions apply:

1. The IFG0EX0A exit does not get invoked for VSAM data sets, and therefore they must be restored manually.
2. The data set must be restored to the same volume to which the 213 was invoked. If sufficient free space does not exist on the volume, the restore fails and the originating task terminates with a system 213 abend.
3. The IFG0EX0A exit is not entered from TSO and ISPF functions.
4. The IFG0EX0A exit is not entered from the IEBCOPY utility.

CA-Disk Message Check is a utility centralizing messages from different jobs for viewing and decision-making from one location. For a given job, the application uses SYSOUT EXIT to filter messages and write them to data sets (message data sets). The system administrator can then run batch jobs to consolidate all message data sets into a single work data set and report on it.

Message filtering is done by comparing JOBNAME's and CA-Disk messages with two user defined tables -- one has JOBNAME/prefix's and the other has CA-Disk message number/word/string's. The two tables are PDS members and can be updated at any time through ISPF panels. The selected messages are written to dynamically allocated message data sets. DSN of a message data set is determined by the user specified prefix (through one of the application's ISPF panel) and JOBNAME. When consolidating message data sets, messages are merged and sorted into generation data sets and all message data sets are then deleted. Message check does not alter any CA-Disk job output.

Currently, message check supports the following DD - MSGPRINT, SYSPRINT, CMDPRINT, RELPRINT, and MOVPRINT.

---

## *Installing CA-Disk Message Check*

**Step 1** - Create a PDS which is used to hold user defined tables (for JOBNAME's, message number/word/string's). The PDS can be RECFM=FB and LRECL=80 with 1 directory block. You need update access to maintain the tables. See Step 7, "Option 0:" on page 132

**Step 2** - Define up to 4 GDG base entries for a consolidation run. Use LIMIT(255) for each of them. Example

```
DEFINE GENERATIONDATAGROUP -  
  (NAME (SAMSDISK.MSGCHECK.CONOLID.GDG.ONE) -  
  LIMIT(255))
```

**Step 3** - Insert the call (listed below) to the application into an ISPF selection panel. The panel can be your primary ISPF panel.

```
11, 'CMD(%SMSPC) NEWAPPL(SMSP)'
```

The corresponding text may look like option 11

```
-----  
ISPF Primary Option Menu  
Option ==>  
  
0 Settings          Terminal and user parameters  
1 View             Display source data or listings  
2 Edit            Create or change source data  
3 Utilities        Perform utility functions  
4 Foreground       Interactive language processing  
5 Batch           Submit job for language processing  
6 Command          Enter TSO or Workstation commands  
7 Dialog Test      Perform dialog testing  
8 LM Facility      Library administrator functions  
9 IBM Products     IBM program development products  
10 SCLM            SW Configuration Library Manager  
11 SMSP           SAMS:Disk Message Check  
IP IPCS           IPCS Problem Analysis Services  
S SDSF            System display and Search Facility  
R BookMgr/Read    IBM BookManager/Read MVS  
TP Third Party    Third Party Products  
M MVS             MVS Support Menu  
N Network         Network Product & Support Menu  
D Database        Database Product & Support Menu
```

Enter X to Terminate using log/list defaults

**Step 4** - Initialize the global environment of the application by entering the following command through the Dialog Service. (ISPF 7.6)

```
SELECT CMD(%SMSPCOK) NEWAPPL(SMSP)
```

For example:

```
----- SAMS:Disk Message Check Application      Enter required field -----  
Command ==>  
  
SAMS:Disk Message Log Output (DSN Prefix will be appended with Jobname)  
data set Prefix ==>  
DASD Unit ==>   can be sysallda  
  
SAMS:Disk Load Library - specify only the default here  
data set Name   ==>  
  
Check Jobs/Steps with DD-Stmt. SYS$EXIT only? ==> (Y/N Yes/No=all Jobs)  
Print Summary Lines from SAMS:Disk?           ==> (Y/N Yes/No)  
Propagate SAMS:Disk Messages to SAMS:Vantage? ==> (0, 1 or 2)  
0 = No, 1 = only recognized Error Messages, 2 = all checked Messages  
  
Library Name   ==>  
for the Tables for this application - Tables are generated automatically.  
(if you want to change this DSN please press HELP for instructions)  
  
Press END KEY to save, Enter CANCEL command to cancel
```

**data set Prefix** - This is the DSN prefix for all message data sets.

**Check Jobs/Steps with DD-Stmt.** - Indicating Yes performs the message check only if SYS\$EXIT DD SYSOUT=\* is in JCL

---

*Note:* You still need to define JOBNAME table (see Step 9) to make this option become effective.

---

**Print Summary Lines from CA-Disk** - Indicating Yes includes report summary lines (for example: total data sets archived ... )

**Library Name** - Is the PDS name created in Step 1. Tables (members) with initial values are created.

These fields can be modified through the panel defined in Step 3.

**Step 5** - Modify the assembler module SYSEXIT\$ (in PTF's source (ASM) library) by adding SYS\$EXIT to the NAME= parameter and then assemble/link the module.

```

SYSEXIT$ TITLE 'SAMS:DISK MESSAGE CHECK EXIT NAME'
*****
      COMPILE ASEM=RENT,LKED=RENT
*
* DESCRIPTION:
*   THIS IS A SAMPLE USERMOD USED TO SPECIFY AN EXIT ROUTINE FOR
*   THE SAMS:DISK MESSAGE CHECK APPLICATION.
*
*   READ THE SYSEXIT$-MACRO PROLOG FOR SPECIFICATIONS OF OPTIONS
*   YOU MAY OVERRIDE.
*****
SYSEXIT$ SYSEXIT$ NAME= SYS$EXIT          SPECIFY AN EXIT NAME FOR THE
*                                     -MESSAGE CHECK APPLICATION
      END

```

Submit the member USERMODL in the INSTALLIB to activate the message check exit.

**Step 6** - You may do one of the following steps if you want to propagate filtered messages to CA-Vantage. You need to repeat this step if any Vantage PTF changes VANSNDM00.

1. Copy load module "VANSNDM00" from CA-Vantage load library to CA-Disk load library
2. Concatenate CA-Vantage load library to CA-Disk load library
3. Put CA-Vantage load library into LINKLIST

**Step 7** - Selecting the option you defined in Step 3 gives you the following panel:

---

*Note:* This Application should be used by Storage Management Experts only.

---

```
----- SAMS:Disk Message Check Application -----  
Select Option ===>  
  
      0 Primary Options  
  
      1 Message Table Maintenance  
  
      2 Jobname Table Maintenance  
  
      3 Activation  
  
      X Exit
```

**Option 0:** - Update fields by selecting this option. To change the PDS created in Step 1, you need to:

1. Allocate a new PDS with the same attributes
2. Copy members (tables with initial values created in Step 4) from the original PDS to the newly allocated PDS
3. Update the library name (the last fields) in this panel

**Option 1:** - To specify message number/word/strings for filtering process. See Step 8 for more details.

**Option 2:** - To specify JOBNAME's for filtering process. See Step 9 for more details.

**Option 3:** - To activate any update made in the previous 3 options.

**Step 8** - To Insert or Update a CA-Disk Message, select option 1 from the panel in the CA-Disk Message Check Application screen to specify a message number, a specific word in message text, or a string in message text to

identify messages you don't want to keep. You can add, delete, or change any entry in this table but make sure there is at least one entry in the table. An example of inserting a new entry is listed below:

```

----- Insert or Update SAMS:Disk Message -----
Command ==>

Message Number    ==>  9999      4 Byte SAMS:Disk Message Number

Check            ==>  W          S=String, W=Word, N=Normal (check only Number)

Offset or Counter ==>  6          Position for "String" or nth "Word" in Message

String or Word    ==>  PDSE

Comment          ==>  Idle space release bypassing PDSE

Enter a SAMS:Disk Message Number and additional check options above:
N: Check Message Number only (normal check)
Other checks:
S: Check for a string starting on column n of the Message Text
   Allowed Range: 1 to 110 (Character 1 is the first character of the Text)
   The Message Number and leading blanks are not counted here.
W: Check for a Word (counts all words in the Message Text and checks the
   The word with the given number)
   Allowed Range: 2 to 64 (word one has to be searched as a string)
   Press END KEY to terminate

```

To interpret the resulting message text:

```
ADSDM341 9999 DATA SET = DDD.SSS.NNN BYPASSED, PDSE DATA SET NOT COMPRESSED
```

9999 indicates the informational message issued by the PDS compress process. Specify W to check for a specific word, the word is the 6<sup>th</sup> word in message text (PDSE) Don't count the module name - ADSDM341 and message number - 9999; any character(s) between two blanks is counted as a word), and the specific word to search is PDSE.

Brief comments are allowed to identify this particular message. Note: We use 9999 as an example because CA-Disk issues various 9999 messages and some of them are not informational but critical errors. For this same message, you may change the entry to check for a string such as 'DATA SET =' while setting the offset field to 1.

Words, a word string in a message text, or message numbers can be entered with N to identify a specific message. A 4 digit message number is required for each entry.

**Step 9** - To Insert or Update a CA-Disk Jobname or Jobname Prefix - Select option 2 from the panel in the CA-Disk Message Check Application screen to specify generic or fully qualified JOBNAME's in this table. These

identify which jobs are eligible for the message check process. You can add, delete, or change any entry in this table. At least one entry in the table is required. An example of inserting a new entry is listed below:

```
-----Insert or Update SAMS:Disk Jobname or Jobname Prefix -----
Command ==>

    Jobname or Prefix ==> dsktest    (see below)

    Type                ==> p        (N=full qualified Jobname, P=Jobname Prefix)

    Comment              ==> sams:disk test jobs

    Include/Exclude     ==> I        (I=Include, E=Exclude this Job)

Enter a Jobname ( full qualified, Type is N )
                Or a
Generic Jobname ( which means you specify a prefix instead a full
                  qualified Jobname and Type must be P )

All Jobs which are excluded by these table entries are bypassed for all
Message checks |

                Press END KEY to terminate
```

To validate a message check against all CA-Disk jobs with JOBNAME DSKTEST\* complete the following steps:

1. Enter the JOBNAME prefix in the Jobname or Prefix field (dsktest)
2. Enter P in the type field (dsktest is the prefix, not full job name)
3. Enter I to include these job(s) in the message check process.

If you have specified Y for Check Jobs/Steps with DD-Stmt. SYS\$EXIT only? ==> (Y/N Yes/No=all Jobs) in Step 2, the DD SYS\$EXIT must be in every JCL of those jobs included for message checking.

**Step 10** - To Update or Change the tables, select option 3 from the panel in the CA-Disk Message Check Application screen to activate the two tables you have updated in Step 8 (Insert or Update CA-Disk Message) and Step 9 (Insert or Update CA-Disk Jobname or Jobname Prefix)

Clearing the first Job Card in this panel sets the entire JCL to defaults.

```
----- JCL to activate the SAMS:Disk Message and Jobname Tables -----
COMMAND ==>
```

JOB Statement Information: (verify before proceeding)

```
====> //ISPAAL1A JOB (ACCOUNT),'NAME'
====> //          NOTIFY=ISPSXC1,CLASS=A,
====> //          MSGLEVEL=(1,1),MSGCLASS=T,
====> //          COND=(0,LT)
====> //*
====> //*
====> //*
```

Press ENTER to proceed

Use END Command to exit

Change the JOB card and review the entire job stream. There are three steps in this job - assemble, link, and copy.

```
----- JCL to activate the SAMS:Disk Message and Jobname Tables --
COMMAND ==>
```

EXEC/DD Statement Information (1 of 3) (verify before proceeding)

```
====> //ASSEMBLY EXEC PGM=ASMA90,PARM='NODECK,OBJECT,LIST'
====> //*
====> //SYSPPRINT DD SYSOUT=*
====> //SYSLIB DD DISP=SHR,DSN=SYS1.MACLIB
====> //SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
====> //SYSLIN DD UNIT=SYSDA,SPACE=(CYL,(1,1,1)),
====> //          DSN=&TEMP(SMSP$TAB),DISP=(,PASS),
====> //          DSORG=PO,BLKSIZE=3200
====> //*
//SYSIN DD * (will be inserted here)
```

Press ENTER to proceed

Use END Command to exit

```
----- JCL to activate the SAMS:Disk Message and Jobname Tables --
COMMAND ==>
```

EXEC/DD Statement Information (2 of 3) (verify before proceeding)

```
====> //BIND EXEC PGM=IEWL,PARM='AC=1,LIST,LET,RENT'
====> //SYSPPRINT DD SYSOUT=*
====> //SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
====> //SYSLMOD DD DISP=(,PASS),DSN=&LOAD,
====> //          SPACE=(CYL,(1,1,1)),DSORG=PS,RECFM=U,
====> //          BLKSIZE=15476,UNIT=SYSDA
====> //SYSLIB DD DISP=(OLD,DELETE),DSN=&TEMP(SMSP$TAB)
====> //SYSLIN DD DDNAME=SYSIN
====> //*
```

---

## Implementation

---

```
//SYSIN      DD *                               will be inserted here
            INCLUDE SYSLIB(HYPSMTAB)           will be inserted here
            NAME SYSEXTAB(R)                   will be inserted here
/*                                                  will be inserted here

            Press ENTER to proceed
            Use END Command to exit
```

If you don't see the copy step listed below, clear the first JCL statement, press enter, and update the job card again.

```
//COPY      EXEC PGM=IEBCOPY,PARM=REPLACE
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD DISP=(OLD,DELETE),DSN=&LOAD
//SYSUT2    DD DISP=OLD,DSN= The.PDS.Created.In.Step1
//SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

Once you have all three steps in place, enter the SUB command and check zero return code of this job. The tables initialized in Step 1 will be replaced.

---

## Implementation

Once you have the application installed and message/jobname table updated, you can start running your CA-Disk jobs with message check in effect. If you specified Y to SYS\$EXIT in the primary options panel, you need to add //SYS\$EXIT DD SYSOUT=\* to the JCL.

For each CA-Disk job qualified for the message check process, a unique message data set will be allocated (and cataloged) or extended with the data set name constructed from the data set prefix specified in the primary options panel (in Step 4 or Step 7 of Installation) and JOBNAME of the job.

For example, if you specified SAMS.DISK.MSGCHK as the data set prefix and job DSKTEST1 is qualified for message check process, the new sequential data set SAMS.DISK.MSGCHK.DSKTEST1 will be allocated and possibly some messages will be written into it. For subsequent jobs with the same jobname DSKTEST1, more messages will be added into the data set.

Eventually you will have many message data sets and should do the consolidate run. Steps for consolidate run are listed below.

### Step 1 - Modify and submit the member MSGCHKJ1.

```
//DOIT      EXEC PGM=IKJEFT01,DYNAMNBR=500,
//          PARM=' %SMSPC005 #DSN.PREFIX.SPECIFIED.IN.ISPF '
//SYSTSPRT  DD SYSOUT=*
//SYSUT2    DD DSN= #INSERT.YOUR.GDGNAME.ONE(+1),
/*          Ø
//          DISP=(,CATLG),SPACE=(189,(5,5),RLSE),AVGREC=K,
//          LRECL=189,RECFM=FB,
//          UNIT=SYSALLDA
//          DCB=(A.VALID.DCB)
/*          Ø
//SYSPROC   DD DSN= #INSERT.SYSPROC.DSN,DISP=SHR
//SYSTSIN   DD DUMMY      Ø
```

Data set prefix specified in the primary options panel.

GDG base defined in Installation Step 2

Create a model DSCB (PS,FB,LRECL=189)

Data set where the application's REXX reside

This job will merge all message data sets into one sequential data set for the next step. All message data sets will be deleted after this point.

The REXX EXEC allocates (concatenates) all message data sets before merging them into a single data set, you may need to run the consolidate jobs more often if the total number of message data sets is more than 255.

**Step 2 - Modify and submit the member MSGCHKJ2.**

```

//SORT2      EXEC  PGM=ICEMAN
//SYSPRINT   DD  SYSOUT=*
//SYSOUT     DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//SORTIN     DD  DSN= #INSERT.YOUR.GDGNAME.ONE ,
//           DISP=OLD
//SORTOUT    DD  DSN= #INSERT.YOUR.GDGNAME.TWO(+1) ,
//           DISP=( ,CATLG,DELETE) ,
//           RECFM=FB,LRECL=189 ,
//           SPACE=(10,(2,1),RLSE) ,AVGREC=K,
//           UNIT=SYSALLDA
//           DCB=(A.VALID.DCB)
//*
//*-----
//SYSIN      DD  *
ALTSEQ CODE=(F402,F501)
SORT FIELDS=(51,1,BI,A,39,4,CH,A,36,2,CH,A,33,2,CH,A,43,8,CH,A,
            1,8,CH,A,51,1,CH,A,53,1,AQ,D,54,2,BI,A)
//*
//SORT3      EXEC  PGM=ICEMAN
//SYSPRINT   DD  SYSOUT=*
//SYSOUT     DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//SORTIN     DD  DSN= #INSERT.YOUR.GDGNAME.TWO(+1) ,
//           DISP=SHR
//SORTOUT    DD  DSN= #INSERT.YOUR.GDGNAME.THREE(+1) ,
//           DISP=( ,CATLG,DELETE) , Ø GDG base defined in Installation Step 2
//           RECFM=FB,LRECL=189 ,
//           SPACE=(100,(2,1),RLSE) ,AVGREC=K,
//           UNIT=SYSALLDA
//           DCB=(A.VALID.DCB)
//*
//*-----
//*          SELECT ERROR MESSAGES
//*-----
//SYSIN      DD  *
OPTION COPY
INCLUDE COND=(51,1,CH,EQ,C' ')
//*
//PRINT4     EXEC  PGM=IEBPTPCH
//SYSPRINT   DD  SYSOUT=*
//SYSUT1     DD  DSN= #INSERT.YOUR.GDGNAME.THREE(+1) ,
//           DISP=SHR
//SYSUT2     DD  SYSOUT=*,CHARS=X15N,
//           LRECL=145,BLKSIZE=145,RECFM=FA
//SYSIN      DD  *
PRINT MAXFLDS=99,MAXLINE=68
TITLE        ITEM=('ERROR MESSAGES FROM SAMS:DISK',11),
            ITEM=(' - ALL JOBS - ',42),
            ITEM=('LISTING FOR STORAGE MANAGEM. TEAM',75),
            ITEM=(' (ROOM:...) ',109)
TITLE        ITEM=('JOBNAME ',1),
            ITEM=('MESSAGE IDENT',11),
            ITEM=('E R R O R M E S S A G E ',28)
RECORD       FIELD=(8,1,,1),
            FIELD=(132,58,,11)

```

GDG base defined in Installation Step 2

GDG base defined in Installation Step 2

Create a model DSCB (PS,FB,LRECL=189)

GDG base defined in Installation Step 2

GDG base defined in Installation Step 2

create a model DSCB (PS,FB,LRECL=189)

GDG base defined in Installation Step 2

```

/**
//PRINT5      EXEC  PGM=IEBTPCH
//SYSPRINT    DD  SYSOUT=*
//SYSUT1T     DD  DSN= #INSERT.YOUR.GDGNAME.TWO(+1),
//            DISP=SHR
//SYSUT2     DD  SYSOUT=*,CHARS=X15N,
//            LRECL=145,BLKSIZE=145,RECFM=FA
//SYSIN       DD  *
PRINT MAXFLDS=99,MAXLINE=68
TITLE        ITEM=('ERROR MSG. AND SUMMARY LINES ',10),
             ITEM=(' - ALL JOBS, COMPLETE LIST ',42),
             ITEM=('PLEASE DISTRIBUTE TO STOR. MGMT. ',75),
             ITEM=(' (ROOM ... )',109)
TITLE        ITEM=('JOBNAME',1),
             ITEM=('STEPNAME',10),
             ITEM=('PROCSTEP',19),
             ITEM=('DATE + START TIME ',28),
             ITEM=('MESSAGE - OR -',49),
             ITEM=('SUM. LINE ',66)
RECORD       FIELD=(8,1,,1),
             FIELD=(8,9,,10),
             FIELD=(8,17,,19),
             FIELD=(10,33,,28),
             FIELD=(8,43,,39),
             FIELD=(102,58,,49)

/**
//COPY5       EXEC  PGM=ICEGENER
//SYSUT1      DD  DSN= #INSERT.YOUR.GDGNAME.ONE,
//            DISP=(OLD,DELETE,KEEP)
//SYSUT2      DD  DSN= #INSERT.YOUR.GDGNAME.FOUR(+1),
//            DISP=(,CATLG,DELETE),
//            SPACE=(500,(2,1),RLSE),AVGREC=K,
//            UNIT=SYSALLDA
//            DCB=(A.VALID.DCB)
//            create a model DSCB
//            (PS,FB,LRECL=189)
/**
//*SCRATCH2   DD  DSN= #INSERT.YOUR.GDGNAME.TWO,
//            DISP=(OLD,DELETE,KEEP)
//*SCRATCH3   DD  DSN= #INSERT.YOUR.GDGNAME.THREE,
//            DISP=(OLD,DELETE,KEEP)
//SYSIN       DD  *
//SYSPRINT    DD  SYSOUT=*

```

GDG base defined in Installation Step 2

This job uses the sequential data set created in the previous step to create various work data sets and generate a final report ( SYSUT2 of COPY5 step). You may de-comment the two DD SCRATCH2 and SCRATCH3 so the work data sets will be deleted. The purpose of keeping these work data sets is to run an optional third job (MSGCHKJ3) to generate different reports.

---

# *Index*

---

## **A**

### ACF2

- installation 26
- security interface 27

### activating

- DASD billing 48
- DBRC interface 48, 49
- ISPF support 43
- MTFDEBUG DD statement security feature 40
- SAMS:Disk features 25
- security features 25, 33
- security interfaces 25
- the SAMS:Disk system 17
- TSO support 46
- VSAM support 41

### activation codes 18

### APF-authorized libraries 10

### auto-restore

- considerations 81
- CSA requirements 11
- DASD Pools 70
- DMSAUTx enqueues 81
- Enqueues 81
- functional overview 58
- GRS problems 81
- implementation guidelines 80
- installing the function 61
- installing the test hook interface 67
- ISPF customization 77

### JES awareness 11

### methods available 59

### MIM problems 81

### operating the test hook interface 68

### preparing to use the test hook interface 66

### restrictions 81

### test interface 64

### testing the facility 107

### TSO customization 77

### VSAM alternate indexes 81

## **C**

### CA1 interface considerations 89

### CA-ACF2

#### installation 26

#### security interface 27

### catalog status controlled tapes 88

### CA-top secret

#### installation 28

#### security interface 29

### connecting the ISPF libraries to your system 45

### controlling tapes

#### catalog status 88

#### EDM interface 86

#### expiration date 88

#### TLMS/EDM 89

---

creating FILEDEFN 23  
customization 9, 51  
customizing the SAMS:Disk ISPF support 45

## D

DASD billing  
    activating the support 48  
DASD Pools 70  
DBRC  
    activating the support 48, 49  
DD statements  
    MTFDEBUG 40  
disk space required for installation 10  
DMSAR  
    considerations 81  
    CSA requirements 11  
    DASD Pools 70  
    DMSAUTx enqueues 81  
    Enqueues 81  
    functional overview 58  
    GRS problems 81  
    implementation guidelines 80  
    installing the function 61  
    installing the test hook interface 67  
    ISPF customization 77  
    JES awareness 11  
    methods available 59  
    MIM problems 81  
    operating the test hook interface 68  
    preparing to use the test hook interface 66  
    restrictions 81  
    test interface 64  
    testing the facility 107  
    TSO customization 77  
DMSAUTx enqueues 81  
download the related installation materials library 14  
DSCB update utility 109  
    condition codes 110  
    conversion of F1-DSCBs 124  
    DSCB command 110  
    syntax 110  
    user exits 114

DSCL 89  
dynamic menu-formatting feature 43  
dynamic restore command processor 47

## E

EDM interface 86  
enqueue usage of SAMS:Disk 24, 81  
environment  
    preparing 14  
evaluation 96  
expiration controlled tapes 88

## F

F1-DSCB Not Found Abend exit 126  
FDSAMPLE 23  
FILEDEFN 23  
files dataset 18, 23  
    conversion 118  
formatting used for on-line reporting 43

## G

GRS enqueue problems 81

## H

help library 11

## I

IDATMSTP 58  
IFG0EX0A exit 126  
IGGDASU2 84  
installation 8, 13  
    planning 10  
    summary 7  
installing  
    auto-restore 61  
    CA-ACF2 security interface 27  
    CA-top secret security interface 29  
    IGGDASU2 user Exit 84  
    RACF security interface 30  
    SVC 54  
    test auto-restore hook interface 67  
    Under CA-ACF2 26  
    under CA-top secret 28  
    Under RACF 30  
    usermods 40  
    VSAM date stamp 58

---

## ISPF

- activating the support 43
- auto-restore customization 77
- capabilities of the interface 89
- connecting the libraries to your system 45
- customizing the support 45
- defining user options 89
- DMSAR customization 77
- dynamic menu-formatting feature 43
- formatting used for on-line reporting 43

## IVP tests 96

## IXCATLG utility 115

- command and parameters 116
- description of 115
- user screening exit 117

## J

## JCL procedures 20

## L

## limiting TSO user access 94

## Linklist library 11

## LPA list library 11

## M

## make the command processor available to TSO 46

## MIM enqueue problems 81

## MTFDEBUG DD statement 40

## multiple releases of SAMS:Disk 14

## N

## naming conventions used during installation 10

## O

## on-line reporting 43

## operating the test auto-restore hook interface 68

## P

## PARMAUTH 33

## PARMLIB 12, 17, 18, 33, 36

## preinstallation 7

## preparing for multiple releases of SAMS:Disk 14

## preparing the environment 14

## R

## RACF

- security interface 30

## restore 47

## RIMLIB 14

## S

## S213 Abend Exit 126

## S213 Abend exit 126

## security features

- activating 33
- DD statements 40
- PARMLIB 33
- system parameter overrides 36

## security interfaces

- CA-ACF2 27
- CA-top secret 29
- RACF 30

## SET command syntax 18

## SMP/E 15

- accept SAMS:Disk list 16
- Allocate SAMS:Disk datasets 15
- apply maintenance 16
- apply SAMS:Disk 16
- DDDEF entries 15
- receive SAMS:Disk 15

## summary of installation 7

## SVC 11, 121

- conversion of F1-DSCBs 122
- installing 54
- modes 53
- overview 52
- tailoring of 54

## SYSPAUTH 36

## system parameter override 36

## T

## table of userMODS 40

## tape management support 85

## tape structure 14

## tape unit name 94

## tape units allocated concurrently 94

## testing the system

- list of IVPs to execute 96

## TLMS/EDM 89

## top secret

- installation 28
- security interface 29

---

## TSO

- activating the support 46
- auto-restore customization 77
- customizing the support 94
- DMSAR customization 77
- dynamic restore command processor 47
- limiting user access 94
- making the command processor available 46
- screening exits for archive and restore 95

## U

user options within ISPF 89

### userMOD

- ADSUMODB 40
- ADSUMODC 40
- ADSUMODG 40
- ADSUMODK 40
- ADSVS939 40
- DSNEXMPT 40
- JOBEXMPT 40
- LINEMAX 40
- MTFLOCK 40
- PETLOCK 40
- PGMEXMPT 40
- SYSOUTEX 40
- table of 40
- TASKLIB 40
- TSOEXMPT 40

usermods 40

- userMOD5 35
- userMOD8 79

### utilities

- DSCB update 110
- IXCATLG 115

## V

### VSAM

- activating the support 41
- alternates indexes and auto-restore 81
- date stamp 57
- installing the date stamp 58